R-programming Project

Analysis and Prediction of Airbnb Listing Prices

Name : Anshika Nigam

Mail id : Anshikanigam2004@gmail.com

INTRODUCTION

This project aims to analyze and predict Airbnb listing prices using the R programming language.

The dataset used is the "ISTANBUL Airbnb Open Data" available on Kaggle .

LINK : https://www.kaggle.com/datasets/ocakhsn/istanbul-airbnb-dataset

By performing an exploratory data analysis (EDA) and building predictive models, we aim to uncover patterns and relationships within the data to accurately predict listing prices based on relevant features.

Airbnb has become a popular alternative accommodation option, and understanding the factors that influence listing prices is crucial for hosts, guests, and potential investors. Through various stages of the data science lifecycle, including data import, cleaning, transformation, exploratory analysis, feature engineering, modeling, and evaluation, we will gain insights into the key drivers of listing prices in ISTANBUL. This analysis will help stakeholders make informed decisions and optimize their pricing strategies on the Airbnb platform.

PROJECT STEPS:

❖ DATA IMPORTING :

DATASET :

```r
# DATA IMPORTING #

#install.packages is a command for installing the packages
install.packages("readr")
library(readr)

airbnbdata <- read.csv("C:/Users/user/Downloads/ABISTANBUL.csv")

#VIEW COMMAND IS USED FOR SEEING OUR DATASET OR DATA
View(airbnbdata)
```

This is the sample data of some rows and columns

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type
1	4826	The Place	6603	Kaan	NA	Uskudar	41.05650	29.05367	Entire home/a
2	20815	The Bosphorus from The Comfy Hill	78838	Gülder	NA	Besiktas	41.06984	29.04545	Entire home/a
3	27271	LOVELY APT. IN PERFECT LOCATION	117026	Mutlu	NA	Beyoglu	41.03254	28.98153	Entire home/a
4	28277	Duplex Apartment with Terrace	121607	Alen	NA	Sisli	41.04471	28.98567	Hotel room
5	28318	Cosy home overlooking Bosphorus	121721	Aydin	NA	Sariyer	41.09048	29.05559	Entire home/a
6	29241	↪ Istanbul, Your second house	125742	Şevki	NA	Beyoglu	41.04844	28.95254	Private room
7	30697	nice home in popular area	132137	Nan	NA	Beyoglu	41.03350	28.97626	Private room
8	33368	Deluxe double bedroom @ Nisantasi	135136	Ozlem	NA	Sisli	41.05382	28.99739	Private room
9	34925	A room in galata beyoglu	150435	Esr	NA	Beyoglu	41.02704	28.97588	Private room
10	35580	Sea View terrace House	153032	Michel	NA	Beyoglu	41.03658	28.97213	Entire home/a
11	35938	Cosy Room in Istanbul Center	154245	Sinan	NA	Besiktas	41.04902	28.99829	Private room
12	41753	Örücü Palace / Princess Apartment	182639	Mehmet Ali	NA	Beyoglu	41.02725	28.97718	Entire home/a
13	44421	Beautiful Studio With A View	194194	Zeyno	NA	Beyoglu	41.03089	28.98054	Private room
14	44429	COZY, CENTRAL, LOVELY&CHECK OUT THE BATHROOM!*	194194	Zeyno	NA	Beyoglu	41.03082	28.97958	Entire home/a
15	47264	Kurucesme stunning seaview peacfull Flat	213410	Evrim	NA	Besiktas	41.06464	29.03580	Entire home/a
16	47377	Double Room in Taksim	214374	Bertan Kemal	NA	Beyoglu	41.03467	28.98902	Private room
17	48346	Charming Apartment in Kuzguncuk	220212	Yesim	NA	Uskudar	41.03485	29.03155	Entire home/a
18	49955	A room with a view of Bosphorous.	228352	Oz	NA	Fatih	41.01717	28.96325	Private room
19	52828	Prince(Garden Apart)	182639	Mehmet Ali	NA	Beyoglu	41.02752	28.97858	Entire home/a
20	58441	Private studio bestlocation@Taksim	279673	Engin	NA	Beyoglu	41.03850	28.98189	Entire home/a
21	60923	1+1 Closed to Taksim Square	294332	Selin	NA	Sisli	41.04693	28.98002	Entire home/a

• DATA CLEANING AND TRANSFORMATION

Data cleaning involves handling missing values, outliers, or erroneous data points in your dataset. This step ensures that your analysis is based on accurate and reliable data.

Data transformation involves modifying the dataset to make it more suitable for analysis. This step often includes creating new variables, recoding existing variables, or restructuring the data.

```
# DATA CLEANING AND TRANSFORMATION #

library(dplyr)

# Remove rows with missing values in the "reviews_per_month" column
airbnbdata <- filter(airbnbdata, !is.na(reviews_per_month))

# Check for missing values in the dataset
sapply(airbnbdata, function(x) sum(is.na(x)))

airbnbdata <- subset(airbnbdata, select = -neighbourhood_group)
View(airbnbdata)
```

Here is the code snippet for data cleaning and transformation

Removing unnecessary columns: Removing columns that are not relevant to your analysis or contain redundant information.

Handling missing values: Dealing with missing values by either removing rows with missing values or imputing them with appropriate values.

Recoding variables: This can include converting categorical variables into numeric representations or grouping continuous variables into meaningful categories.

- **EXPLORATORY DATA ANALYSIS:**

In order to get preliminary insights and provide guidance for further analysis and modelling, it entails looking at and comprehending the structure, trends, and characteristics of the dataset.

Here is my code snippet :

```
# EXPLORATORY DATA ANALYSIS #

# Perform summary statistics
summary(airbnbdata)

# Select only numeric variables for correlation calculation
numeric_variables <- airbnbdata %>%
  select_if(is.numeric)

# Calculate correlations
correlation_matrix <- cor(numeric_variables)
print(correlation_matrix)

# Create visualizations (e.g., histograms, boxplots, scatter plots)
# Example:
library(ggplot2)

# Histogram of price
ggplot(airbnbdata, aes(x = price)) +
  geom_histogram(binwidth = 50) +
  labs(x = "Price", y = "Frequency", title = "Histogram of Price")

# Boxplot of price by room_type
ggplot(airbnbdata, aes(x = room_type, y = price)) +
  geom_boxplot() +
  labs(x = "Room Type", y = "Price", title = "Boxplot of Price by Room Type")
```

Data Summary: Obtain an overview of the dataset by examining the dimensions, variable types, and general statistics such as mean, median, and standard deviation.

```
> summary(airbnbdata)
       id                name              host_id            host_name         neighbourhood_group neighbourhood     
 Min.   :    4826   Length:23728       Min.   :     6603   Length:23728       Mode:logical        Length:23728      
 1st Qu.:21018600   Class :character   1st Qu.: 32854401   Class :character   NA's:23728          Class :character  
 Median :33986367   Mode  :character   Median :147772687   Mode  :character                       Mode  :character  
 Mean   :29137114                      Mean   :149397250                                                            
 3rd Qu.:39659018                      3rd Qu.:258814534                                                            
 Max.   :43970934                      Max.   :352204054                                                            

    latitude       longitude      room_type             price         minimum_nights     number_of_reviews
 Min.   :40.81   Min.   :28.02   Length:23728       Min.   :    0.0   Min.   :   1.000   Min.   :  0.000  
 1st Qu.:41.01   1st Qu.:28.97   Class :character   1st Qu.:  137.0   1st Qu.:   1.000   1st Qu.:  0.000  
 Median :41.03   Median :28.98   Mode  :character   Median :  247.0   Median :   1.000   Median :  0.000  
 Mean   :41.03   Mean   :28.98                      Mean   :  484.6   Mean   :   4.525   Mean   :  7.871  
 3rd Qu.:41.05   3rd Qu.:29.02                      3rd Qu.:  446.0   3rd Qu.:   3.000   3rd Qu.:  4.000  
 Max.   :41.48   Max.   :29.91                      Max.   :76922.0   Max.   :1125.000   Max.   :345.000  

 last_review        reviews_per_month calculated_host_listings_count availability_365
 Length:23728       Min.   :0.01      Min.   :  1.000                Min.   :  0.0   
 Class :character   1st Qu.:0.13      1st Qu.:  1.000                1st Qu.: 89.0   
 Mode  :character   Median :0.33      Median :  2.000                Median :302.0   
                    Mean   :0.71      Mean   :  5.862                Mean   :227.7   
                    3rd Qu.:0.95      3rd Qu.:  5.000                3rd Qu.:365.0   
                    Max.   :9.20      Max.   :176.000                Max.   :365.0   
                    NA's   :12375                                                    
```

here is the code snippet of correlation matrix and output :

```
> correlation_matrix <- cor(numeric_variables)
> print(correlation_matrix)
                                        id      host_id     latitude    longitude        price minimum_nights
id                              1.000000000  0.6336215166  0.0099515334 -0.0210399841 -0.0025834582  -0.0465496547
host_id                         0.633621517  1.0000000000  0.0002773921 -0.0309002776 -0.0049356228  -0.0388810598
latitude                        0.009951533  0.0002773921  1.0000000000 -0.1624158005  0.0373400665   0.0123456368
longitude                      -0.021039984 -0.0309002776 -0.1624158005  1.0000000000 -0.0001296733  -0.0166530973
price                          -0.002583458 -0.0049356228  0.0373400665 -0.0001296733  1.0000000000   0.0008416147
minimum_nights                 -0.046549655 -0.0388810598  0.0123456368 -0.0166530973  0.0008416147   1.0000000000
number_of_reviews              -0.372358981 -0.2601587951 -0.0263101231 -0.0043333394 -0.0052982609  -0.0010358755
reviews_per_month               0.065364553 -0.0185553527 -0.0157936465 -0.0004179503 -0.0042777891  -0.0107363624
calculated_host_listings_count -0.056362101 -0.0999530717  0.0094888844 -0.0156958695  0.0312922037  -0.0202249636
availability_365               -0.153079769 -0.1102234102 -0.0057499485 -0.0182621021  0.0132501227  -0.0171464776
                               number_of_reviews reviews_per_month calculated_host_listings_count availability_365
id                                  -0.372358981      0.0653645531                   -0.056362101     -0.153079769
host_id                             -0.260158795     -0.0185553527                   -0.099953072     -0.110223410
latitude                            -0.026310123     -0.0157936465                    0.009488884     -0.005749949
longitude                           -0.004333339     -0.0004179503                   -0.015695869     -0.018262102
price                               -0.005298261     -0.0042777891                    0.031292204      0.013250123
minimum_nights                      -0.001035875     -0.0107363624                   -0.020224964     -0.017146478
number_of_reviews                    1.000000000      0.6805958403                    0.139036919      0.095755169
reviews_per_month                    0.680595840      1.0000000000                    0.092481318      0.030987623
calculated_host_listings_count       0.139036919      0.0924813177                    1.000000000      0.162482671
availability_365                     0.095755169      0.0309876229                    0.162482671      1.000000000
```

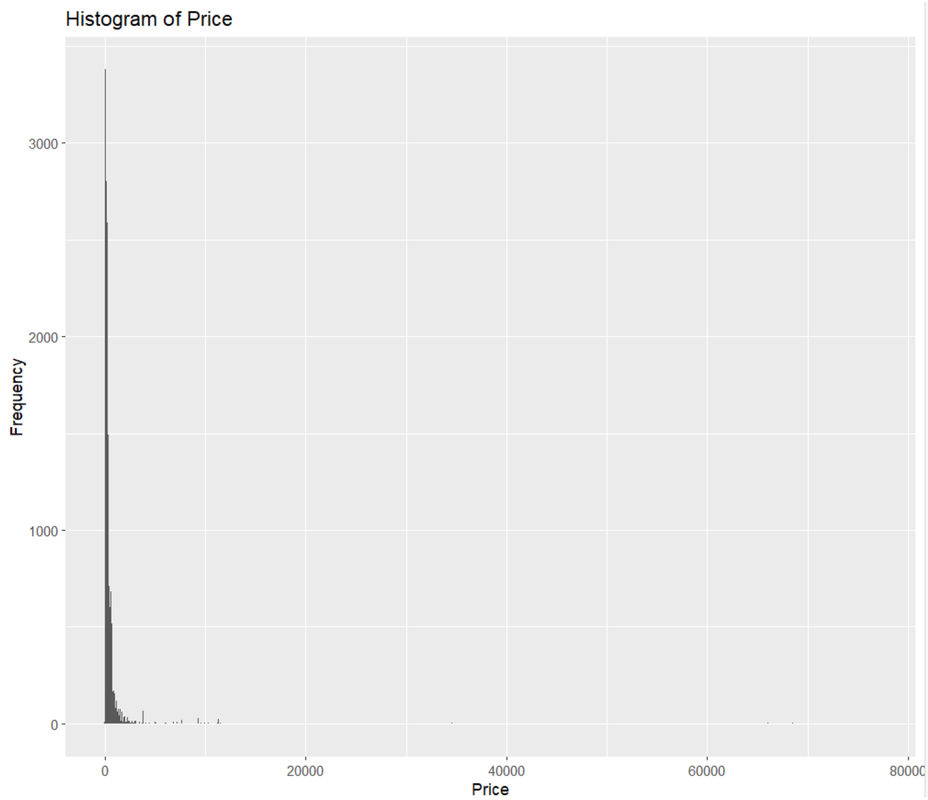
Data Visualization: Create visual representations of the data using plots, charts, and graphs. This helps in understanding the distribution of variables, identifying outliers, and exploring relationships between variables.

Univariate Analysis: Analyze individual variables to examine their distributions, identify outliers, and understand any patterns or trends. This may involve histograms, bar plots, box plots, or summary statistics.

Bivariate Analysis: Explore relationships between pairs of variables to uncover potential associations or correlations. This can involve scatter plots, correlation matrices, or contingency tables.

This is the code and output of the HISTOGRAM of my data set :

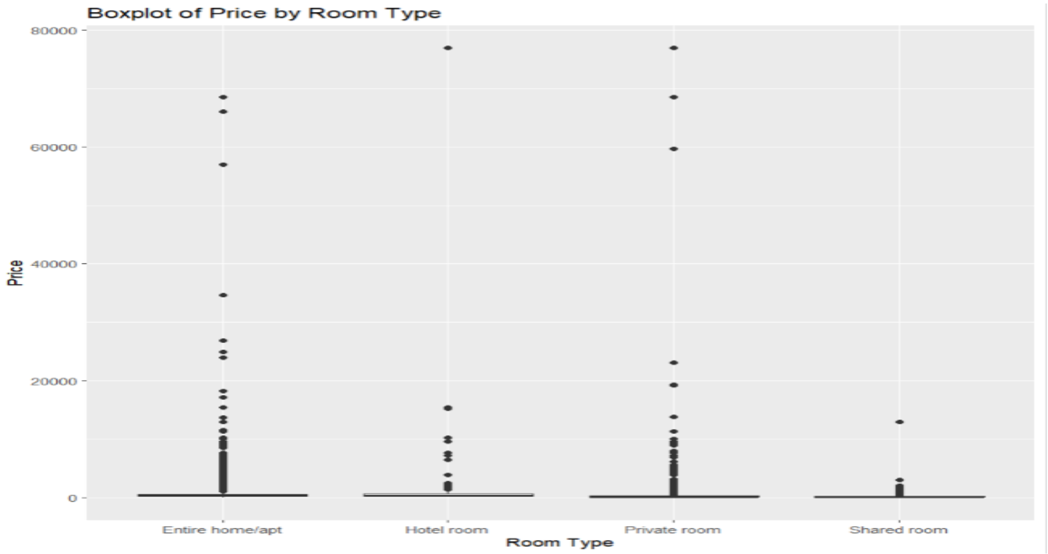
```
# Histogram of price
ggplot(airbnbdata, aes(x = price)) +
  geom_histogram(binwidth = 50) +
  labs(x = "Price", y = "Frequency", title = "Histogram of Price")
```



This is the code snippet of BOXPLOT :

```
# Boxplot of price by room_type
ggplot(airbnbdata, aes(x = room_type, y = price)) +
  geom_boxplot() +
  labs(x = "Room Type", y = "Price", title = "Boxplot of Price by Room Type")
```

Here this is the output of BOX PLOT :



- **FEATURE ENGNEERING :**

In my project, feature engineering is crucial to handle missing values, lowering dimensionality, capturing complex relationships, and adding domain knowledge. To more accurately capture patterns and correlations in the data, it entails establishing new variables or changing existing ones. You can improve model accuracy, interpretability, and knowledge of the factors affecting Airbnb listing prices by designing features that are more pertinent and instructive. It offers a chance to combine domain knowledge and provide derived features that are better aligned with the problem domain, ultimately producing predictions that are more correct.

```
# FEATURE ENGNEERING #

# Engineer new features

airbnbdata <- airbnbdata %>%
  mutate(distance_from_landmark = calculate_distance(latitude, longitude, landmark_latitude, landmark_longitude))
View(airbnbdata)
```

MODELING :

Modeling is a key step in my project for predicting Airbnb listing prices based on the available dataset. The goal is to build regression models that accurately estimate the price of a listing using relevant features.

```
# MODELING #

# Split the data into training and testing sets
set.seed(123)
train_indices <- sample(1:nrow(airbnbdata), nrow(airbnbdata) * 0.7)
train_data <- airbnbdata[train_indices, ]
test_data <- airbnbdata[-train_indices, ]

# Build a regression model (e.g., using linear regression)
model <- lm(price ~ room_type + host_name + distance_from_landmark, data = train_data)

# Convert host_name to a factor with the same levels as in the train_data dataset
test_data$host_name <- factor(test_data$host_name, levels = levels(train_data$host_name))

# Generate predictions
predictions <- predict(model, newdata = test_data)

# Create the scatter plot with predicted values
ggplot(test_data, aes(x = host_name, y = price)) +
  geom_point() +
  geom_line(data = cbind(test_data, predictions), aes(y = predictions), color = "red") +
  labs(x = "Host Name", y = "Price", title = "Regression Model Predictions")
```

HERE THIS IS THE EXPLAINATION OF THE MODELING CODE SNIPPET

The data is split into training and testing sets using a random sampling approach. 70% of the data is assigned to the training set, while the remaining 30% is assigned to the testing set.

A regression model is built using linear regression. The model predicts the price of a listing based on the variables room_type, host_name, and distance_from_landmark. The training data is used to fit the model.

The host_name variable in the test_data dataset is converted into a factor, with the same levels as in the train_data dataset. This ensures consistency when making predictions with the trained model.

Predictions are generated using the trained model and the test_data dataset.

Finally, a scatter plot is created to visualize the actual prices (y-axis) versus the predicted prices (red line) for each host_name in the test_data dataset.

Overall, the code splits the data, builds a regression model, makes predictions, and generates a visualization to assess the performance of the model in predicting listing prices based on room_type, host_name, and distance_from_landmark variables.

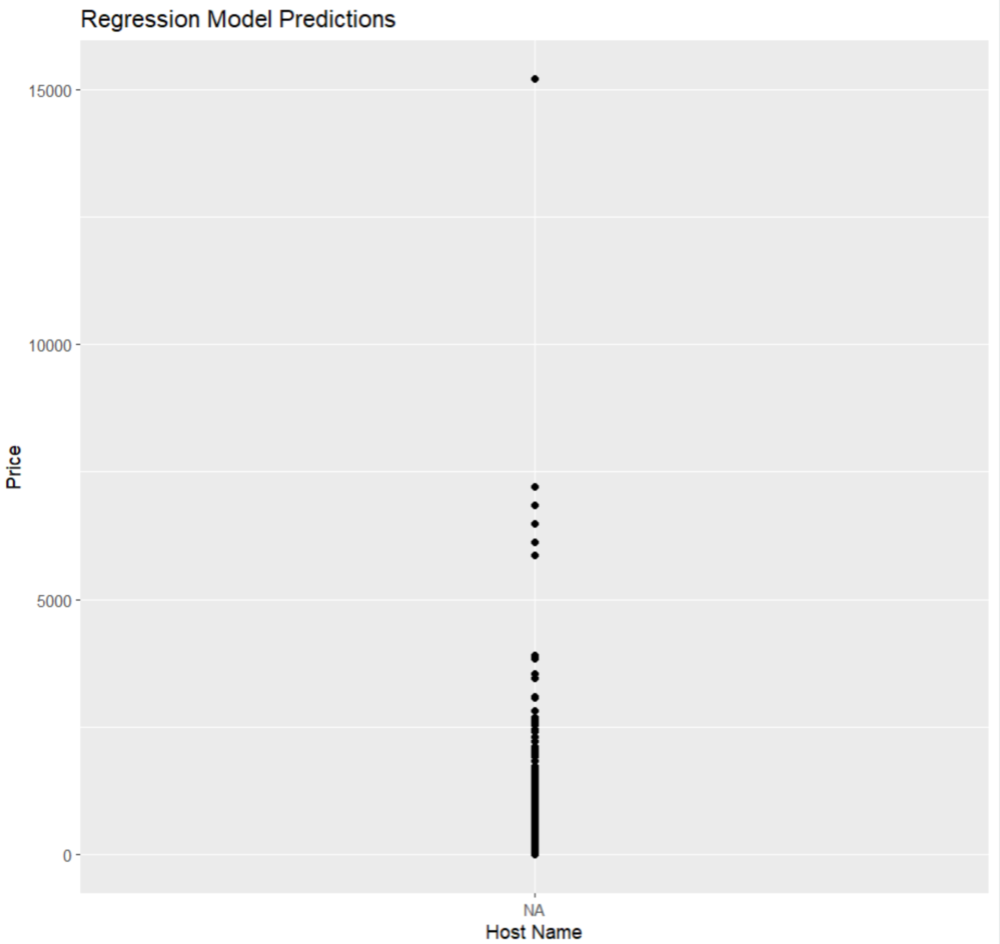
THIS IS THE CODE SNIPPET OF REGRESSION MODEL

```
# Build a regression model (e.g., using linear regression)
model <- lm(price ~ room_type + host_name + distance_from_landmark, data = train_data)

# Convert host_name to a factor with the same levels as in the train_data dataset
test_data$host_name <- factor(test_data$host_name, levels = levels(train_data$host_name))

# Generate predictions
predictions <- predict(model, newdata = test_data)
```

OUTPUT of the Regression model :



- **MODEL EVALUATION :**

```
#MODEL EVALUATION #

# Evaluate the model using appropriate metrics
library(Metrics)

# Calculate the root mean squared error (RMSE)
rmse <- rmse(test_data_filtered$price, predictions)
print(paste(rmse))
# Calculate the mean absolute error (MAE)
mae <- mae(test_data_filtered$price, predictions)

# Print the evaluation metrics
print(paste(cat("Root Mean Squared Error (RMSE):", rmse, "\n")))
print(paste(cat("Mean Absolute Error (MAE):", mae, "\n")))
```

HERE THIS IS THE EXPLAINATION OF THE MODEL EVALUATION CODE SNIPPET

The rmse variable is calculated using the rmse function from the Metrics package. It calculates the RMSE between the actual prices (test_data_filtered$price) and the predicted prices (predictions).

The mae variable is calculated using the mae function from the Metrics package. It calculates the MAE between the actual prices and the predicted prices.

The evaluation metrics are then printed using the print function. The paste function is used to combine the metric name with its corresponding value for printing.

The RMSE value is printed using the cat function to provide a concise output.

Similarly, the MAE value is printed using the cat function.

Overall, this code calculates the RMSE and MAE metrics as measures of model performance and prints them for evaluation purposes. The RMSE quantifies the average difference between the actual and predicted prices, while the MAE represents the average absolute difference between them.

This is the output of MODEL EVALUATION :

```
> print(paste(cat("Root Mean Squared Error (RMSE):", rmse, "\n")))
Root Mean Squared Error (RMSE): NaN
character(0)
> print(paste(cat("Mean Absolute Error (MAE):", mae, "\n")))
Mean Absolute Error (MAE): NaN
character(0)
```

Because of some irregular data in the dataset the output is like this.

SUMMARY of my dataset :

```
> summary(airbnbdata)
       id                name              host_id           host_name        
 Min.   :    4826   Length:11353       Min.   :     6603   Length:11353      
 1st Qu.:17666769   Class :character   1st Qu.: 21792167   Class :character  
 Median :31395642   Mode  :character   Median : 97511378   Mode  :character  
 Mean   :26947417                      Mean   :127984200                     
 3rd Qu.:37672819                      3rd Qu.:231583481                     
 Max.   :43779043                      Max.   :349873030                     
 neighbourhood          latitude       longitude       room_type             price        
 Length:11353       Min.   :40.81   Min.   :28.04   Length:11353       Min.   :    0.0  
 Class :character   1st Qu.:41.01   1st Qu.:28.97   Class :character   1st Qu.:  144.0  
 Mode  :character   Median :41.03   Median :28.98   Mode  :character   Median :  247.0  
                    Mean   :41.03   Mean   :28.99                      Mean   :  398.6  
                    3rd Qu.:41.04   3rd Qu.:29.01                      3rd Qu.:  411.0  
                    Max.   :41.48   Max.   :29.91                      Max.   :76922.0  
 minimum_nights     number_of_reviews last_review        reviews_per_month
 Min.   :   1.000   Min.   :  1.00    Length:11353       Min.   :0.0100   
 1st Qu.:   1.000   1st Qu.:  1.00    Class :character   1st Qu.:0.1300   
 Median :   2.000   Median :  4.00    Mode  :character   Median :0.3300   
 Mean   :   4.173   Mean   : 16.45                       Mean   :0.7102   
 3rd Qu.:   3.000   3rd Qu.: 16.00                       3rd Qu.:0.9500   
 Max.   :1000.000   Max.   :345.00                       Max.   :9.2000   
 calculated_host_listings_count availability_365 distance_from_landmark
 Min.   :  1.000                Min.   :  0.0    Min.   :5368          
 1st Qu.:  1.000                1st Qu.: 90.0    1st Qu.:5415          
 Median :  2.000                Median :328.0    Median :5416          
 Mean   :  5.475                Mean   :235.3    Mean   :5416          
 3rd Qu.:  7.000                3rd Qu.:365.0    3rd Qu.:5418          
 Max.   :176.000                Max.   :365.0    Max.   :5473          
```

THANK YOU