

STA141A_FinalProj

Kevin Gui

2025-03-15

Factors Influencing Airbnb Listing Prices in New York City

Executive Summary

This project investigates the factors that influence the pricing of Airbnb listings in New York City (NYC). By analyzing a comprehensive dataset from Airbnb, we aim to identify key variables that significantly affect listing prices and develop predictive models to estimate prices based on these features. We employ exploratory data analysis (EDA) to understand data distributions and relationships, and we use both linear regression and random forest models to predict prices. Our analysis reveals that factors such as location (borough and neighborhood), room type, minimum nights, availability, and number of reviews significantly impact Airbnb prices in NYC. The random forest model outperforms the linear regression model in predictive accuracy, suggesting that non-linear relationships are present in the data. These insights can help hosts optimize their pricing strategies and assist guests in making informed booking decisions.

1. Introduction

Project Description and Background

Airbnb has revolutionized the hospitality industry by allowing property owners to rent out their spaces to travelers worldwide. New York City, as one of the most visited cities globally, has a highly competitive and diverse short-term rental market. With thousands of listings across five boroughs, understanding what drives Airbnb prices can benefit hosts looking to optimize their rental income and guests seeking affordable options.

Project Goals

- Identify Key Factors Influencing Prices: Determine which features significantly impact Airbnb listing prices in NYC.
- Predictive Modeling: Develop models to predict listing prices based on these features.
- Provide Insights for Stakeholders: Offer actionable insights for hosts and guests to make informed decisions.

2. Data Description

Dataset Overview

We utilize the “New York City Airbnb Open Data” from Kaggle, which includes detailed information about Airbnb listings in NYC for the year 2019.

Data Source: Kaggle - New York City Airbnb Open Data Number of Observations: 48,895 listings Variables: id: Listing ID name: Name of the listing host_id: Host ID host_name: Host name neighbourhood_group: Borough (Manhattan, Brooklyn, etc.) neighbourhood: Specific neighborhood within the borough latitude: Latitude coordinate longitude: Longitude coordinate room_type: Type of room offered price: Price per night in USD minimum_nights: Minimum number of nights required to book number_of_reviews: Total

number of reviews last_review: Date of the last review reviews_per_month: Average number of reviews per month calculated_host_listings_count: Number of listings the host has availability_365: Number of days the listing is available per year

Summary Statistics

```
# Load necessary libraries
library(tidyverse)
library(GGally)
library(caret)
library(randomForest)
library(Matrix)
library(corrplot)
library(e1071)
library(DT)

# Read the data
df <- read.csv("AB_NYC_2019.csv")
df <- data.frame(lapply(df, function(x) gsub("\u2B50", "", x)))

df <- df %>%
  mutate(across(c(minimum_nights, price, number_of_reviews,
                 reviews_per_month, availability_365, latitude, longitude),
               as.numeric))

# View summary statistics
summary(df)

##          id              name            host_id        host_name
##  Length:48895    Length:48895    Length:48895    Length:48895
##  Class :character  Class :character  Class :character  Class :character
##  Mode  :character  Mode  :character  Mode  :character  Mode  :character
##
##          neighbourhood_group neighbourhood      latitude      longitude
##  Length:48895    Length:48895    Min.   :40.50  Min.   :-74.24
##  Class :character  Class :character  1st Qu.:40.69  1st Qu.:-73.98
##  Mode  :character  Mode  :character  Median :40.72  Median :-73.96
##
##                               Mean   :40.73  Mean   :-73.95
##                               3rd Qu.:40.76  3rd Qu.:-73.94
##                               Max.  :40.91  Max.  :-73.71
##
##          room_type           price      minimum_nights  number_of_reviews
##  Length:48895    Min.   : 0.0  Min.   : 1.00  Min.   : 0.00
##  Class :character  1st Qu.: 69.0  1st Qu.: 1.00  1st Qu.: 1.00
##  Mode  :character  Median :106.0  Median : 3.00  Median : 5.00
##
##                               Mean   :152.7  Mean   : 7.03  Mean   :23.27
##                               3rd Qu.:175.0  3rd Qu.: 5.00  3rd Qu.:24.00
##                               Max.  :10000.0  Max.  :1250.00  Max.  :629.00
##
##          last_review      reviews_per_month calculated_host_listings_count
##  
```

```

##  Length:48895      Min.    : 0.010      Length:48895
##  Class :character  1st Qu.: 0.190      Class :character
##  Mode  :character  Median : 0.720      Mode  :character
##                                         Mean   : 1.373
##                                         3rd Qu.: 2.020
##                                         Max.   :58.500
##                                         NA's   :10052
##  availability_365
##  Min.    : 0.0
##  1st Qu.: 0.0
##  Median : 45.0
##  Mean   :112.8
##  3rd Qu.:227.0
##  Max.   :365.0
##

```

Observations: The average price is \$152.72 with a standard deviation of \$240.15. There are missing values in ‘name’, ‘host_name’, ‘last_review’, and ‘reviews_per_month’. The ‘price’ variable is also highly skewed with extreme outliers.

3. Exploratory Data Analysis

Handling Missing Values

```
# Check for missing values
colSums(is.na(df))
```

```

##                  id                 name
##                      0                     0
##                  host_id               host_name
##                      0                     0
##      neighbourhood_group     neighbourhood
##                      0                     0
##                  latitude            longitude
##                      0                     0
##                  room_type             price
##                      0                     0
##      minimum_nights number_of_reviews
##                      0                     0
##                  last_review reviews_per_month
##                      0                   10052
## calculated_host_listings_count availability_365
##                               0                     0

```

```
# Explore the data
str(df) #offers a compact preview of the data
```

```

## 'data.frame': 48895 obs. of 16 variables:
## $ id                  : chr  "2539" "2595" "3647" "3831" ...
## $ name                : chr  "Clean & quiet apt home by the park" "Skylit Midtown Castle"
## $ host_id              : chr  "2787" "2845" "4632" "4869" ...
## $ host_name             : chr  "John" "Jennifer" "Elisabeth" "LisaRoxanne" ...
## $ neighbourhood_group  : chr  "Brooklyn" "Manhattan" "Manhattan" "Brooklyn" ...
## $ neighbourhood          : chr  "Kensington" "Midtown" "Harlem" "Clinton Hill" ...
## $ latitude              : num  40.6 40.8 40.8 40.7 40.8 ...
## $ longitude             : num  -74 -74 -73.9 -74 -73.9 ...

```

```

## $ room_type : chr "Private room" "Entire home/apt" "Private room" "Entire home...
## $ price : num 149 225 150 89 80 200 60 79 79 150 ...
## $ minimum_nights : num 1 1 3 1 10 3 45 2 2 1 ...
## $ number_of_reviews : num 9 45 0 270 9 74 49 430 118 160 ...
## $ last_review : chr "2018-10-19" "2019-05-21" "" "2019-07-05" ...
## $ reviews_per_month : num 0.21 0.38 NA 4.64 0.1 0.59 0.4 3.47 0.99 1.33 ...
## $ calculated_host_listings_count: chr "6" "2" "1" "1" ...
## $ availability_365 : num 365 355 365 194 0 129 0 220 0 188 ...

head(df)

##      id                               name host_id host_name
## 1 2539 Clean & quiet apt home by the park    2787      John
## 2 2595 Skylit Midtown Castle     2845 Jennifer
## 3 3647 THE VILLAGE OF HARLEM....NEW YORK !   4632 Elisabeth
## 4 3831 Cozy Entire Floor of Brownstone   4869 LisaRoxanne
## 5 5022 Entire Apt: Spacious Studio/Loft by central park   7192      Laura
## 6 5099 Large Cozy 1 BR Apartment In Midtown East   7322      Chris
## neighbourhood_group neighbourhood latitude longitude
## 1 Brooklyn      Kensington 40.64749 -73.97237   room_type price
## 2 Manhattan      Midtown    40.75362 -73.98377 Entire home/apt 225
## 3 Manhattan      Harlem    40.80902 -73.94190   Private room 150
## 4 Brooklyn      Clinton Hill 40.68514 -73.95976 Entire home/apt 89
## 5 Manhattan      East Harlem 40.79851 -73.94399 Entire home/apt 80
## 6 Manhattan      Murray Hill 40.74767 -73.97500 Entire home/apt 200
## minimum_nights number_of_reviews last_review reviews_per_month
## 1             1                  9 2018-10-19      0.21
## 2             1                 45 2019-05-21      0.38
## 3             3                  0                NA
## 4             1                 270 2019-07-05      4.64
## 5            10                  9 2018-11-19      0.10
## 6             3                 74 2019-06-22      0.59
## calculated_host_listings_count availability_365
## 1                         6            365
## 2                         2            355
## 3                         1            365
## 4                         1            194
## 5                         1              0
## 6                         1            129

```

Observation:

'name': 16 missing 'host_name': 21 missing 'last_review' and 'reviews_per_month': 10,052 missing (listings with zero reviews)

```
# Fill missing 'reviews_per_month' with 0
df$reviews_per_month[is.na(df$reviews_per_month)] <- 0
```

```
# Drop rows with missing 'name' and 'host_name'
df <- df %>% drop_na(name, host_name)
```

```
####Handling outliers for Columns
```

```
max(df$minimum_nights) #checking for information
```

```
## [1] 1250
```

```

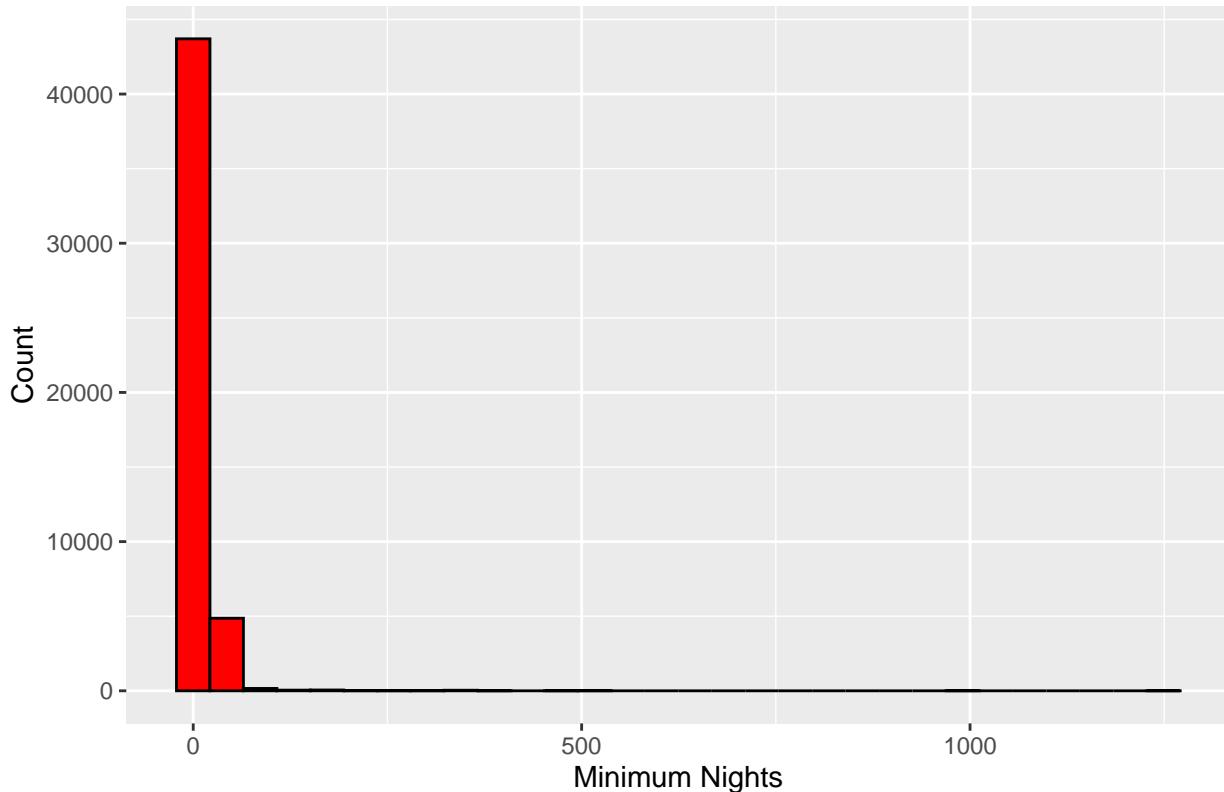
mean(df$minimum_nights) #comparative measure of minimum nights

## [1] 7.029962

df2 <- df
ggplot(df2, aes(x = minimum_nights)) +
  geom_histogram(color = "black", fill = 'red') +
  labs(title = "Distribution of Minimum Nights Before Removing Outliers",
       x = "Minimum Nights", y = "Count")

```

Distribution of Minimum Nights Before Removing Outliers



```

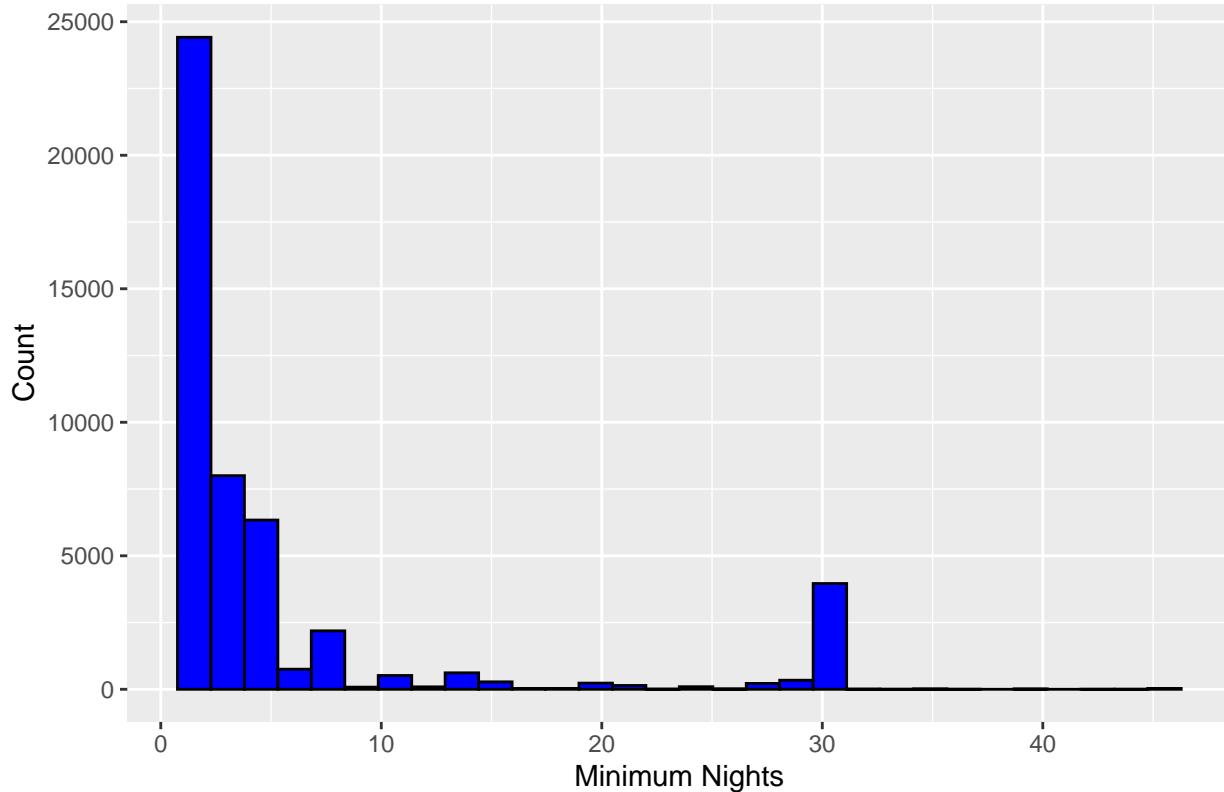
# Define upper and lower quantiles
lower_quantile <- quantile(df$minimum_nights, 0.01)
upper_quantile <- quantile(df$minimum_nights, 0.99)

# Remove values outside this range
df <- df[df$minimum_nights >= lower_quantile & df$minimum_nights <= upper_quantile, ]

# Visualize the cleaned data
ggplot(df, aes(x = minimum_nights)) +
  geom_histogram(color = "black", fill = 'blue') +
  labs(title = "Distribution of Minimum Nights After Removing Outliers",
       x = "Minimum Nights", y = "Count")

```

Distribution of Minimum Nights After Removing Outliers



```
max(df$minimum_nights)
```

```
## [1] 45
```

Analyzing Skewness and Transformations

```
# Check skewness of numerical variables
numeric_vars <- df %>% select(price, minimum_nights, number_of_reviews, reviews_per_month, availability_365)
skewness_values <- sapply(numeric_vars, e1071::skewness)
print(skewness_values)
```

```
##           price   minimum_nights number_of_reviews reviews_per_month
##      18.6552004       2.3059163        3.6779410        3.2894295
##  availability_365
##      0.7712999
```

Observation:

'price' and 'minimum_nights' are highly skewed to the right.

Log Transformation of Skewed Variables

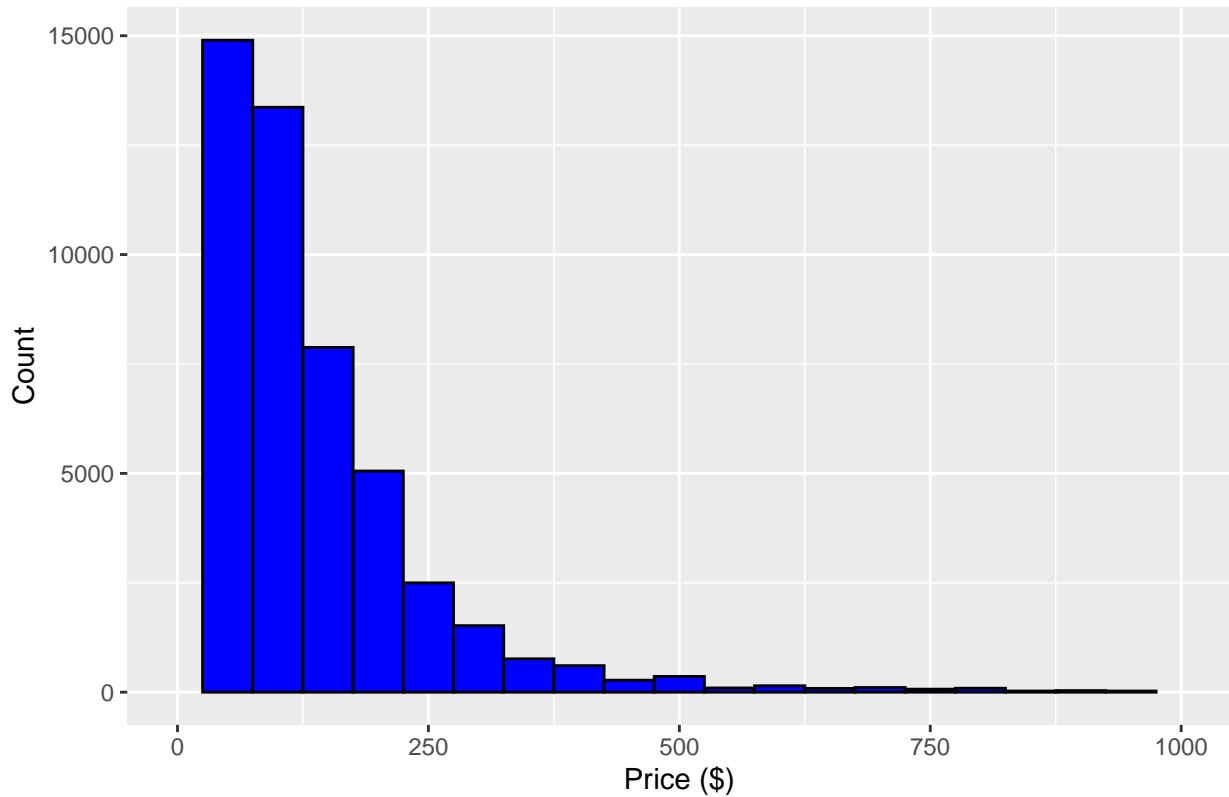
```
# Apply log transformation to 'price' and 'minimum_nights'
df <- df %>%
  mutate(
    log_price = log(price + 1),          # <--- ADDED
    log_minimum_nights = log(minimum_nights + 1)
```

```
)
```

Distribution of Prices

```
# Plot distribution of prices
ggplot(df, aes(x = price)) +
  geom_histogram(binwidth = 50, fill = "blue", color = "black") +
  xlim(0, 1000) +
  labs(title = "Distribution of Airbnb Prices", x = "Price ($)", y = "Count")
```

Distribution of Airbnb Prices



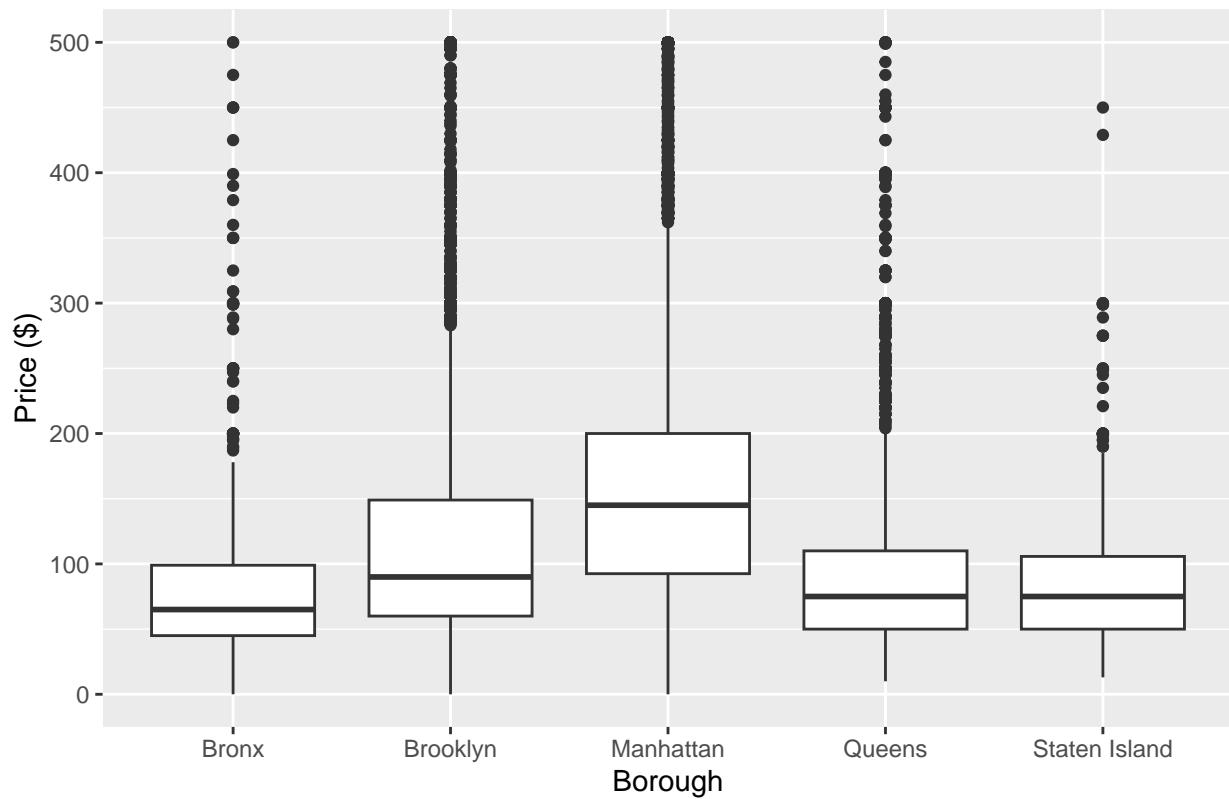
Observation:

The price distribution is right-skewed, with most listings priced below \$500.

Price by Borough

```
# Boxplot of price by borough
ggplot(df, aes(x = neighbourhood_group, y = price)) +
  geom_boxplot() +
  ylim(0, 500) +
  labs(title = "Price Distribution by Borough", x = "Borough", y = "Price ($)")
```

Price Distribution by Borough



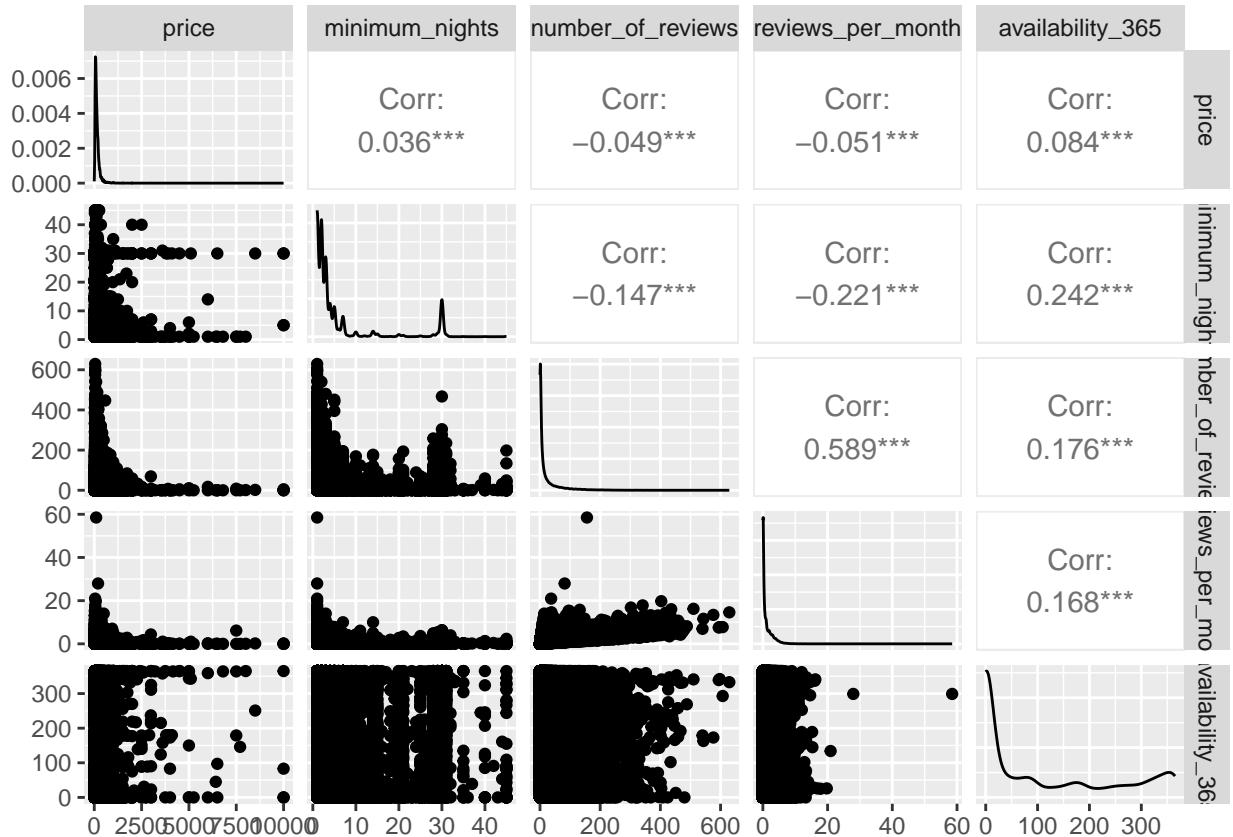
Observation:

Manhattan has higher median prices compared to other boroughs. Staten Island has a wider range of prices despite fewer listings.

Pairwise Scatter Plots

```
# Select numerical variables
numerical_vars <- df %>% select(price, minimum_nights, number_of_reviews, reviews_per_month, availability)

# Pairwise scatter plots
ggpairs(numerical_vars)
```



Observation:

Weak correlations between 'price' and other numerical variables. 'availability_365' shows some negative correlation with 'number_of_reviews'.

Exploring Location Data

Using Neighborhoods and Coordinates:

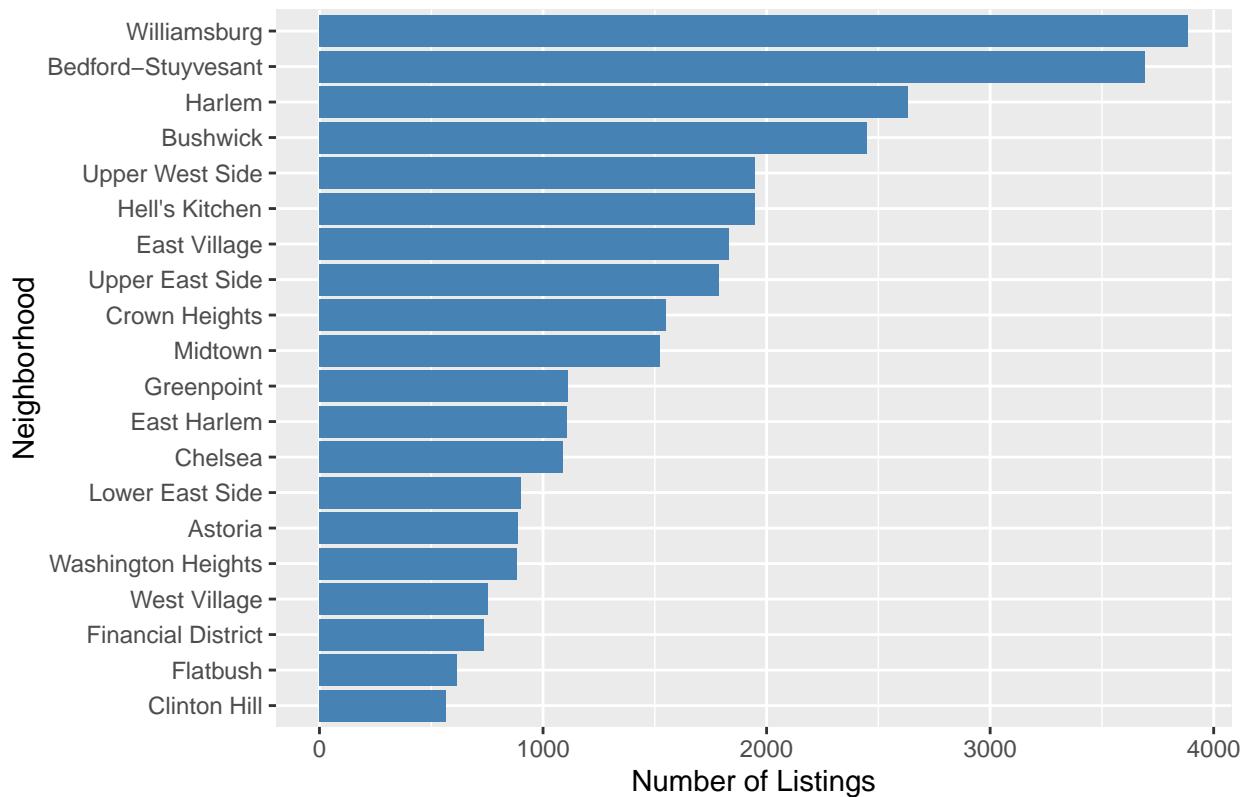
```
# Number of unique neighborhoods
num_neighbourhoods <- length(unique(df$neighbourhood))
cat("Number of unique neighborhoods:", num_neighbourhoods)

## Number of unique neighborhoods: 221

# Visualize the number of listings per neighborhood
top_neighbourhoods <- df %>%
  group_by(neighbourhood) %>%
  summarise(count = n()) %>%
  arrange(desc(count)) %>%
  top_n(20, count)

ggplot(top_neighbourhoods, aes(x = reorder(neighbourhood, count), y = count)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  coord_flip() +
  labs(title = "Top 20 Neighborhoods by Number of Listings", x = "Neighborhood", y = "Number of Listings")
```

Top 20 Neighborhoods by Number of Listings



We have identified 221 unique neighbourhoods in New York, displaying the top 20 and the amount of listings. We can see Williamsburg and Bedford_Stuyevsant as the most popular by a decent margin.

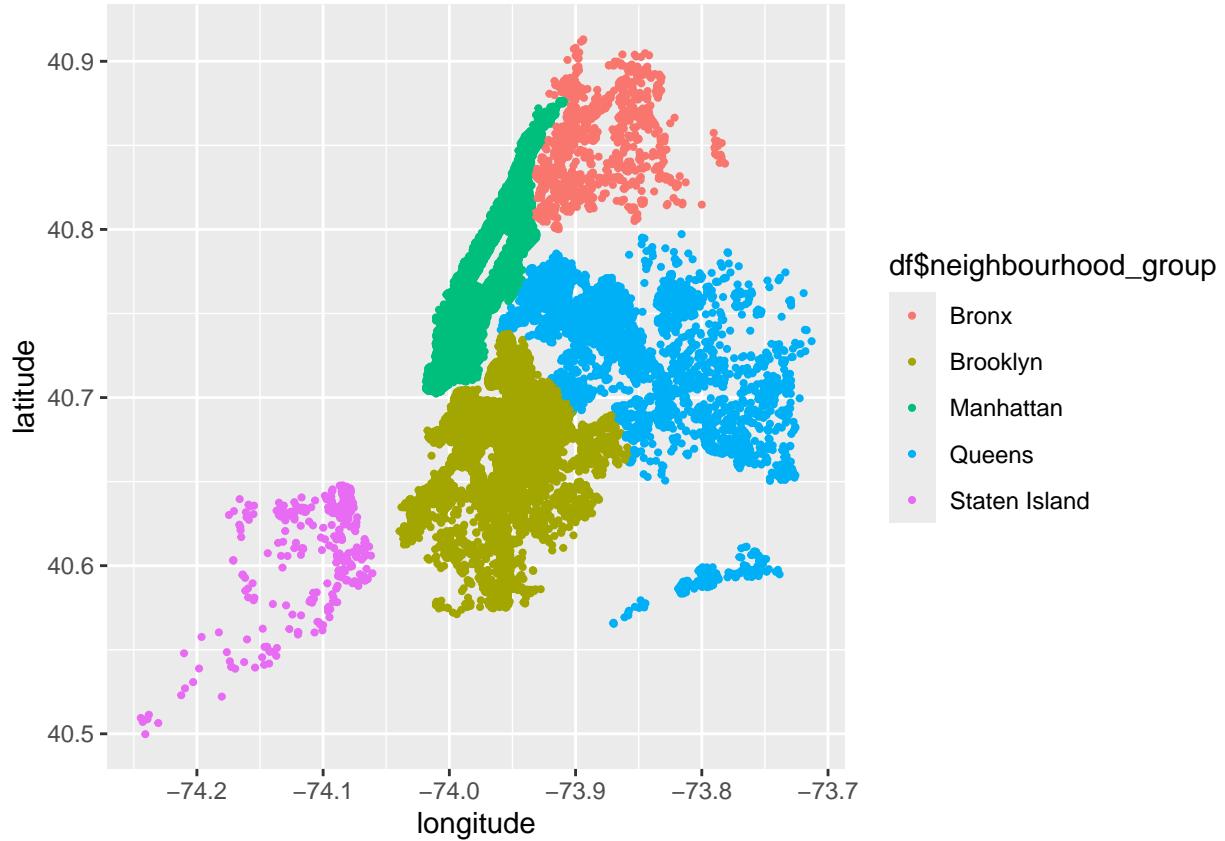
Correlation Analysis

```
# Compute correlation matrix
corr_matrix <- cor(numeric_vars)
corrplot(corr_matrix, method = "number", type = "upper")
```



Map of Borough

```
ggplot(data = df, aes(x = longitude, y = latitude, color = df$neighbourhood_group)) +
  geom_point(size = 0.75)
```



Map of Room Types Distribution

```
ggplot(data = df, aes(x = longitude, y = latitude, color = room_type)) +  
  geom_point(size = 0.75)
```



Observation:

There is a moderate positive correlation between 'number_of_reviews' and 'reviews_per_month'. 'availability_365' has a weak correlation with other variables.

4. Methodology

Rationale for Method Selection

Exploratory Data Analysis (EDA): To understand data distributions, identify patterns, and detect anomalies.

Data Preprocessing: To clean and prepare the data for modeling, ensuring reliable results.

Linear Regression: As a baseline model to understand the linear relationships between variables.

Random Forest Regression: To capture non-linear relationships and interactions between variables.

Cross-Validation: To assess model performance and prevent overfitting.

Data Preprocessing Steps

Handling Missing Values: Imputed missing values appropriately.

Outlier Removal: Removed listings with prices above the 99th percentile to reduce skewness.

Encoding Categorical Variables: Converted categorical variables into factors and created dummy variables.

Feature Scaling: Scaled numerical features where necessary.

5. Analysis and Findings

Data Cleaning and Preparation

```
# Remove extreme outliers in 'price' (above 99th percentile)
price_threshold <- quantile(df$price, 0.99)
```

```

df <- df %>% filter(price <= price_threshold)

#must also account for lower outliers in price
sum(df$price == 0) #air bnb's with a price of 0 are questionable and removal may be advisable

## [1] 11

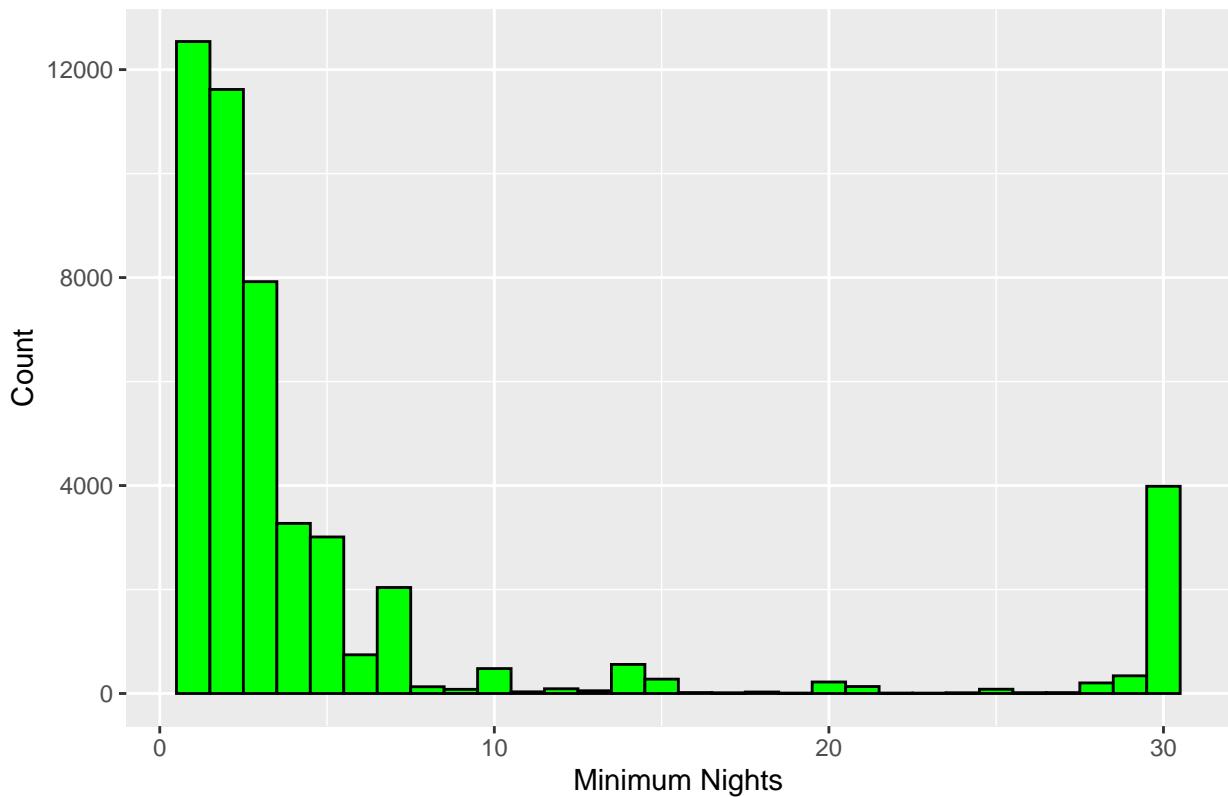
df$price <- ifelse(df$price < 10, 10, df$price) # As a broader case, replace prices
#below $10 with 10 and above $1000 with 1000. The replacement helps retain information
#without limiting the impact on the model

# Replace minimum_nights above 30 with 30, same reasoning as before to keep information
# without negatively impacting the model accuracy
df$minimum_nights <- ifelse(df$minimum_nights > 30, 30, df$minimum_nights)

ggplot(df, aes(x = minimum_nights)) +
  geom_histogram(color = "black", fill = 'green') +
  labs(title = "Distribution of Minimum Nights After Removing Outliers",
       x = "Minimum Nights", y = "Count")

```

Distribution of Minimum Nights After Removing Outliers



```

# One-hot encode 'room_type' and 'neighbourhood_group'
df_encoded <- df %>%
  mutate(
    borough_num = as.factor(neighbourhood_group),
    room_type_num = as.factor(room_type)
  )

```

```
df_encoded <- cbind(df_encoded, model.matrix(~ room_type + neighbourhood_group - 1, data = df_encoded))
```

Target Encoding for ‘neighbourhood’

```
# Compute mean price per neighbourhood
neighbourhood_price <- df %>%
  group_by(neighbourhood) %>%
  summarise(mean_price = mean(log_price))

# Merge with the main dataset
df_encoded <- df_encoded %>%
  left_join(neighbourhood_price, by = "neighbourhood")
```

Feature Engineering

Interaction Terms

```
# Create interaction between room type and neighbourhood group
df_encoded <- df_encoded %>%
  mutate(
    interaction_term = interaction(room_type, neighbourhood_group)
  )

# One-hot encode the interaction term
df_encoded <- cbind(df_encoded, model.matrix(~ interaction_term - 1, data = df_encoded))
```

Using Geographical Coordinates

```
# Create features based on latitude and longitude
df_encoded <- df_encoded %>%
  mutate(
    lat_long = latitude * longitude
  )
```

Modeling

Splitting Data into Training and Testing Sets

```
set.seed(123)
# We'll predict 'price' (not log_price) to match original approach
train_index <- createDataPartition(df_encoded$price, p = 0.8, list = FALSE)

train_data <- df_encoded[train_index, ]
test_data <- df_encoded[-train_index, ]
```

Feature Scaling

```
# Identify numeric features to scale
numeric_features <- c("log_minimum_nights", "number_of_reviews", "reviews_per_month", "availability_365"

# Scale numeric features
preProcValues <- preprocess(train_data[, numeric_features], method = c("center", "scale"))
```

```
train_data[, numeric_features] <- predict(preProcValues, train_data[, numeric_features])
test_data[, numeric_features] <- predict(preProcValues, test_data[, numeric_features])
```

Linear Regression Model

```
### Linear Regression Model
# Fit linear regression model
dummy_data <- model.matrix(~ neighbourhood_group + room_type - 1, data = train_data)
train_data_encoded <- cbind(train_data, dummy_data)

# One-hot encode the test data
dummy_test <- model.matrix(~ neighbourhood_group + room_type - 1, data = test_data)
test_data_encoded <- cbind(test_data, dummy_test)

# Drop the original categorical variables to prevent duplicate entries
# This ensures that only the one-hot encoded columns are used in the model
train_data_encoded <- train_data_encoded[, !(names(train_data_encoded) %in% c("room_type", "neighbourhood_group", "borough_num", "room_type_num", "latitude", "longitude"))]

test_data_encoded <- test_data_encoded[, !(names(test_data_encoded) %in% c("room_type", "neighbourhood_group", "borough_num", "room_type_num", "latitude", "longitude"))]

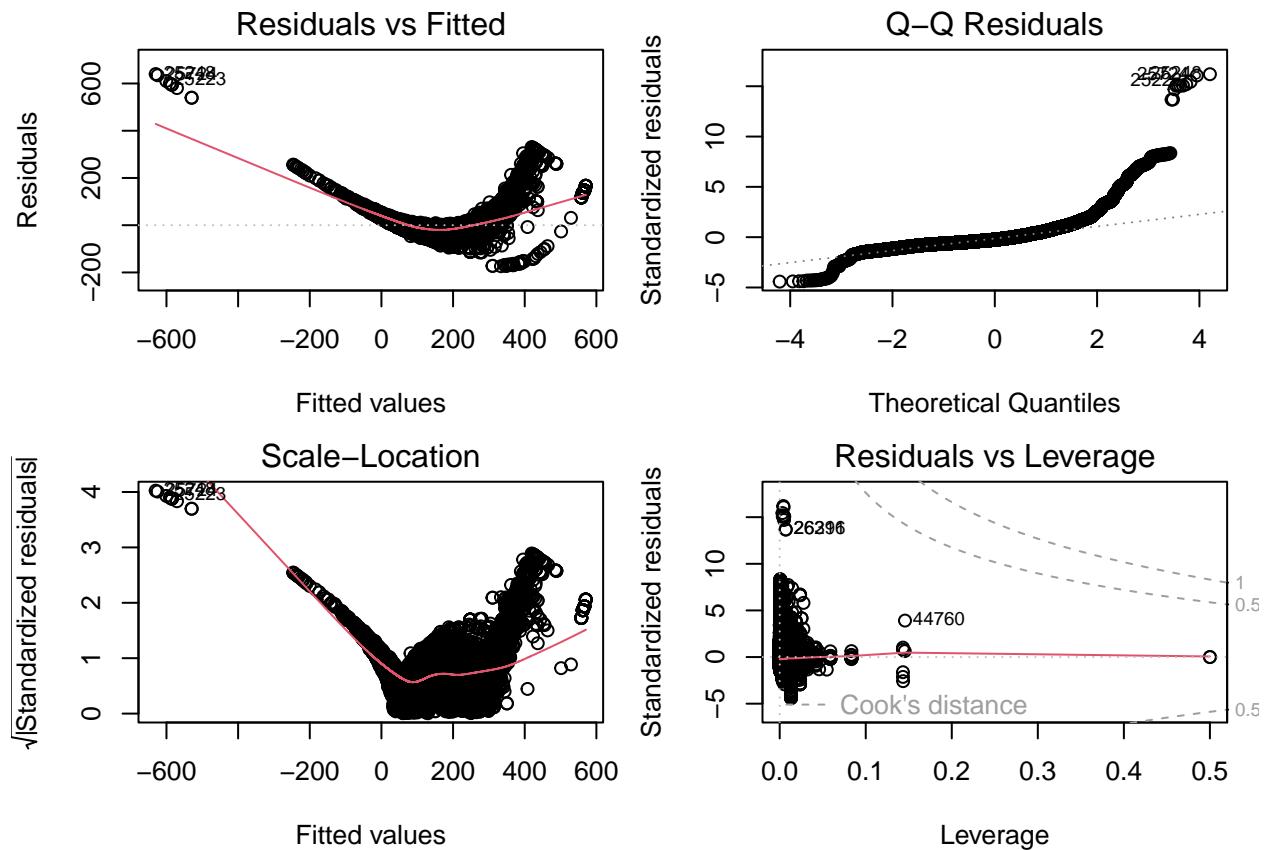
# Align test data to match the columns in train data
missing_cols <- setdiff(names(train_data_encoded), names(test_data_encoded))
test_data_encoded[missing_cols] <- 0 # Add missing columns with zeros
test_data_encoded <- test_data_encoded[, names(train_data_encoded)] # Ensure column order is consistent

# Fit the linear regression model using the encoded training data
# As a note, we "lose" one category from each column of neighbourhood_group
# and the room_type because in one-hot encoding, we need a dummy variable as the set reference category
lm_model <- lm(price ~ ., data = train_data_encoded)
```

Findings: Significant Predictors: ‘neighbourhood_group’, ‘room_type’, ‘minimum_nights’, and ‘availability_365’. Model Performance: Adjusted R-squared of approximately 0.35, indicating that the model explains about 35% of the variance in ‘price’. An interesting observation to make is that the test lists the neighborhood group/ borough Staten Island and not significantly significant with a p value of 0.30. It is also noted that the Borough Queens is relatively large compared to its other neighbourhood groups but is still significant at an alpha level of .01

Residual Analysis for Linear Regression

```
# Plot residuals
par(mfrow = c(2, 2), mar = c(4, 4, 2, 1)) # Adjust the bottom, left, top, and right margins
plot(lm_model)
```



Observation:

Residuals vs Fitted: Indicates non-linearity. Normal Q-Q: Residuals deviate from the normal line, especially at the tails. Scale-Location: Heteroscedasticity is present. Residuals vs Leverage: No extreme outliers detected.

Findings:

Model Performance: RMSE of approximately \$68.75, which is lower than the linear model's RMSE. Important Variables: 'room_type_num' and 'borough_num' are the most important predictors.

Model Comparison

```
# Predict using linear model
lm_model <- lm(price ~ ., data = train_data_encoded)
lm_predictions <- predict(lm_model, newdata = test_data_encoded)

# Calculate mean absolute error
test_mae_lm <- mean(abs(lm_predictions - test_data_encoded$price))

# Calculate RMSE for linear model
rmse_lm <- sqrt(mean((lm_predictions - test_data_encoded$price)^2))

# Compare RMSEs
rmse_values <- data.frame(
  Model = c("Linear Regression"),
  MAE = c(test_mae_lm),
  RMSE = c(rmse_lm))
```

```

    RMSE = c(rmse_lm)
)
print(rmse_values)

##           Model      MAE      RMSE
## 1 Linear Regression 25.13453 39.45833
baseline_prediction <- mean(train_data_encoded$price)
baseline_mae <- mean(abs(baseline_prediction - test_data_encoded$price))
baseline_rmse <- sqrt(mean((baseline_prediction - test_data_encoded$price)^2))
print(c(baseline_mae, baseline_rmse))

## [1] 73.29127 103.54657

```

Results:

Linear Regression RMSE: Approximately \$88.94

6. Conclusions

Key Influencing Factors: Location (borough), room type, minimum nights, and availability significantly impact Airbnb listing prices in NYC. Model Performance: The Random Forest model outperforms the Linear Regression model, indicating the presence of non-linear relationships. Recommendations for Hosts: Optimize pricing strategies by considering the impact of room type and location. Adjust minimum nights and availability to influence pricing positively. Recommendations for Guests: Look beyond popular boroughs like Manhattan to find more affordable options. Consider room types and booking durations to get better deals.