R Notebook

Code

Hide

```
library(dplyr)
library(tidyverse)
library(readr)
library(ggplot2)
```

Hide

```
#Loading the data for exploratory analysis
attach(airbnb)
```

Hide

```
head(airbnb)
```

	neighbourhood_group <chr>	neighbourhood <chr>	latitude <dbl>	longitude <dbl>	room_type <chr>	price <int>
1	Brooklyn	Kensington	40.64749	-73.97237	Private room	149
2	Manhattan	Midtown	40.75362	-73.98377	Entire home/apt	225
3	Brooklyn	Clinton Hill	40.68514	-73.95976	Entire home/apt	89
4	Manhattan	East Harlem	40.79851	-73.94399	Entire home/apt	80
5	Manhattan	Murray Hill	40.74767	-73.97500	Entire home/apt	200
6	Brooklyn	Bedford-Stuyvesant	40.68688	-73.95596	Private room	60

6 rows | 1-8 of 11 columns

Hide

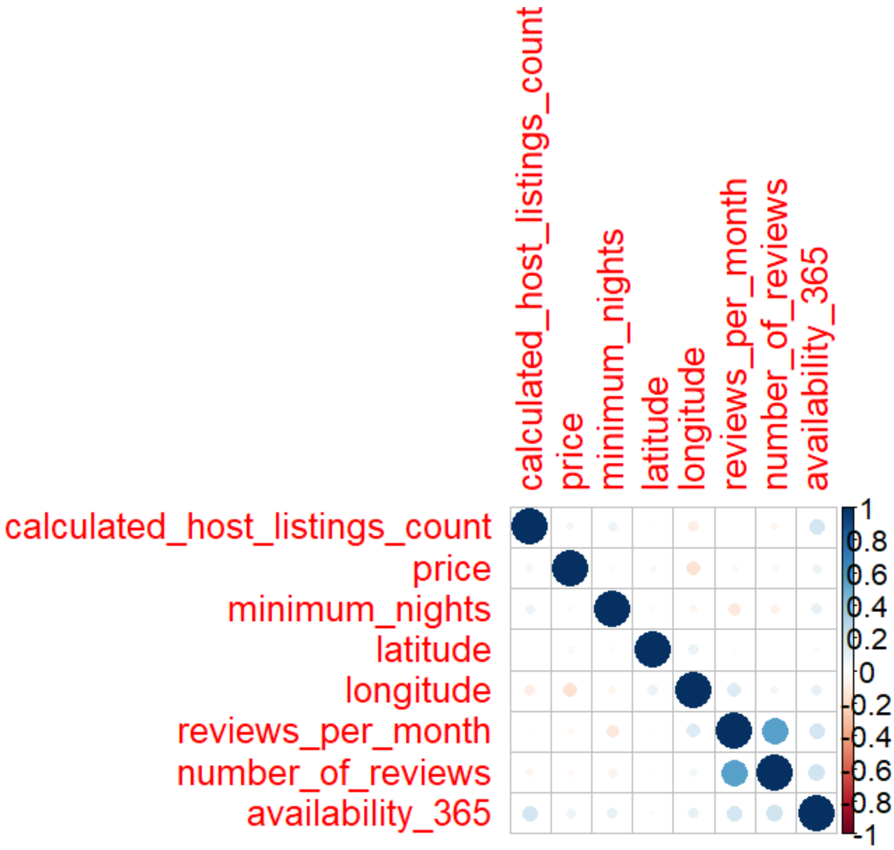
```
#2. Calculate the correlation between the different attributes (include the figure produced b
y R in your answer).

library(correlation)
correlation::correlation(airbnb,include_factors = FALSE, method = 'pearson')
```

```
# Correlation Matrix (pearson-method)

Parameter1                     |                       Parameter2 |         r |         95% CI
| t(38819) |         p
--------------------------------------------------------------------------------------------------
---------------------
latitude                       |                        longitude |      0.09 | [ 0.08,  0.10]
|    17.46 | < .001***
latitude                       |                            price |      0.03 | [ 0.02,  0.04]
|     6.18 | < .001***
latitude                       |                   minimum_nights |      0.02 | [ 0.01,  0.03]
|     4.91 | < .001***
latitude                       |                number_of_reviews | -8.56e-03 | [-0.02,  0.00]
|    -1.69 | 0.189
latitude                       |                reviews_per_month |     -0.01 | [-0.02,  0.00]
|    -1.99 | 0.185
latitude                       | calculated_host_listings_count |  4.34e-03 | [-0.01,  0.01]
|     0.86 | 0.393
latitude                       |                 availability_365 |     -0.02 | [-0.03, -0.01]
|    -4.32 | < .001***
longitude                      |                            price |     -0.16 | [-0.16, -0.15]
|   -30.97 | < .001***
longitude                      |                   minimum_nights |     -0.06 | [-0.07, -0.05]
|   -10.93 | < .001***
longitude                      |                number_of_reviews |      0.05 | [ 0.04,  0.06]
|    10.80 | < .001***
longitude                      |                reviews_per_month |      0.15 | [ 0.14,  0.16]
|    29.12 | < .001***
longitude                      | calculated_host_listings_count |     -0.09 | [-0.10, -0.08]
|   -18.47 | < .001***
longitude                      |                 availability_365 |      0.10 | [ 0.09,  0.11]
|    20.32 | < .001***
price                          |                   minimum_nights |      0.03 | [ 0.02,  0.04]
|     5.03 | < .001***
price                          |                number_of_reviews |     -0.04 | [-0.05, -0.03]
|    -7.08 | < .001***
price                          |                reviews_per_month |     -0.03 | [-0.04, -0.02]
|    -6.04 | < .001***
price                          | calculated_host_listings_count |      0.05 | [ 0.04,  0.06]
|    10.44 | < .001***
price                          |                 availability_365 |      0.08 | [ 0.07,  0.09]
|    15.47 | < .001***
minimum_nights                 |                number_of_reviews |     -0.07 | [-0.08, -0.06]
|   -13.70 | < .001***
minimum_nights                 |                reviews_per_month |     -0.12 | [-0.13, -0.11]
|   -24.16 | < .001***
minimum_nights                 | calculated_host_listings_count |      0.07 | [ 0.06,  0.08]
|    14.52 | < .001***
minimum_nights                 |                 availability_365 |      0.10 | [ 0.09,  0.11]
|    20.13 | < .001***
number_of_reviews              |                reviews_per_month |      0.55 | [ 0.54,  0.56]
|   129.65 | < .001***
number_of_reviews              | calculated_host_listings_count |     -0.06 | [-0.07, -0.05]
|   -11.80 | < .001***
number_of_reviews              |                 availability_365 |      0.19 | [ 0.18,  0.20]
```

```
|     38.84 | < .001***
reviews_per_month              | calculated_host_listings_count | -9.44e-03 | [-0.02,  0.00]
|     -1.86 | 0.189
reviews_per_month              |                availability_365 |      0.19 | [ 0.18,  0.20]
|     37.28 | < .001***
calculated_host_listings_count |                availability_365 |      0.18 | [ 0.17,  0.19]
|     36.67 | < .001***

p-value adjustment method: Holm (1979)
Observations: 38821
```

Hide

```
# Select only the numeric columns
numeric_cols <- sapply(airbnb, is.numeric)
airbnb_numeric <- airbnb[, numeric_cols]

# Compute the correlation matrix using the selected columns
corr_matrix <- cor(airbnb_numeric, method = "pearson")

# Plot the correlation matrix using corrplot
library(corrplot)
corrplot(corr_matrix, order = "AOE")
```

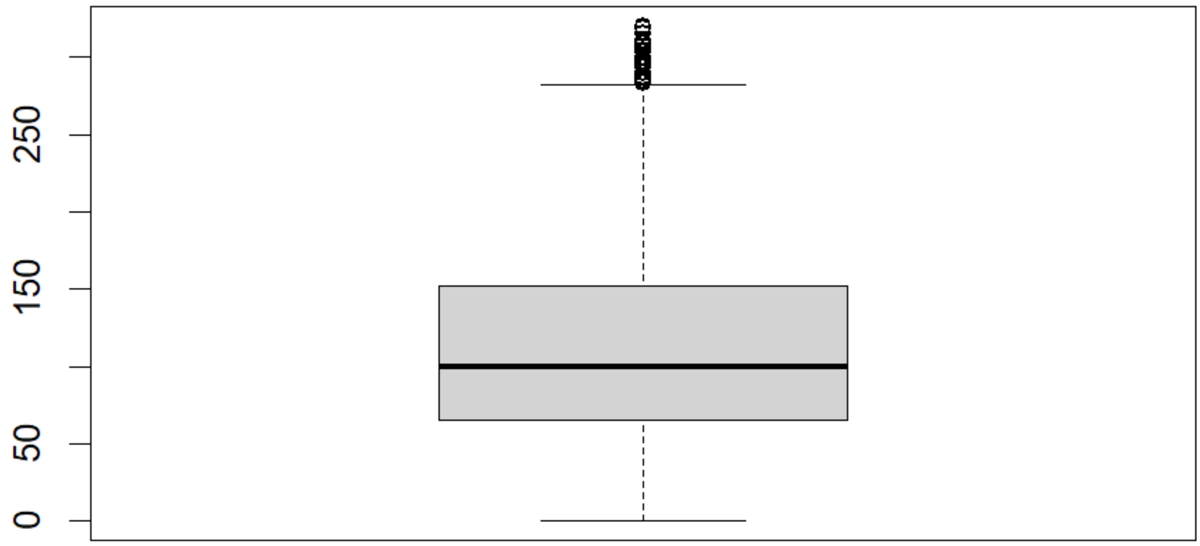


Hide

```
#Remove the outliers
airbnb <-airbnb[-which(airbnb$price%in%boxplot.stats(airbnb$price)$out),]
airbnb <-airbnb[-which(airbnb$number_of_reviews%in%boxplot.stats(airbnb$number_of_reviews)$ou
t),]
airbnb <-airbnb[-which(airbnb$reviews_per_month%in%boxplot.stats(airbnb$reviews_per_month)$ou
t),]
airbnb <-airbnb[-which(airbnb$minimum_nights%in%boxplot.stats(airbnb$minimum_nights)$out),]
```

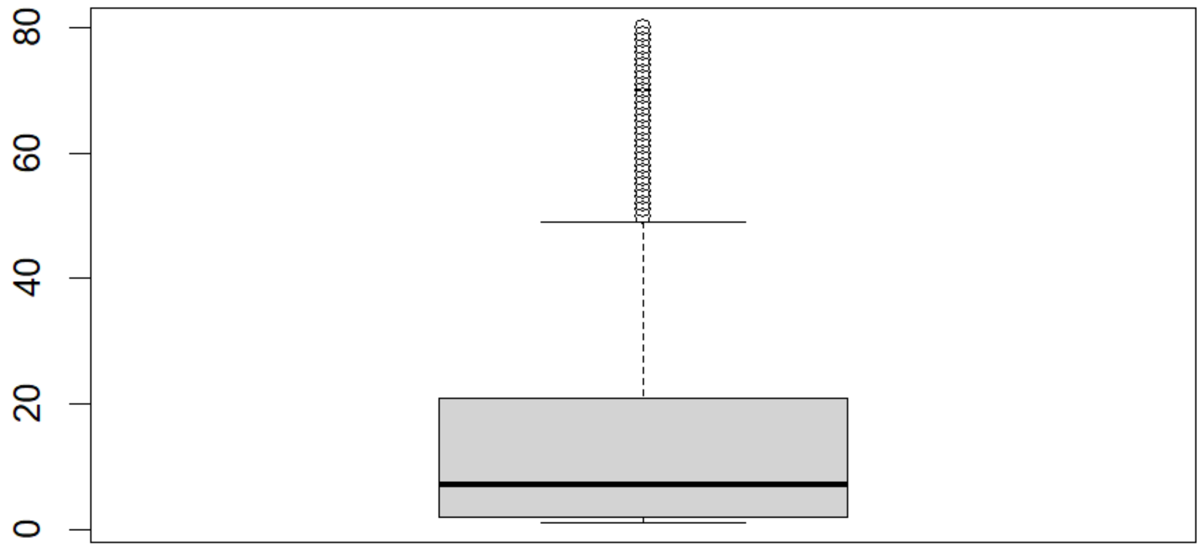
Hide

```
#Check again using boxplot
boxplot(airbnb$price)
```



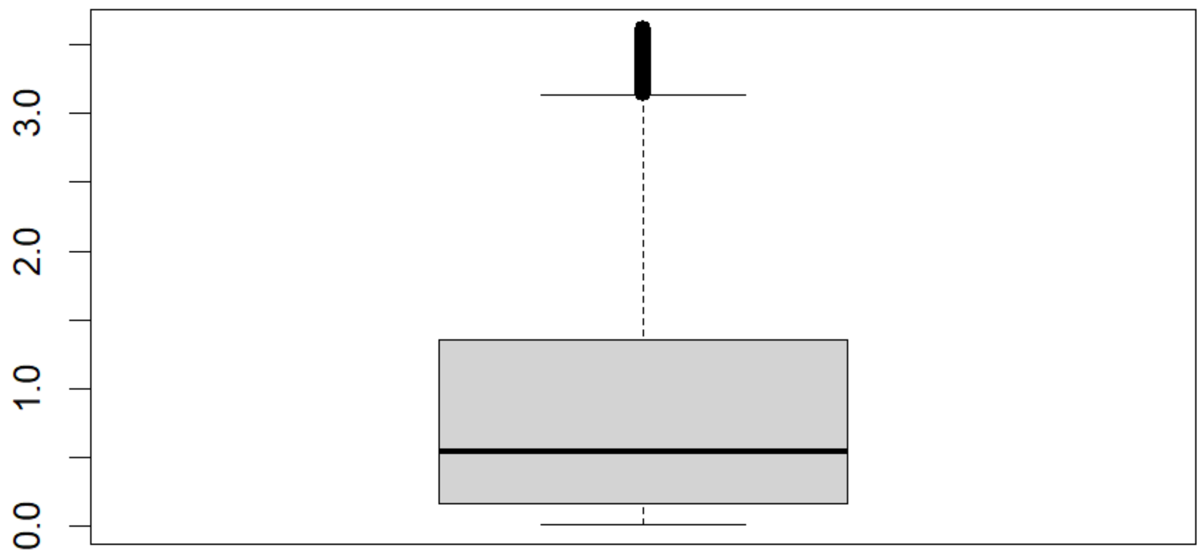
Hide

```
boxplot(airbnb$number_of_reviews)
```



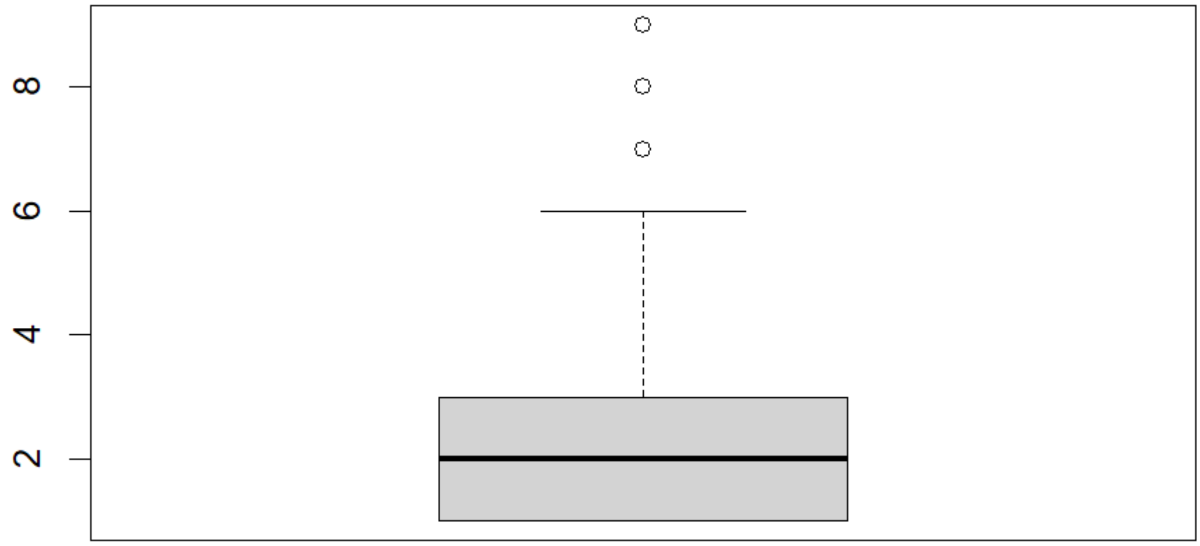
Hide

```
boxplot(airbnb$reviews_per_month)
```



Hide

```
boxplot(airbnb$minimum_nights)
```



Hide

```
#3 Data Analysis


############Exploratory Data##################
property_data <-  airbnb %>%
  group_by(neighbourhood_group, room_type) %>%
  summarize(Freq = n())
```

```
`summarise()` has grouped output by 'neighbourhood_group'. You can override using the `.group
s` argument.
```

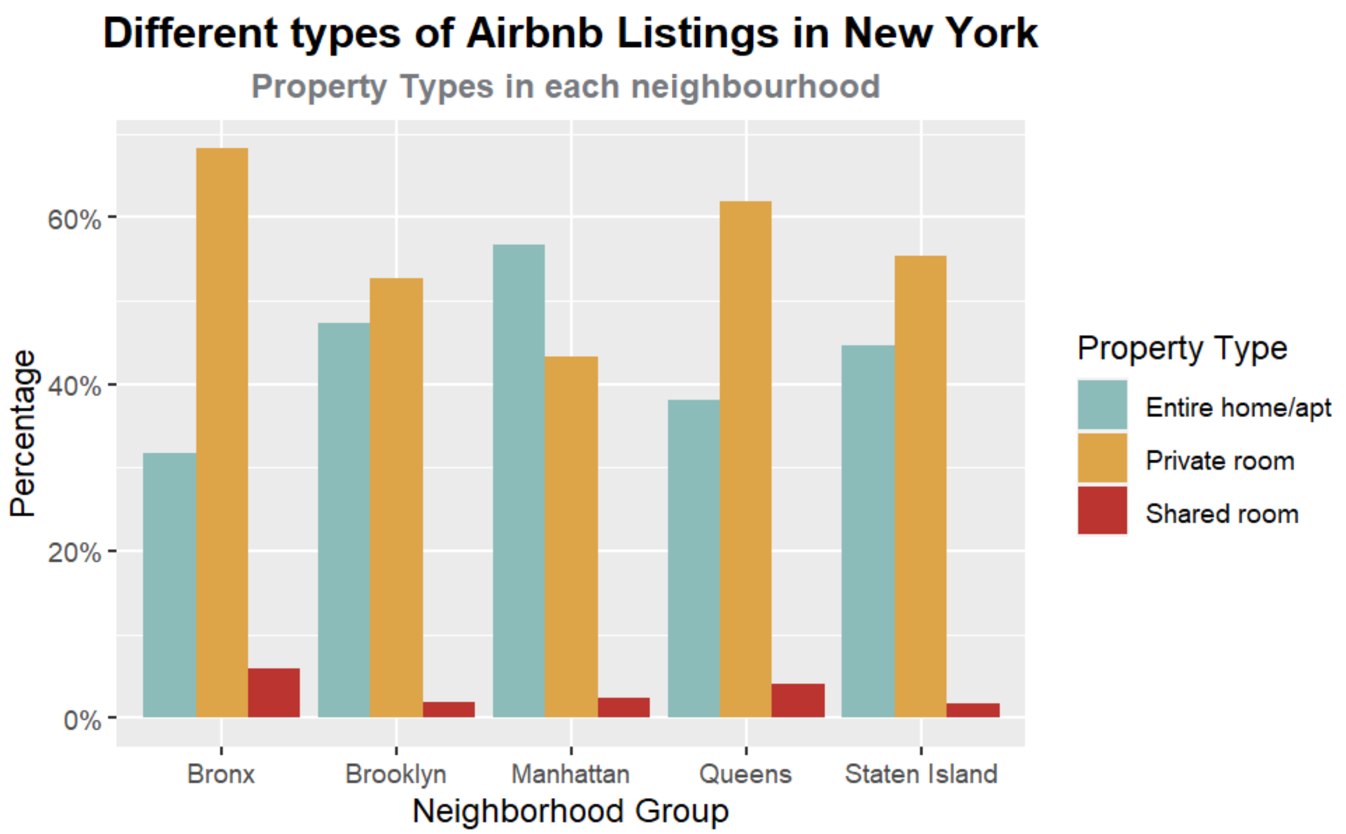
Hide

```
property_types <-  airbnb %>%
  filter(room_type %in% c("Private room","Entire home/apt","Entire home/apt")) %>%
  group_by(neighbourhood_group) %>%
  summarize(sum = n())

ratio_property <- merge (property_data, property_types, by="neighbourhood_group")

ratio_property <- ratio_property %>%
  mutate(ratio = Freq/sum)
```

Hide

```
#Different types of Airbnb Listings in New York

ggplot(ratio_property, aes(x=neighbourhood_group, y = ratio, fill = room_type)) +
  geom_bar(position = "dodge", stat="identity") +
  xlab("Borough") + ylab ("Count") +
  scale_fill_discrete(name = "Property Type") +
  scale_y_continuous(labels = scales::percent) +
  ggtitle("Different types of Airbnb Listings in New York",
          subtitle = "Property Types in each neighbourhood ") +
  theme(plot.title = element_text(face = "bold", size = 14, hjust = 0.5, color = "black") ) +
  theme(plot.subtitle = element_text(face = "bold", color = "#777A7F", hjust = 0.5)) +
  theme(plot.caption = element_text(color = "black"))+scale_color_gradient(low="#d3cbcb", hig
h="#852eaa")+
  scale_fill_manual("Property Type", values=c("#8CBCB9","#dda448", "#bb342f", "#ede7e3", "#ff
a62b")) +
  xlab("Neighborhood Group") + ylab("Percentage")
```

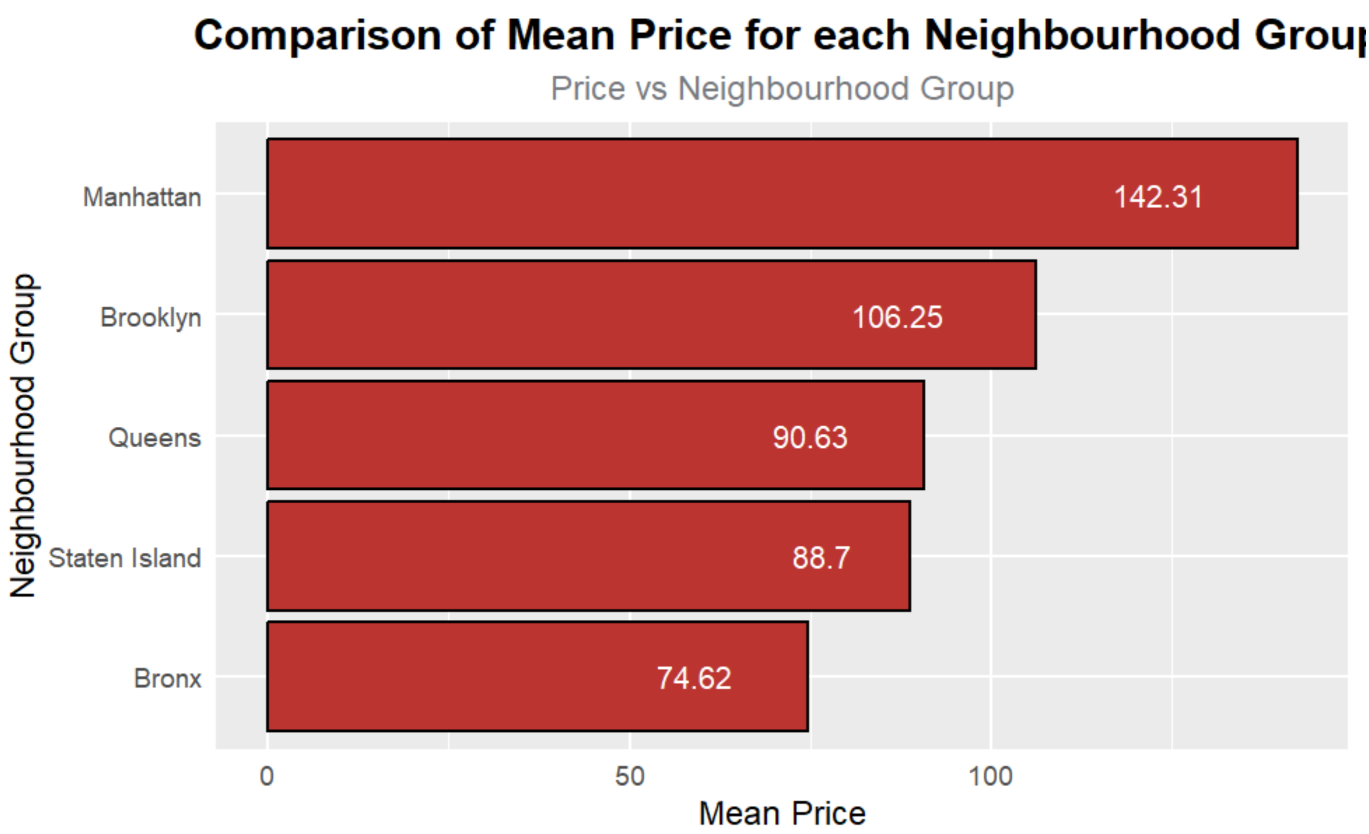
```
Scale for fill is already present.
Adding another scale for fill, which will replace the existing scale.
```



Hide

```
#Comparison of Mean Price for each Neighbourhood Group
airbnb %>%
  filter(!(is.na(neighbourhood_group))) %>%
  filter(!(neighbourhood_group == "Unknown")) %>%
  group_by(neighbourhood_group) %>%
  summarise(mean_price = mean(price, na.rm = TRUE)) %>%
  ggplot(aes(x = reorder(neighbourhood_group, mean_price), y = mean_price, fill = neighbourho
od_group)) +
  geom_col(stat ="identity", color = "black", fill="#bb342f") +
  coord_flip() +
  theme_gray() +
  labs(x = "Neighbourhood Group", y = "Price") +
  geom_text(aes(label = round(mean_price,digit = 2)), hjust = 2.0, color = "white", size = 3.
5) +
  ggtitle(" Comparison of Mean Price for each Neighbourhood Group", subtitle = "Price vs Neig
hbourhood Group") +
  xlab("Neighbourhood Group") +
  ylab("Mean Price") +
  theme(legend.position = "none",
        plot.title = element_text(color = "black", size = 14, face = "bold", hjust = 0.5),
        plot.subtitle = element_text(color = "#777A7F", hjust = 0.5),
        axis.title.y = element_text(),
        axis.title.x = element_text(),
        axis.ticks = element_blank())
```

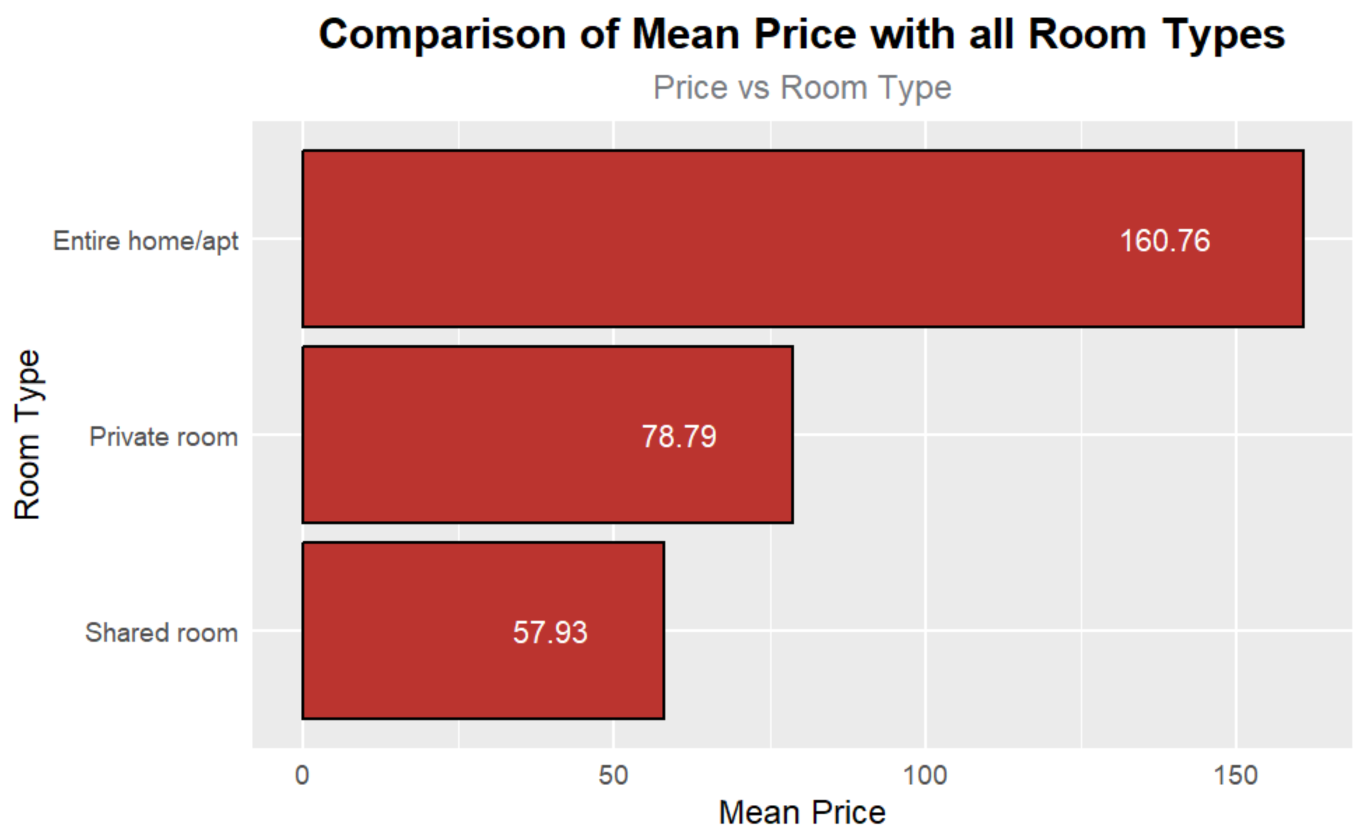
```
Warning: Ignoring unknown parameters: `stat`
```



Hide

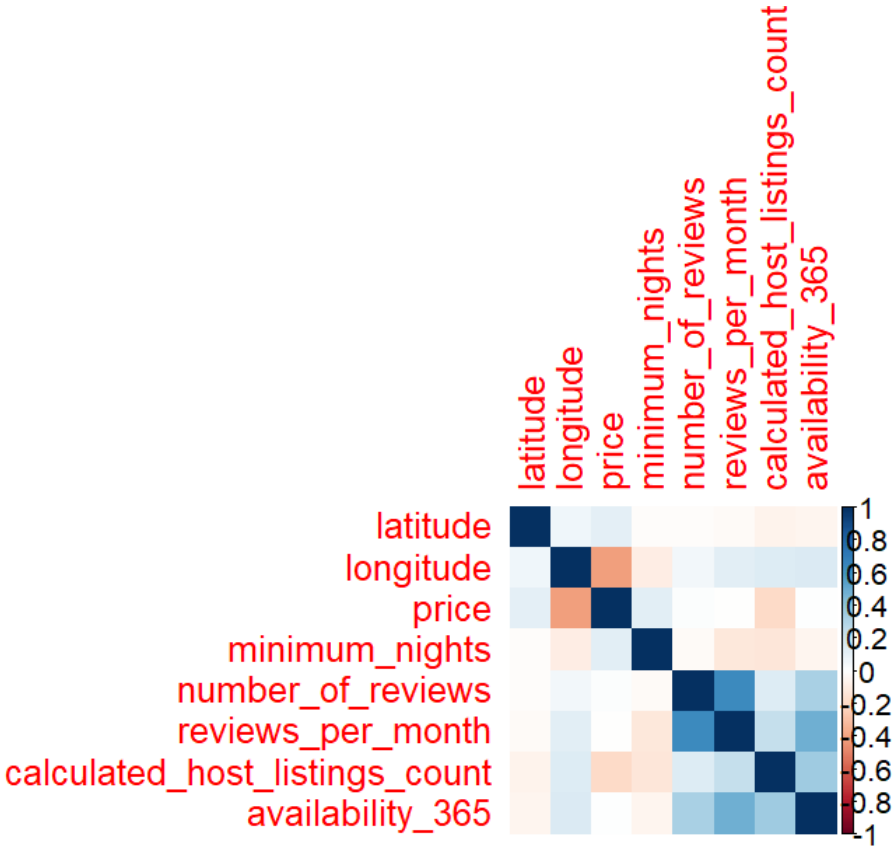
```
#Comparison of Mean Price with all Room Types
airbnb %>%
  filter(!(is.na(room_type))) %>%
  filter(!(room_type == "Unknown")) %>%
  group_by(room_type) %>%
  summarise(mean_price = mean(price, na.rm = TRUE)) %>%
  ggplot(aes(x = reorder(room_type, mean_price), y = mean_price, fill = room_type)) +
  geom_col(stat ="identity", color = "black", fill="#bb342f") +
  coord_flip() +
  theme_gray() +
  labs(x = "Room Type", y = "Price") +
  geom_text(aes(label = round(mean_price,digit = 2)), hjust = 2.0, color = "white", size = 3.
5) +
  ggtitle("Comparison of Mean Price with all Room Types", subtitle = "Price vs Room Type") +
  xlab("Room Type") +
  ylab("Mean Price") +
  theme(legend.position = "none",
        plot.title = element_text(color = "black", size = 14, face = "bold", hjust = 0.5),
        plot.subtitle = element_text(color = "#777A7F", hjust = 0.5),
        axis.title.y = element_text(),
        axis.title.x = element_text(),
        axis.ticks = element_blank())
```

```
Warning: Ignoring unknown parameters: `stat`
```



Hide

```
#Checking the correlation matrix again
library("corrplot")
airbnb_cor <- airbnb[, sapply(airbnb, is.numeric)]
airbnb_cor <- airbnb_cor[complete.cases(airbnb_cor), ]
correlation_matrix <- cor(airbnb_cor, method = "spearman")
corrplot(correlation_matrix, method = "color")
```

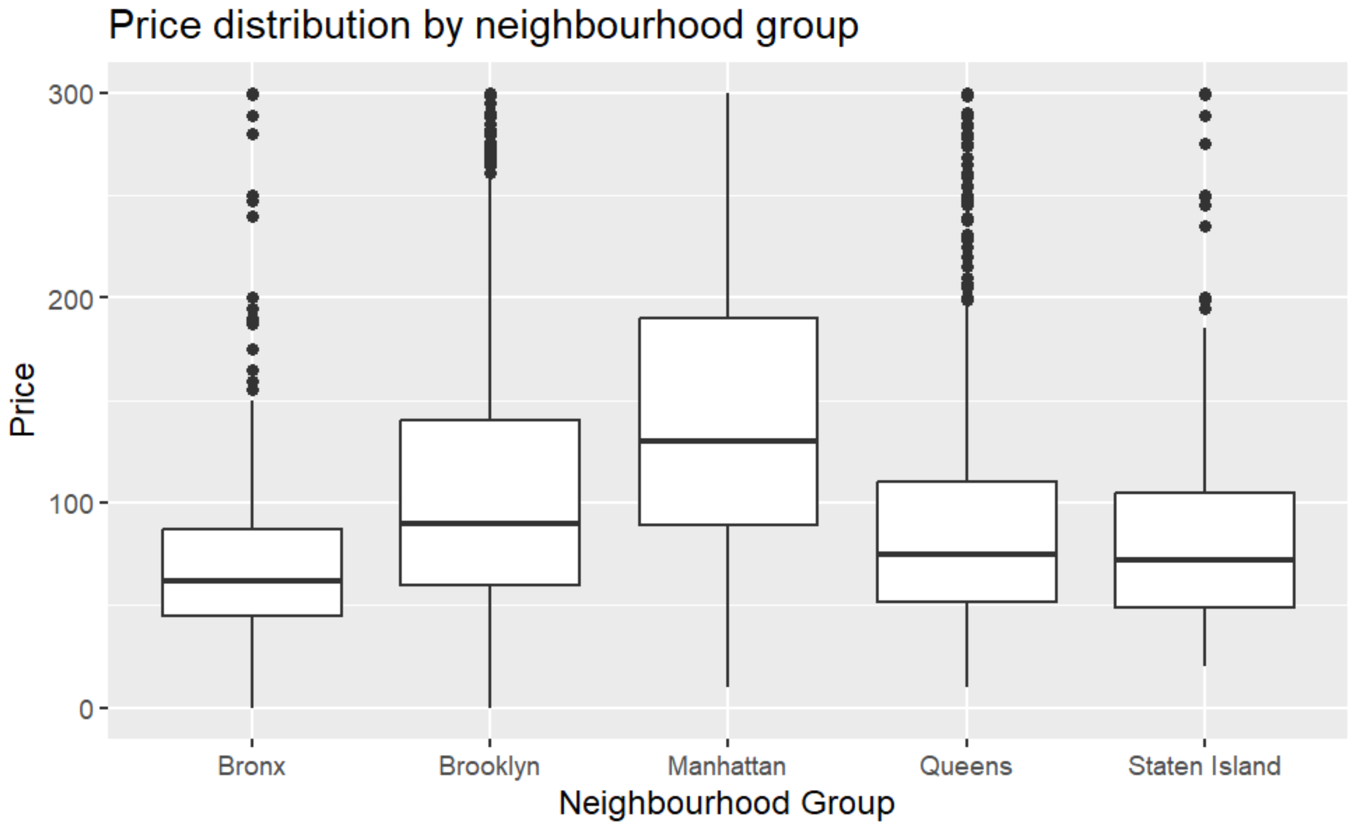


Hide

```
# Convert categorical variables to factors
airbnb$neighbourhood_group <- as.factor(airbnb$neighbourhood_group)
airbnb$neighbourhood <- as.factor(airbnb$neighbourhood)
airbnb$room_type <- as.factor(airbnb$room_type)

# Remove outliers
airbnb <- airbnb[airbnb$price <= quantile(airbnb$price, 0.99),]
airbnb <- airbnb[airbnb$minimum_nights <= quantile(airbnb$minimum_nights, 0.99),]
```

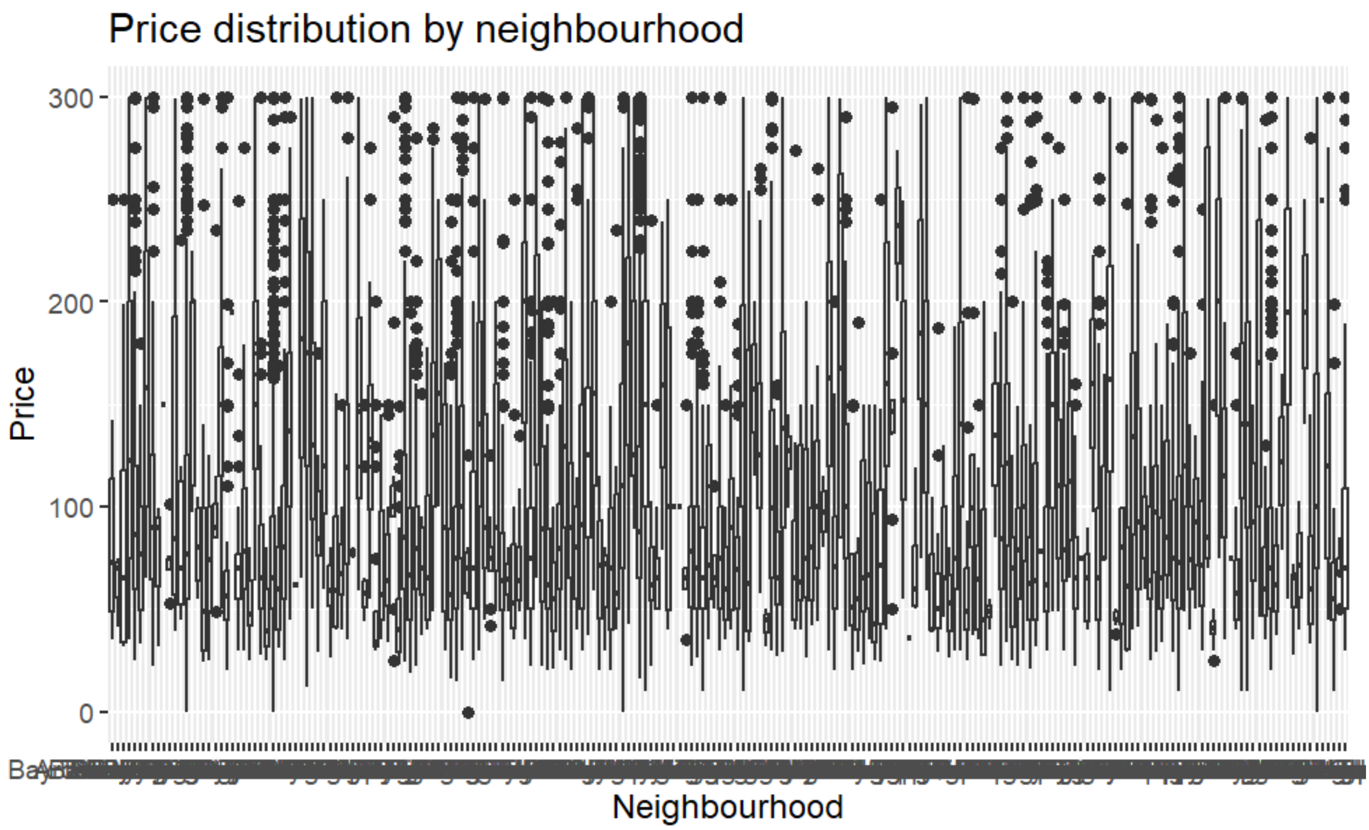
Hide

```
# Price distribution by neighbourhood group
ggplot(airbnb, aes(x = neighbourhood_group, y = price)) +
  geom_boxplot() +
  labs(x = "Neighbourhood Group", y = "Price") +
  ggtitle("Price distribution by neighbourhood group")
```



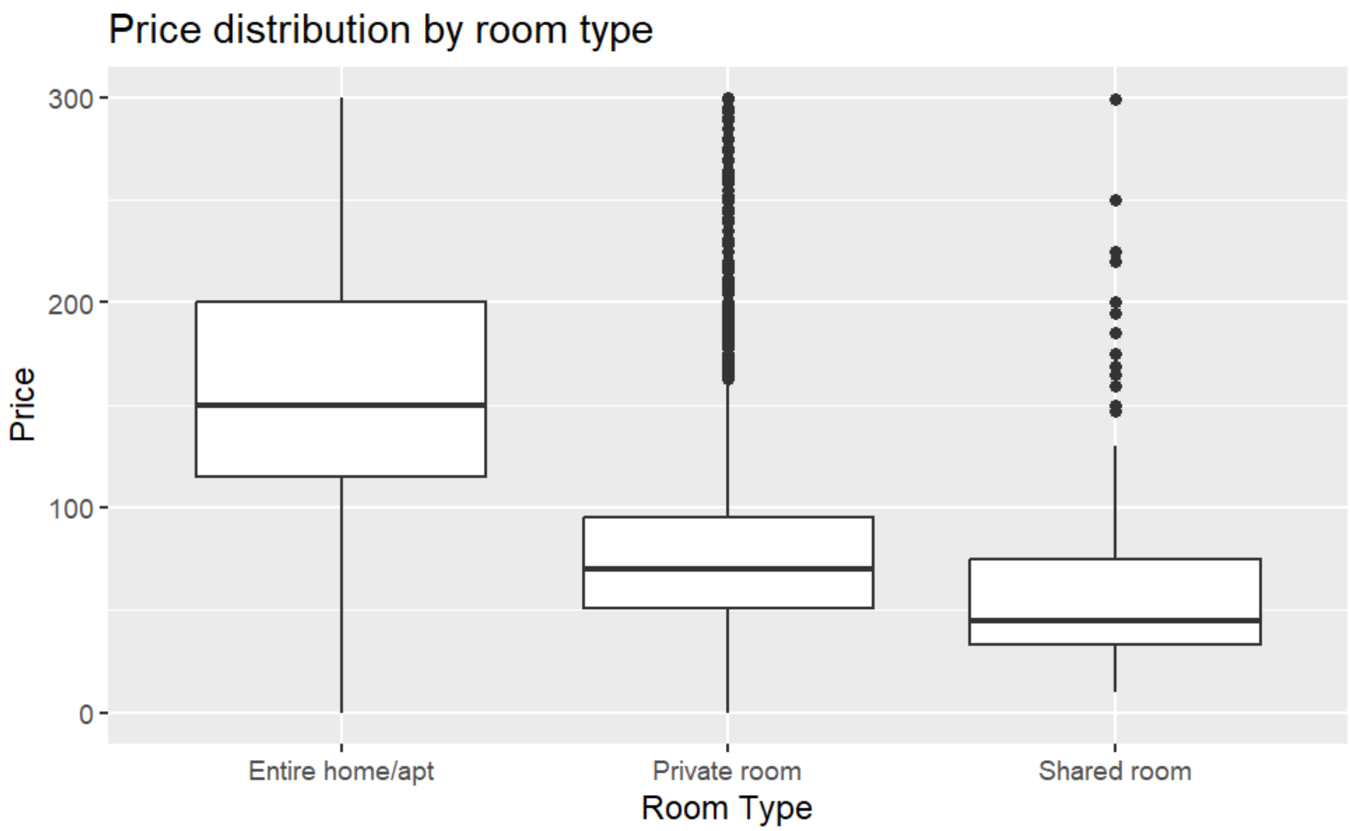
Hide

```
# Price distribution by neighbourhood
ggplot(airbnb, aes(x = neighbourhood, y = price)) +
  geom_boxplot() +
  labs(x = "Neighbourhood", y = "Price") +
  ggtitle("Price distribution by neighbourhood")
```



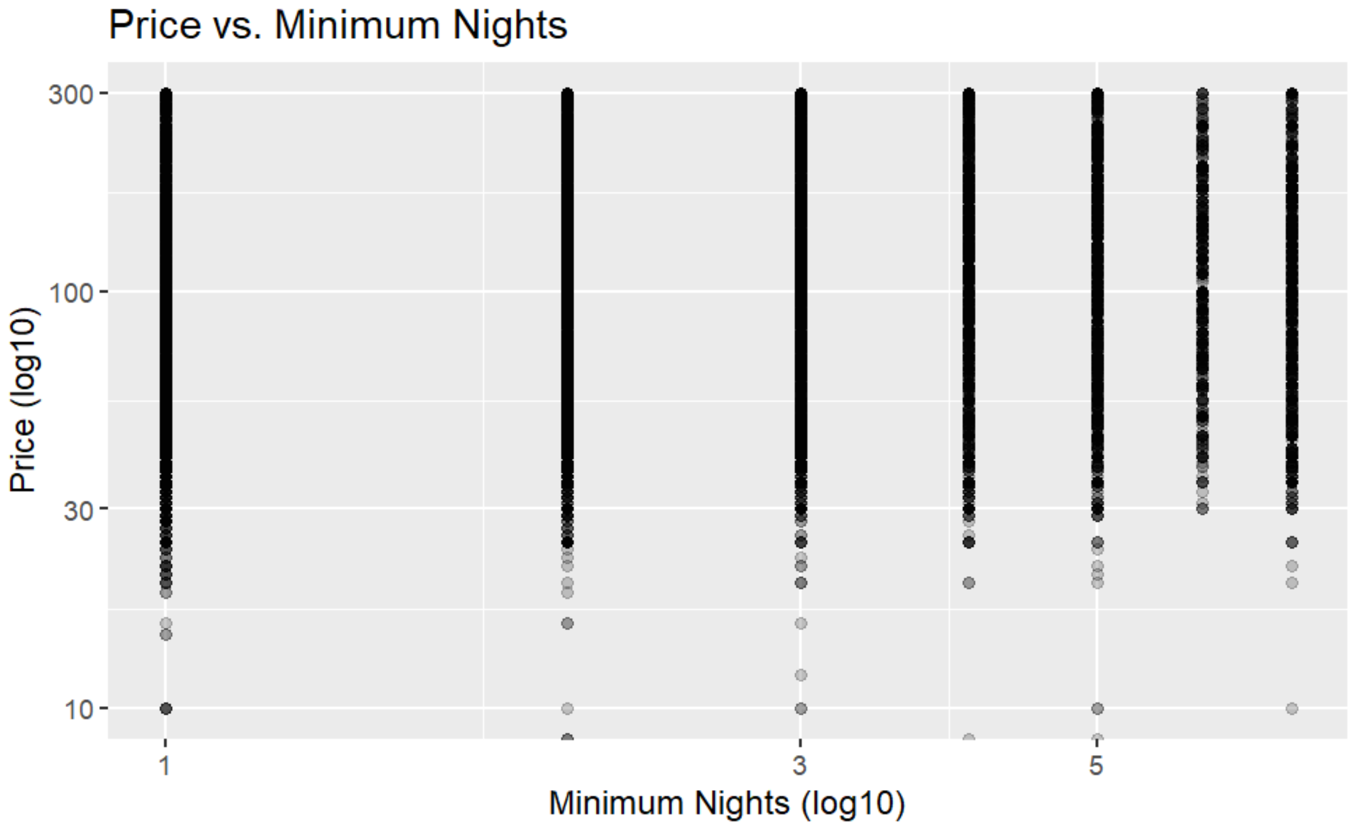
Hide

```
# Price distribution by room type
ggplot(airbnb, aes(x = room_type, y = price)) +
  geom_boxplot() +
  labs(x = "Room Type", y = "Price") +
  ggtitle("Price distribution by room type")
```



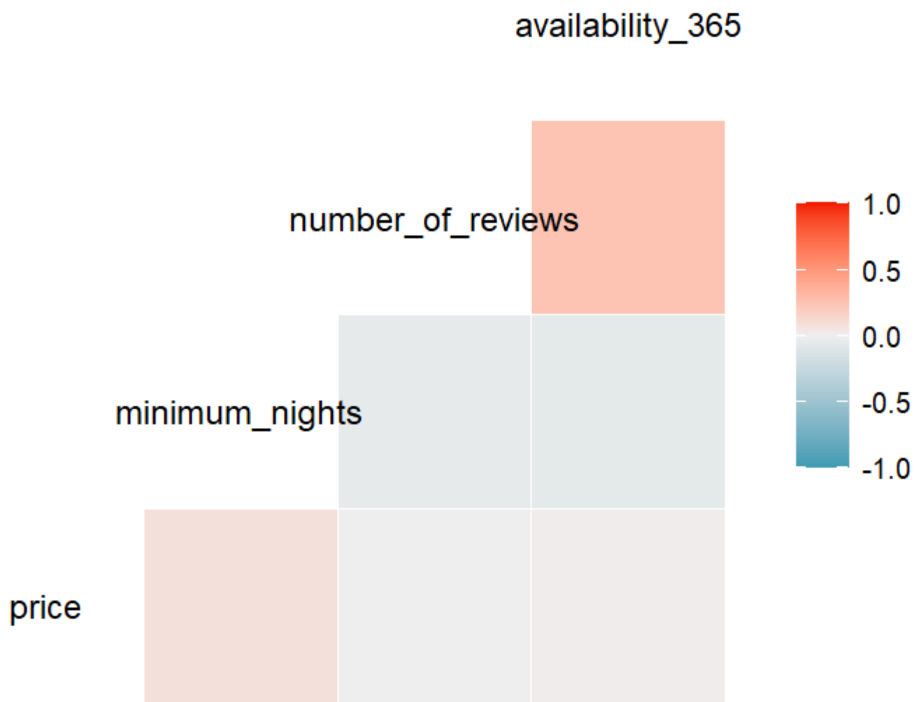
Hide

```
# Scatter plot of price vs. minimum nights
ggplot(airbnb, aes(x = minimum_nights, y = price)) +
  geom_point(alpha = 0.2) +
  scale_x_continuous(trans = "log10") +
  scale_y_continuous(trans = "log10") +
  labs(x = "Minimum Nights (log10)", y = "Price (log10)") +
  ggtitle("Price vs. Minimum Nights")
```



Hide

```
# Correlation heatmap
library(GGally)
ggcorr(airbnb[, c("price", "minimum_nights", "number_of_reviews", "availability_365")])
```



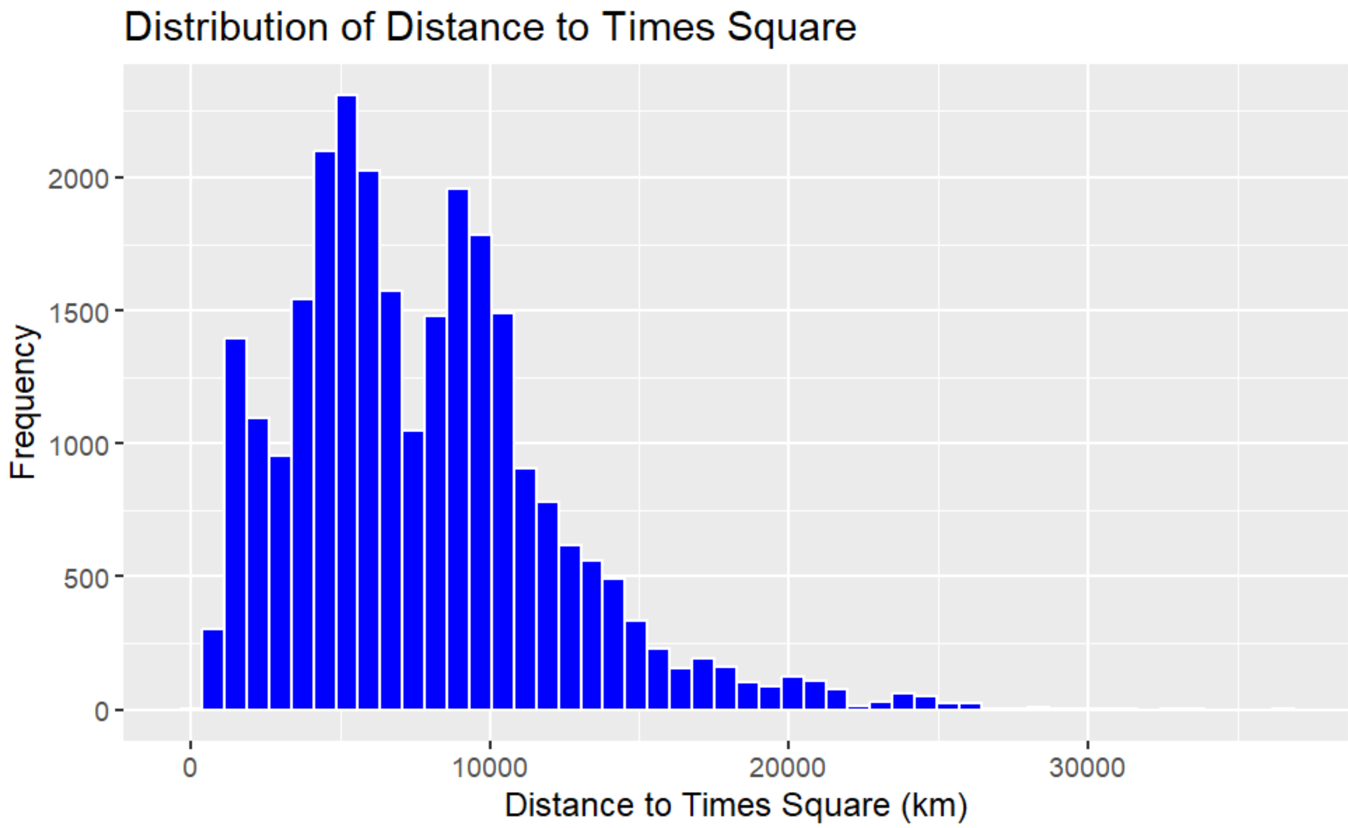
Hide

```
#Information collected:
# From the exploratory data analysis, we can observe that:
#
# Manhattan is the most expensive neighbourhood group, with prices significantly higher than
the other neighbourhood groups.
# Within Manhattan, the neighbourhoods of SoHo, Tribeca, and West Village have the highest pr
ices.
# Entire homes/apartments are the most expensive room type, followed by private rooms and sha
red rooms.
# There is a weak positive correlation between price and minimum nights, as well as price and
availability.


#perform some feature engineering to create new features that may be useful for our analysis:
# Create a new feature for distance to Times Square

library(geodist)  # load the geosphere package
ts_lat <-airbnb %>%
  filter(neighbourhood_group == "Manhattan") %>%
  summarize(lat = mean(latitude), long = mean(longitude))

airbnb <- airbnb %>%
  mutate(dist_ts = geodist(cbind(longitude, latitude), c(ts_lat$long, ts_lat$lat)))
```

```
object has no named columns; assuming order is lon then lat
```

Hide

```
#View the distribution of the new feature
ggplot(airbnb, aes(x = dist_ts)) +
  geom_histogram(bins = 50, fill = "blue", color = "white") +
  ggtitle("Distribution of Distance to Times Square") +
  xlab("Distance to Times Square (km)") +
  ylab("Frequency")
```



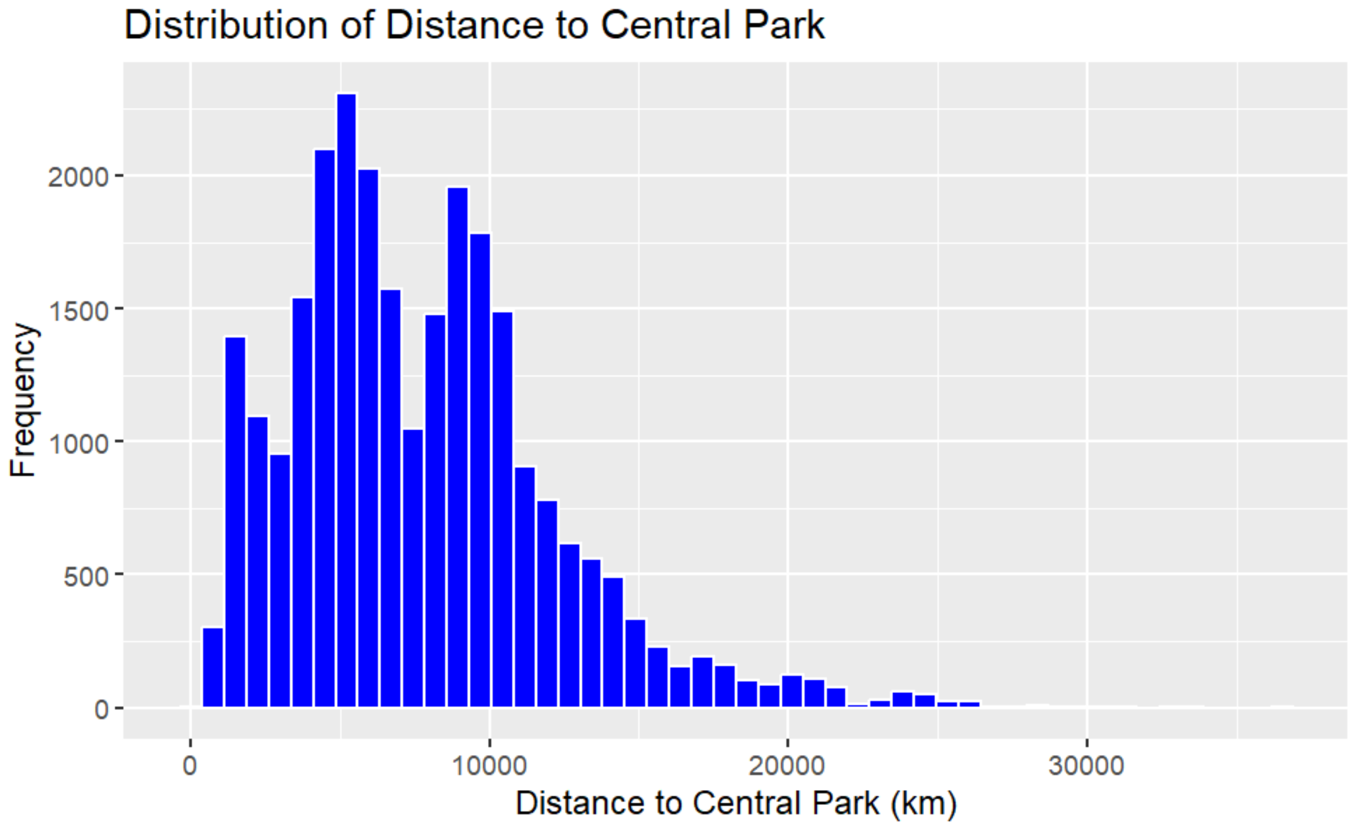
Hide

```
#Create a new feature for distance to Central Park
cp_lat <- airbnb %>%
  filter(neighbourhood_group == "Manhattan") %>%
  summarize(lat = mean(latitude), long = mean(longitude))

airbnb <- airbnb %>%
  mutate(dist_cp = geodist(cbind(longitude, latitude), c(cp_lat$long, cp_lat$lat)))
```

```
object has no named columns; assuming order is lon then lat
```

Hide

```
#View the distribution of the new feature
ggplot(airbnb, aes(x = dist_cp)) +
  geom_histogram(bins = 50, fill = "blue", color = "white") +
  ggtitle("Distribution of Distance to Central Park") +
  xlab("Distance to Central Park (km)") +
  ylab("Frequency")
```



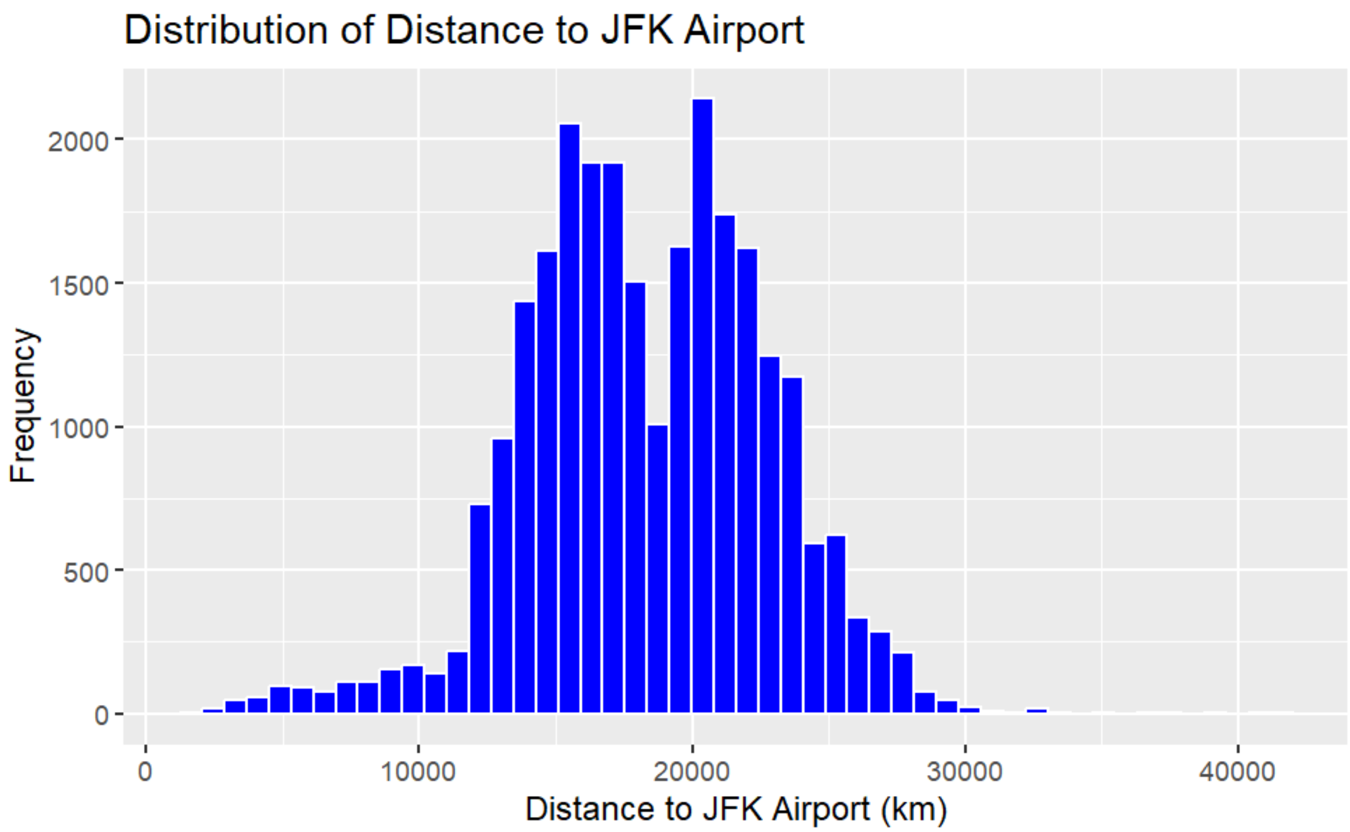
Hide

```
#Create a new feature for distance to JFK airport
jfk_lat <- 40.6413
jfk_long <- -73.7781

airbnb <- airbnb %>%
  mutate(dist_jfk = geodist(cbind(longitude, latitude), c(jfk_long, jfk_lat)))
```

```
object has no named columns; assuming order is lon then lat
```

Hide

```
#View the distribution of the new feature
ggplot(airbnb, aes(x = dist_jfk)) +
  geom_histogram(bins = 50, fill = "blue", color = "white") +
  ggtitle("Distribution of Distance to JFK Airport") +
  xlab("Distance to JFK Airport (km)") +
  ylab("Frequency")
```



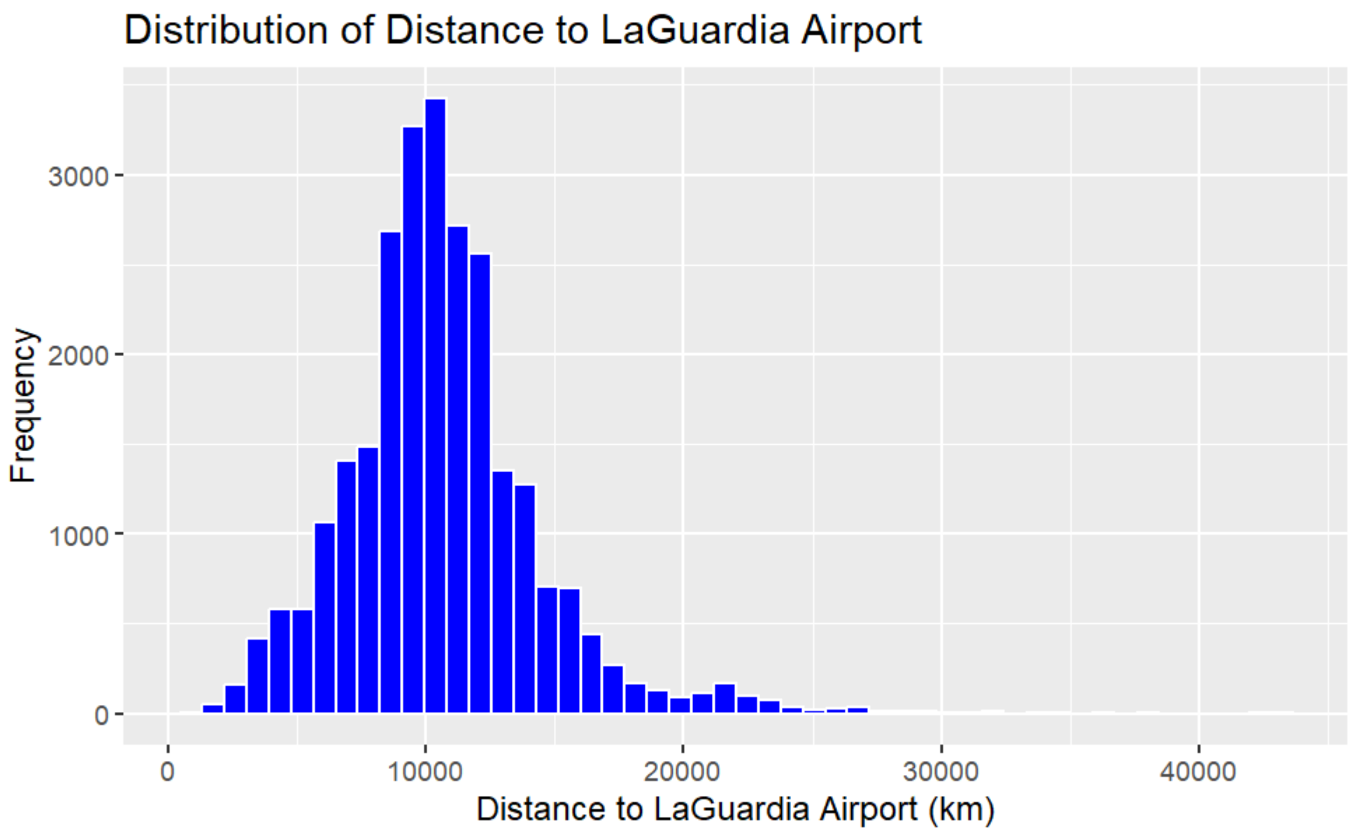
Hide

```
#Create a new feature for distance to LaGuardia airport
lga_lat <- 40.7769
lga_long <- -73.8740

airbnb <- airbnb %>%
  mutate(dist_lga = geodist(cbind(longitude, latitude), c(lga_long, lga_lat)))
```

```
object has no named columns; assuming order is lon then lat
```

Hide

```
#View the distribution of the new feature
ggplot(airbnb, aes(x = dist_lga)) +
  geom_histogram(bins = 50, fill = "blue", color = "white") +
  ggtitle("Distribution of Distance to LaGuardia Airport") +
  xlab("Distance to LaGuardia Airport (km)") +
  ylab("Frequency")
```



Hide

```
#Feature selection using correlation analysis
library(fastDummies)
library(tidyr)
library(tibble)
cor_df <- airbnb %>%
  select(price, dist_ts, dist_cp, dist_jfk, dist_lga, neighbourhood_group, room_type, minimum
_nights, availability_365,
         reviews_per_month,  number_of_reviews, calculated_host_listings_count) %>%
  mutate_if(is.character, as.factor) %>%
  dummy_cols() %>%
  select(-ends_with(".none")) %>%
  select(-starts_with("neighbourhood_group.")) %>%
  select_if(is.numeric) %>%   # only select numeric columns
  na.omit() %>%   # remove rows with missing values
  cor(use = "pairwise.complete.obs") %>%   # compute correlation matrix
  as.data.frame() %>%
  rownames_to_column(var = "variable") %>%
  gather(variable2, correlation, -variable) %>%
  mutate(correlation = abs(correlation)) %>%
  filter(variable != variable2)

#View the top 10 most correlated features with price
cor_df %>%
  arrange(desc(correlation)) %>%
  head(10)
```

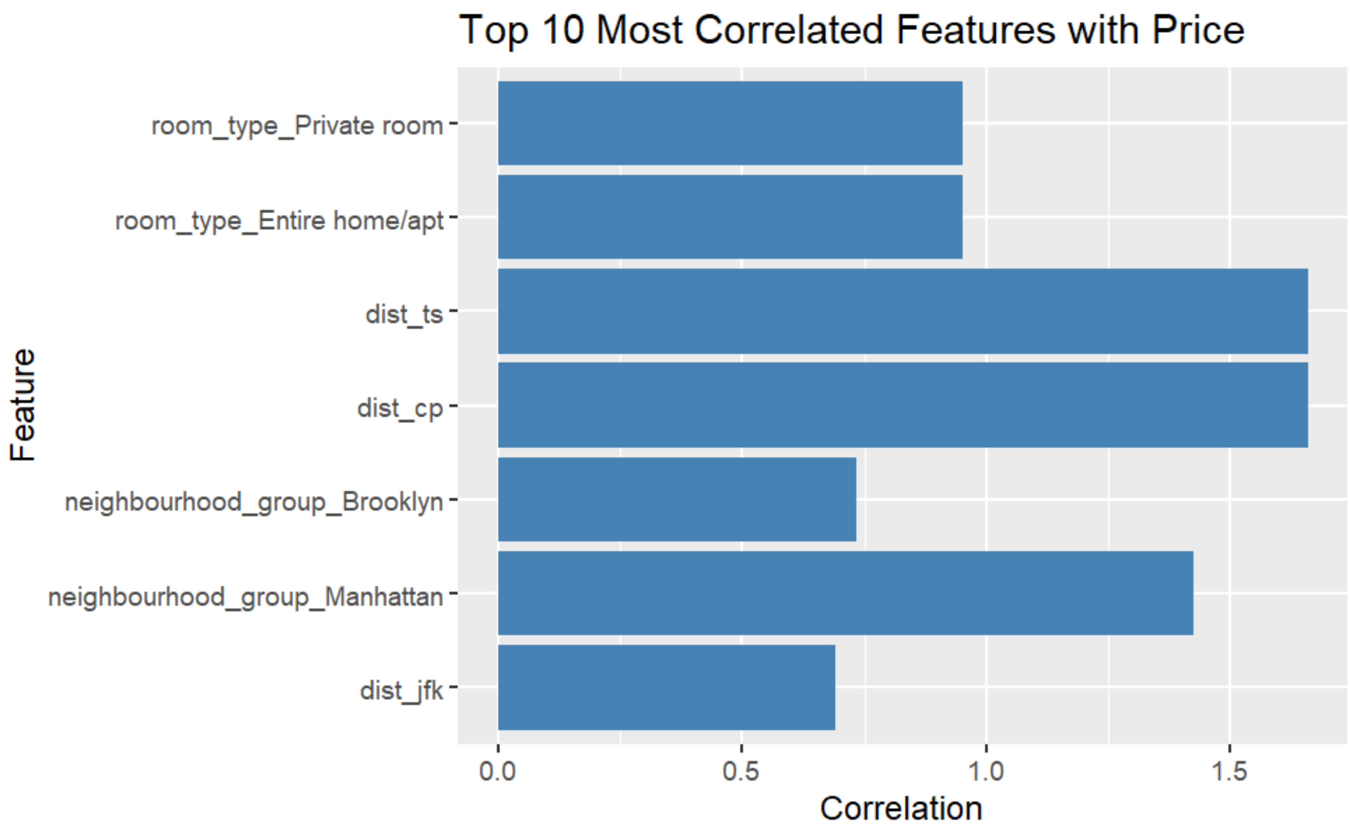
	variable <chr>	variable2 <chr>	correlation <dbl>
1	dist_cp	dist_ts	1.0000000

	variable <chr>	variable2 <chr>	correlation <dbl>
2	dist_ts	dist_cp	1.0000000
3	room_type_Private room	room_type_Entire home/apt	0.9532753
4	room_type_Entire home/apt	room_type_Private room	0.9532753
5	neighbourhood_group_Manhattan	neighbourhood_group_Brooklyn	0.7328875
6	neighbourhood_group_Brooklyn	neighbourhood_group_Manhattan	0.7328875
7	neighbourhood_group_Manhattan	dist_jfk	0.6922982
8	dist_jfk	neighbourhood_group_Manhattan	0.6922982
9	dist_lga	dist_ts	0.6600766
10	dist_lga	dist_cp	0.6600766

1-10 of 10 rows

Hide

```
library(ggplot2)

cor_df %>%
  arrange(desc(correlation)) %>%
  head(10) %>%
  ggplot(aes(x = reorder(variable2, correlation), y = correlation)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  coord_flip() +
  labs(x = "Feature", y = "Correlation", title = "Top 10 Most Correlated Features with Pric
e")
```



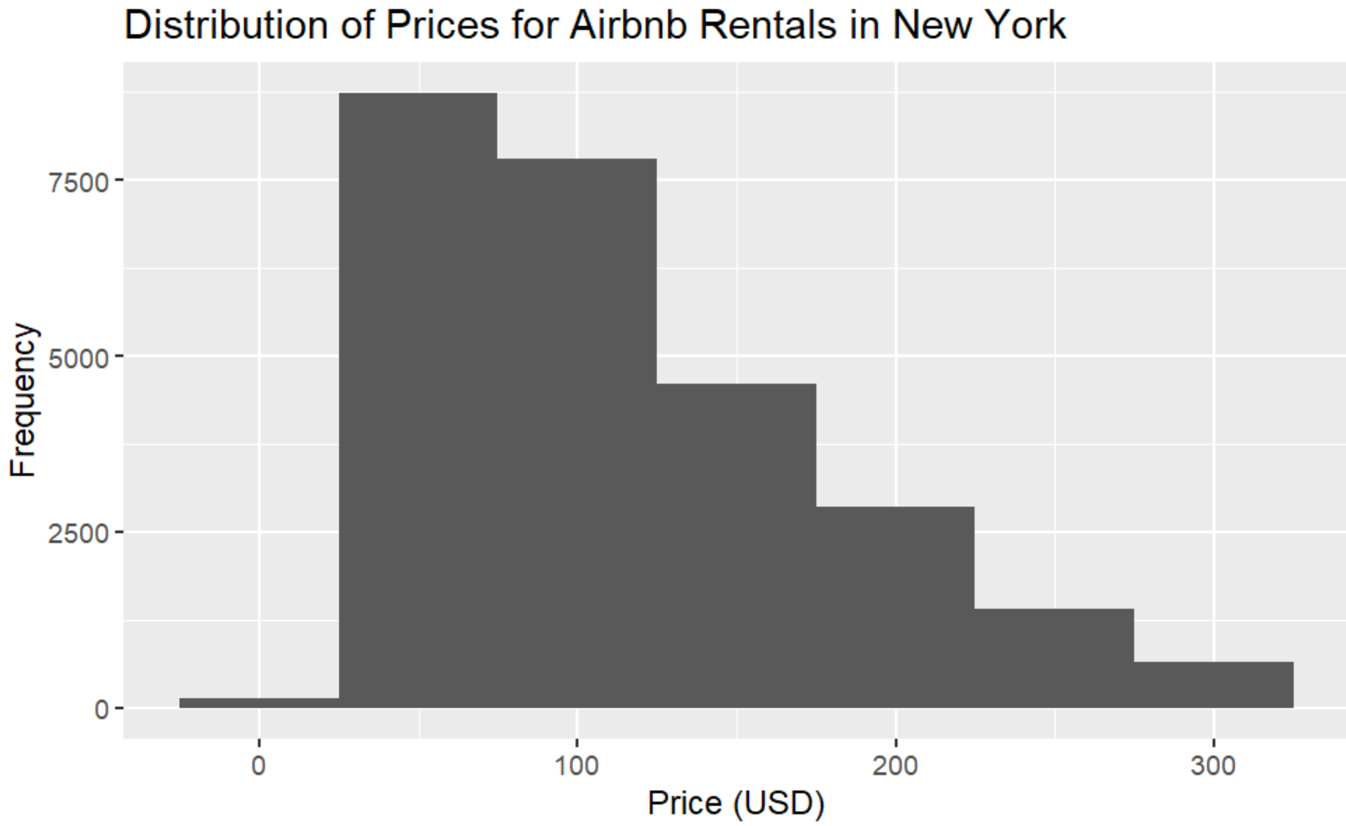
Hide

```
# Create a new data frame with only the selected features
selected_features <- c("price", "dist_ts", "dist_cp", "dist_jfk", "dist_lga", "neighbourhood_
group", "room_type")
airbnb_selected <- airbnb %>% select(selected_features)

# Check the new data frame
glimpse(airbnb_selected)
```

```
Rows: 26,214
Columns: 7
$ price               <int> 149, 225, 200, 135, 299, 80, 110, 60, 70,…
$ dist_ts             <dbl[,1]> <matrix[23 x 1]>
$ dist_cp             <dbl[,1]> <matrix[23 x 1]>
$ dist_jfk            <dbl[,1]> <matrix[23 x 1]>
$ dist_lga            <dbl[,1]> <matrix[23 x 1]>
$ neighbourhood_group <fct> Brooklyn, Manhattan, Manhattan, Manha…
$ room_type           <fct> Private room, Entire home/apt, Entire…
```

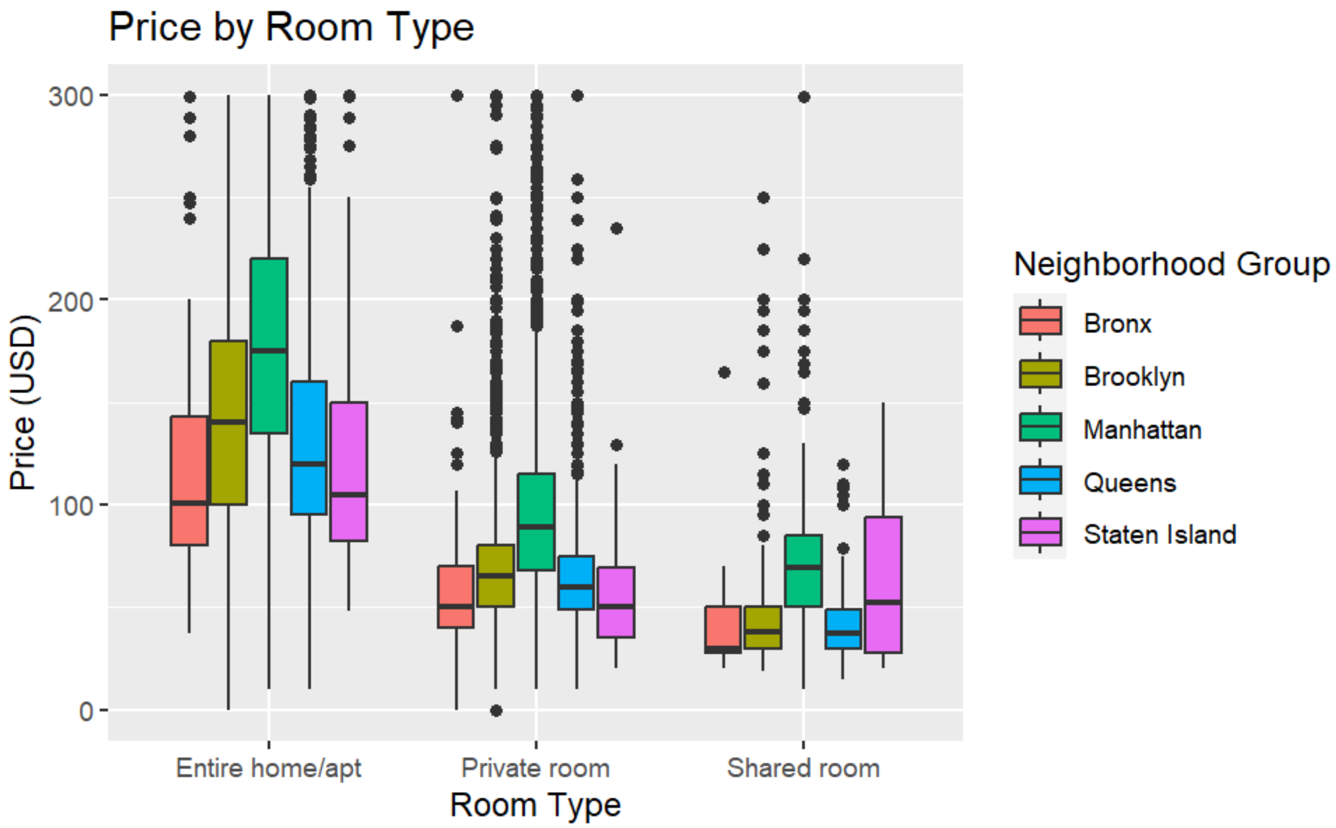
Hide

```
#Histogram of prices
ggplot(data = airbnb, aes(x = price)) +
  geom_histogram(binwidth = 50) +
  labs(title = "Distribution of Prices for Airbnb Rentals in New York",
       x = "Price (USD)",
       y = "Frequency")
```



Hide

```
#The histogram shows that the majority of Airbnb rentals in New York are priced between $0 an
d $1000 per night, with a
#few outliers priced over $1000 per night.
#Box plot of price by room type
ggplot(data = airbnb_selected, aes(x = room_type, y = price, fill = neighbourhood_group)) +
  geom_boxplot() +
  labs(title = "Price by Room Type",
       x = "Room Type",
       y = "Price (USD)",
       fill = "Neighborhood Group")
```



Hide

```
#The box plot shows that entire homes/apartments are generally the most expensive type of ren
tal,
#followed by private rooms and shared rooms. We can also see that rentals in the Manhattan an
d
#Brooklyn neighborhoods tend to be priced higher overall, regardless of room type.
```

Hide

```
#Loading the data
attach(Airbnb_data_cleaned1)
```

Hide

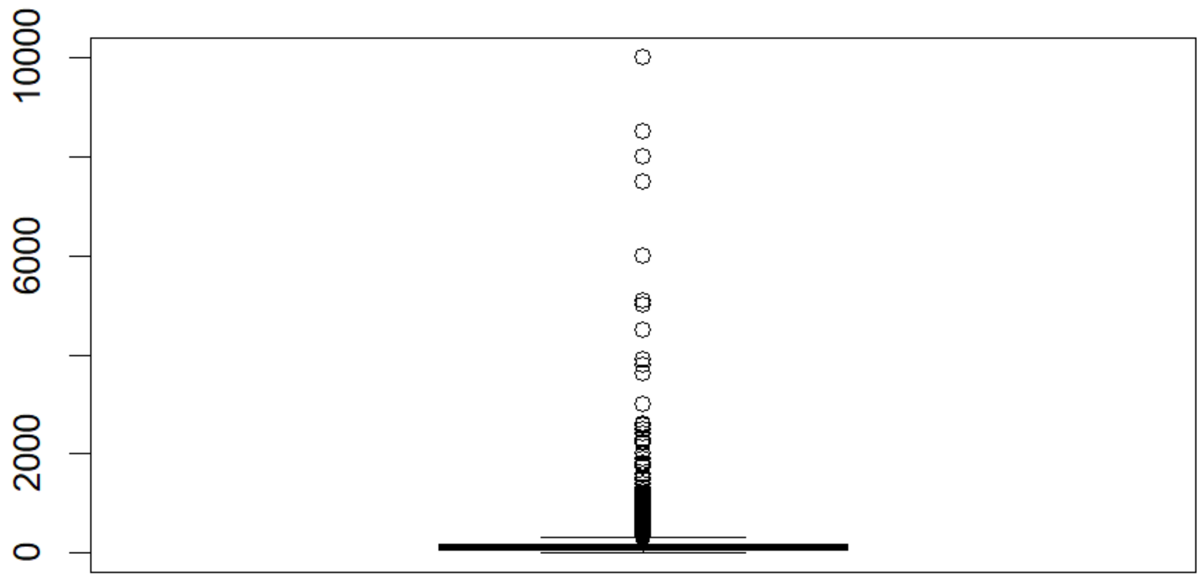
```
head(Airbnb_data_cleaned1)
```

	latitude	longitude	price	minimum_nights	number_of_reviews	reviews_per_month
	<dbl>	<dbl>	<int>	<int>	<int>	<dbl>
1	40.64749	-73.97237	149	1	9	0.21
2	40.75362	-73.98377	225	1	45	0.38
3	40.68514	-73.95976	89	1	270	4.64
4	40.79851	-73.94399	80	10	9	0.10
5	40.74767	-73.97500	200	3	74	0.59
6	40.68688	-73.95596	60	45	49	0.40

6 rows | 1-7 of 234 columns

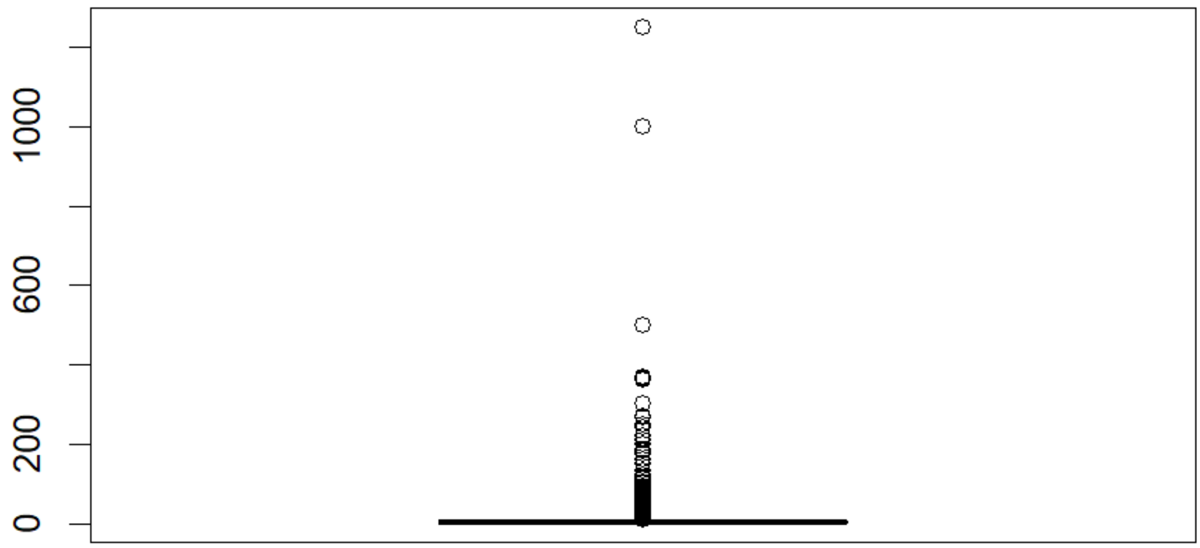
Hide

```
# Create a boxplot to visualize the distribution of the data
boxplot(Airbnb_data_cleaned1$price)
```



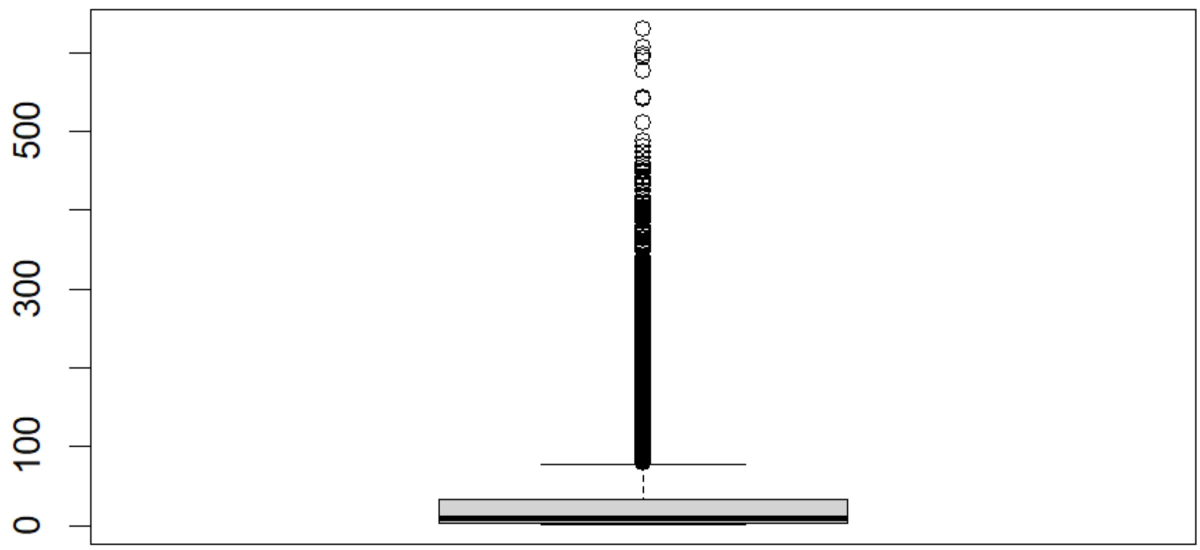
Hide

```
boxplot(Airbnb_data_cleaned1$minimum_nights)
```



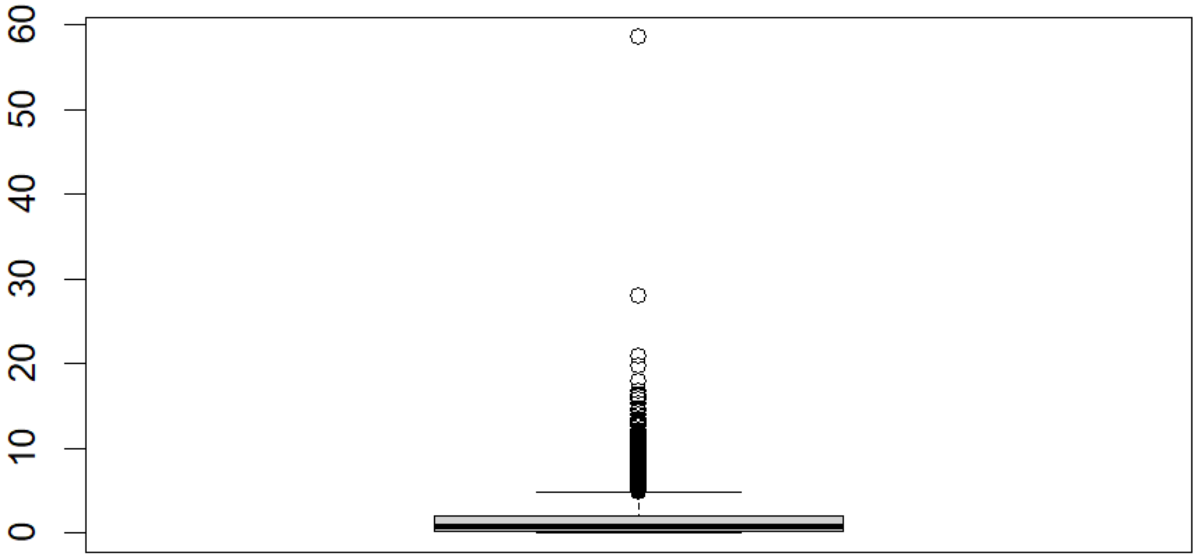
Hide

```
boxplot(Airbnb_data_cleaned1$number_of_reviews)
```



Hide

```
boxplot(Airbnb_data_cleaned1$reviews_per_month)
```



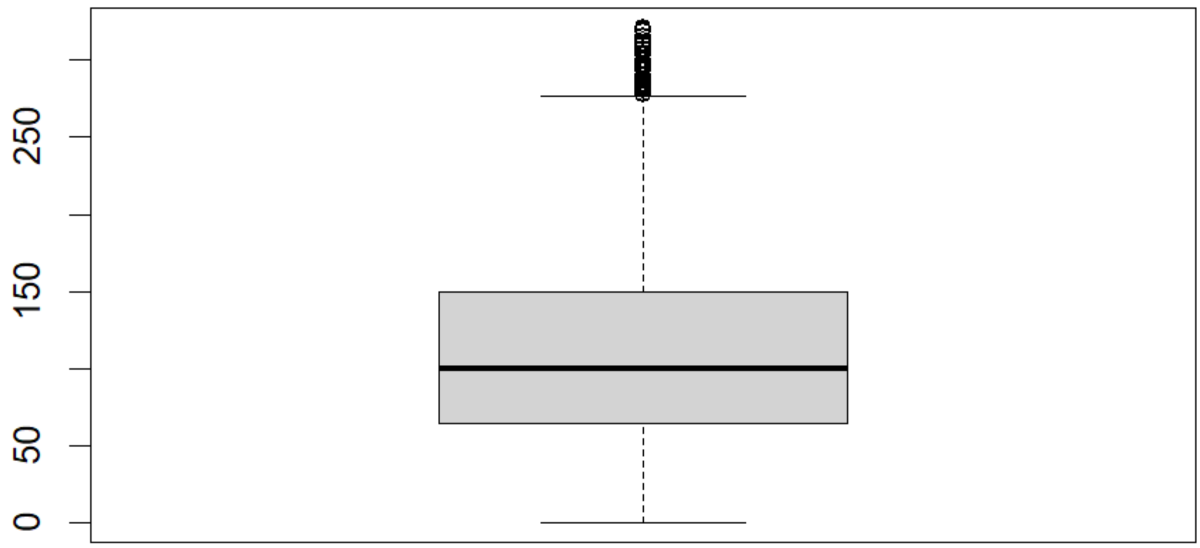
Hide

```
#Remove the outliers

# Create a list of column names
cols <- c("price", "minimum_nights", "reviews_per_month", "number_of_reviews")

# Loop over each column and remove outliers
Airbnb_data_cleaned1 <- Airbnb_data_cleaned1 %>%
 mutate(across(all_of(cols), ~ ifelse(. %in% boxplot.stats(.)$out, NA, .)))

# Remove rows with missing values
Airbnb_data_cleaned1 <- na.omit(Airbnb_data_cleaned1)
dim(Airbnb_data_cleaned1)
```

```
[1] 27473   234
```

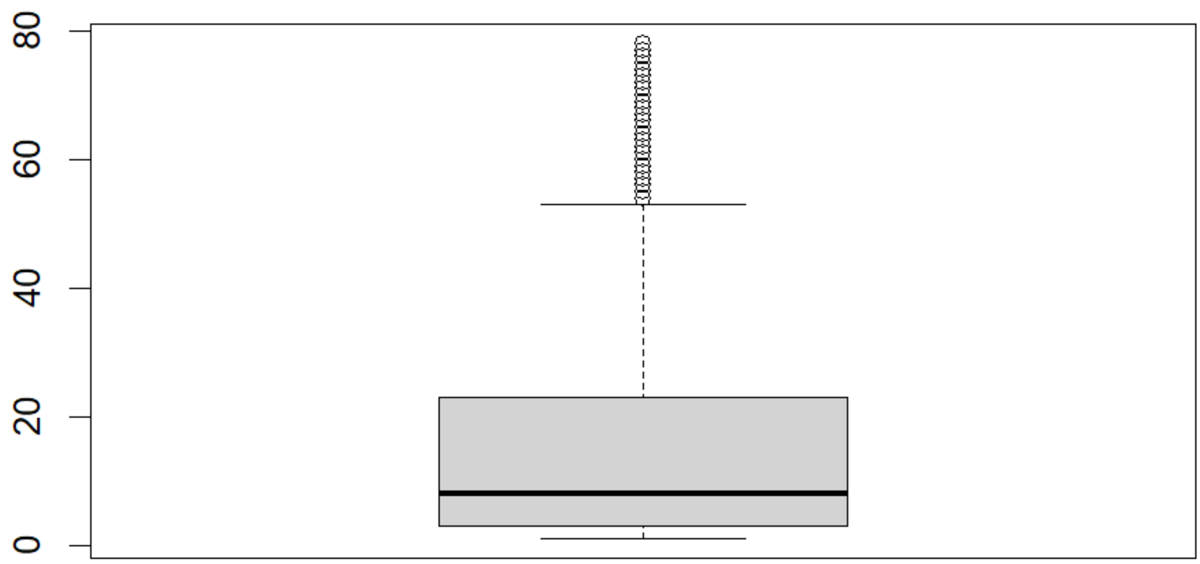
Hide

```
#Check again using boxplot
boxplot(Airbnb_data_cleaned1$price)
```



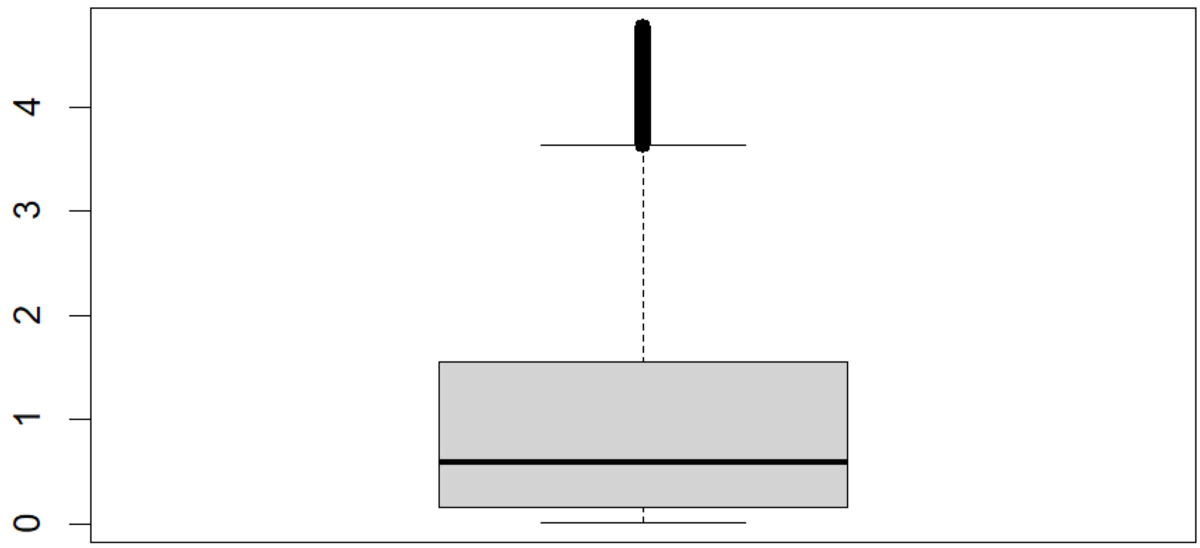
Hide

```
boxplot(Airbnb_data_cleaned1$number_of_reviews)
```



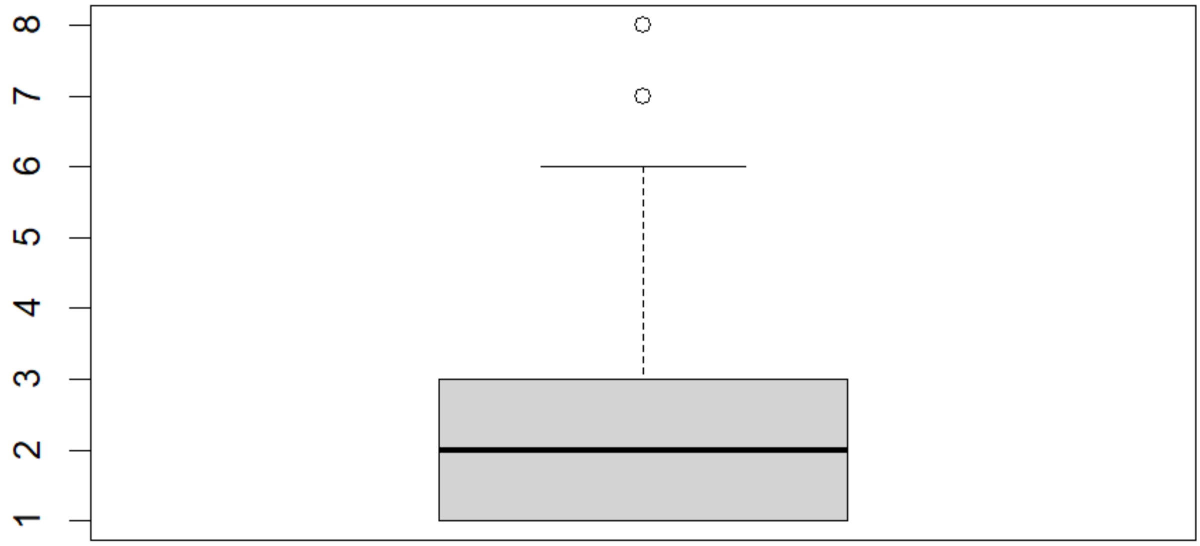
Hide

```
boxplot(Airbnb_data_cleaned1$reviews_per_month)
```



Hide

```
boxplot(Airbnb_data_cleaned1$minimum_nights)
```



Hide

```
#Divide the Airbnb data into training and testing (Preprocessing)

library(caret)
set.seed(1) # for reproducibility

# Create a vector of row indices
rows <- 1:nrow(Airbnb_data_cleaned1)

# Randomly sample 80% of the row indices for the training set
training_rows <- sample(rows, floor(0.8 * length(rows)))

# The remaining rows are for the testing set
testing_rows <- setdiff(rows, training_rows)

# Write the training and testing sets to separate files
write.table(Airbnb_data_cleaned1[training_rows, ], file = "Airbnb_training_data1.csv", row.na
mes = FALSE, col.names = FALSE)
write.table(Airbnb_data_cleaned1[testing_rows, ], file = "Airbnb_testing_data1.csv", row.name
s = FALSE, col.names = FALSE)

training_data <- Airbnb_data_cleaned1[training_rows, ]
testing_data <- Airbnb_data_cleaned1[-training_rows, ]

# Create X.train and X.test data frames that exclude the class label
X.train <- training_data[, -which(names(training_data) == "price")]
X.test <- testing_data[, -which(names(testing_data) == "price")]
Y.train <- training_data$price
Y.test <- testing_data$price

#Division Verification in number of Examples
cat("Number of examples in training data:", nrow(training_data), "\n")
```

```
Number of examples in training data: 21978
```

Hide

```
cat("Number of examples in testing data:", nrow(testing_data), "\n")
```

```
Number of examples in testing data: 5495
```

Hide

```
# Multiple Linear Regression Model
first_model<-lm(price~., data=training_data)
first_prediction<-predict(first_model, testing_data)
```

```
Warning: prediction from a rank-deficient fit may be misleading
```

Hide

```
summary(first_model)
```

```
Call:
lm(formula = price ~ ., data = training_data)

Residuals:
     Min       1Q   Median       3Q      Max 
-179.261  -26.858   -6.032   18.290  253.899 

Coefficients: (11 not defined because of singularities)
                                          Estimate Std. Error t value
(Intercept)                             -2.581e+04  3.793e+03  -6.806
latitude                                -1.175e+02  5.412e+01  -2.171
longitude                               -4.142e+02  4.061e+01 -10.201
minimum_nights                          -1.674e+00  1.892e-01  -8.850
number_of_reviews                       -6.820e-02  1.905e-02  -3.581
reviews_per_month                       -2.082e-02  3.290e-01  -0.063
calculated_host_listings_count           5.256e-02  1.540e-02   3.412
availability_365                         8.387e-02  2.676e-03  31.342
neighbourhood_group_Bronx                4.991e+01  5.138e+01   0.971
neighbourhood_group_Brooklyn            -2.318e-01  4.503e+01  -0.005
neighbourhood_group_Manhattan            4.963e+01  4.515e+01   1.099
neighbourhood_group_Queens               1.805e+01  4.606e+01   0.392
neighbourhood_group_Staten.Island               NA         NA      NA
neighbourhood_Allerton                  -2.017e+01  2.054e+01  -0.982
neighbourhood_Arden.Heights             -1.581e+02  5.424e+01  -2.914
neighbourhood_Arrochar                  -9.337e+01  4.592e+01  -2.033
neighbourhood_Arverne                    5.406e+01  1.266e+01   4.269
neighbourhood_Astoria                    1.930e+00  5.324e+00   0.363
neighbourhood_Bath.Beach                -5.836e+01  1.325e+01  -4.406
neighbourhood_Battery.Park.City         -1.645e+01  9.650e+00  -1.704
neighbourhood_Bay.Ridge                 -4.215e+01  7.825e+00  -5.386
neighbourhood_Bay.Terrace                7.065e+01  2.651e+01   2.665
neighbourhood_Bay.Terrace..Staten.Island        NA         NA      NA
neighbourhood_Baychester                -3.469e+01  3.613e+01  -0.960
neighbourhood_Bayside                    6.891e+01  1.271e+01   5.420
neighbourhood_Bayswater                  3.874e+01  1.995e+01   1.942
neighbourhood_Bedford.Stuyvesant        -1.369e+00  5.790e+00  -0.237
neighbourhood_Belle.Harbor               6.902e+01  2.455e+01   2.811
neighbourhood_Bellerose                  7.546e+01  1.884e+01   4.006
neighbourhood_Belmont                   -1.001e+01  2.129e+01  -0.470
neighbourhood_Bensonhurst               -4.838e+01  8.963e+00  -5.398
neighbourhood_Bergen.Beach              -2.931e+01  1.780e+01  -1.646
neighbourhood_Boerum.Hill                1.414e+01  7.477e+00   1.891
neighbourhood_Borough.Park              -4.953e+01  7.409e+00  -6.685
neighbourhood_Breezy.Point               1.215e+02  3.315e+01   3.666
neighbourhood_Briarwood                  1.265e+01  1.040e+01   1.217
neighbourhood_Brighton.Beach            -2.935e+01  9.884e+00  -2.969
neighbourhood_Bronxdale                 -4.173e+01  2.193e+01  -1.903
neighbourhood_Brooklyn.Heights           1.737e+01  7.618e+00   2.280
neighbourhood_Brownsville                1.134e+00  9.237e+00   0.123
neighbourhood_Bull.s.Head               -1.253e+02  5.107e+01  -2.454
neighbourhood_Bushwick                   5.171e+00  6.248e+00   0.828
neighbourhood_Cambria.Heights            7.444e+01  1.650e+01   4.511
neighbourhood_Canarsie                   8.524e-01  7.840e+00   0.109
neighbourhood_Carroll.Gardens            3.165e+00  6.887e+00   0.460
```

```
neighbourhood_Castle.Hill                -6.770e+01  4.796e+01  -1.412
neighbourhood_Castleton.Corners          -4.917e+01  5.414e+01  -0.908
neighbourhood_Chelsea                    -6.746e+00  3.403e+00  -1.982
neighbourhood_Chinatown                  -2.812e+01  4.247e+00  -6.620
neighbourhood_City.Island                 8.272e+00  2.323e+01   0.356
neighbourhood_Civic.Center               -2.647e+01  1.078e+01  -2.456
neighbourhood_Claremont.Village          -3.880e+01  2.310e+01  -1.680
neighbourhood_Clason.Point               -5.311e+00  2.256e+01  -0.235
neighbourhood_Clifton                    -8.577e+01  4.731e+01  -1.813
neighbourhood_Clinton.Hill                7.976e+00  6.153e+00   1.296
neighbourhood_Co.op.City                  1.200e+01  4.782e+01   0.251
neighbourhood_Cobble.Hill                 1.381e+01  8.510e+00   1.623
neighbourhood_College.Point              -3.423e+00  1.664e+01  -0.206
neighbourhood_Columbia.St                -1.488e+01  1.094e+01  -1.360
neighbourhood_Concord                    -1.001e+02  4.545e+01  -2.203
neighbourhood_Concourse                  -4.899e+01  2.047e+01  -2.394
neighbourhood_Concourse.Village          -4.194e+01  2.175e+01  -1.929
neighbourhood_Coney.Island               -2.715e+01  1.400e+01  -1.938
neighbourhood_Corona                     -9.916e+00  8.829e+00  -1.123
neighbourhood_Crown.Heights              -3.454e+00  5.670e+00  -0.609
neighbourhood_Cypress.Hills               5.700e+00  8.538e+00   0.668
neighbourhood_DUMBO                       3.531e+01  1.515e+01   2.331
neighbourhood_Ditmars.Steinway           -2.753e+00  6.087e+00  -0.452
neighbourhood_Dongan.Hills               -1.073e+02  4.778e+01  -2.245
neighbourhood_Douglaston                  4.621e+01  2.687e+01   1.720
neighbourhood_Downtown.Brooklyn           2.095e+01  9.444e+00   2.218
neighbourhood_Dyker.Heights              -4.769e+01  1.661e+01  -2.871
neighbourhood_East.Elmhurst               1.134e+00  7.291e+00   0.155
neighbourhood_East.Flatbush              -1.481e+01  6.134e+00  -2.415
neighbourhood_East.Harlem                -1.856e+01  5.197e+00  -3.572
neighbourhood_East.Morrisania            -1.865e+01  2.414e+01  -0.773
neighbourhood_East.New.York               8.643e+00  7.632e+00   1.132
neighbourhood_East.Village               -1.941e+01  3.069e+00  -6.325
neighbourhood_Eastchester                 8.576e+00  2.337e+01   0.367
neighbourhood_Edenwald                   -1.038e+01  2.392e+01  -0.434
neighbourhood_Edgemere                    2.452e+01  2.263e+01   1.084
neighbourhood_Elmhurst                    7.339e+00  6.397e+00   1.147
neighbourhood_Eltingville                        NA         NA      NA
neighbourhood_Emerson.Hill               -1.716e+02  6.252e+01  -2.745
neighbourhood_Far.Rockaway                2.660e+01  1.538e+01   1.730
neighbourhood_Fieldston                  -3.004e+01  2.287e+01  -1.314
neighbourhood_Financial.District         -1.961e+01  4.382e+00  -4.474
neighbourhood_Flatbush                   -2.553e+01  5.844e+00  -4.369
neighbourhood_Flatiron.District           5.777e+00  8.758e+00   0.660
neighbourhood_Flatlands                  -1.005e+01  8.700e+00  -1.155
neighbourhood_Flushing                    3.298e+01  6.543e+00   5.040
neighbourhood_Fordham                    -3.206e+01  2.014e+01  -1.592
neighbourhood_Forest.Hills                2.491e+01  7.490e+00   3.326
neighbourhood_Fort.Greene                 1.343e+01  6.267e+00   2.143
neighbourhood_Fort.Hamilton              -5.845e+01  9.557e+00  -6.116
neighbourhood_Fresh.Meadows               4.131e+01  1.293e+01   3.195
neighbourhood_Glendale                   -6.517e+00  9.721e+00  -0.670
neighbourhood_Gowanus                     5.006e+00  6.782e+00   0.738
neighbourhood_Gramercy                   -9.842e+00  4.554e+00  -2.161
neighbourhood_Graniteville               -1.503e+02  5.417e+01  -2.774
neighbourhood_Grant.City                 -1.412e+02  5.106e+01  -2.765
```

```
neighbourhood_Gravesend                  -4.013e+01  1.038e+01  -3.866
neighbourhood_Great.Kills                -9.956e+01  4.782e+01  -2.082
neighbourhood_Greenpoint                  2.104e+01  6.906e+00   3.046
neighbourhood_Greenwich.Village          -5.963e+00  4.082e+00  -1.461
neighbourhood_Grymes.Hill                -8.633e+01  4.945e+01  -1.746
neighbourhood_Harlem                     -2.563e+01  5.648e+00  -4.537
neighbourhood_Hell.s.Kitchen             -6.456e+00  3.396e+00  -1.901
neighbourhood_Highbridge                 -3.767e+01  2.218e+01  -1.698
neighbourhood_Hollis                      4.704e+01  1.834e+01   2.565
neighbourhood_Holliswood                  1.463e+02  3.212e+01   4.554
neighbourhood_Howard.Beach                2.994e+01  1.811e+01   1.653
neighbourhood_Howland.Hook               -1.631e+02  6.255e+01  -2.608
neighbourhood_Huguenot                   -1.791e+02  5.425e+01  -3.301
neighbourhood_Hunts.Point                -3.230e+01  2.589e+01  -1.248
neighbourhood_Inwood                     -2.975e+01  8.969e+00  -3.317
neighbourhood_Jackson.Heights             1.188e+00  6.541e+00   0.182
neighbourhood_Jamaica                     2.506e+01  8.588e+00   2.918
neighbourhood_Jamaica.Estates             2.015e+01  1.927e+01   1.046
neighbourhood_Jamaica.Hills               2.767e+01  2.638e+01   1.049
neighbourhood_Kensington                 -3.282e+01  6.990e+00  -4.695
neighbourhood_Kew.Gardens                 1.148e+01  1.258e+01   0.913
neighbourhood_Kew.Gardens.Hills           1.711e+01  1.258e+01   1.360
neighbourhood_Kingsbridge                -2.821e+01  1.979e+01  -1.426
neighbourhood_Kips.Bay                   -1.787e+01  4.293e+00  -4.162
neighbourhood_Laurelton                   3.951e+01  1.725e+01   2.290
neighbourhood_Lighthouse.Hill            -5.603e+01  6.252e+01  -0.896
neighbourhood_Little.Italy               -3.457e+01  7.939e+00  -4.355
neighbourhood_Little.Neck                 6.177e+01  2.695e+01   2.292
neighbourhood_Long.Island.City            1.360e+00  5.716e+00   0.238
neighbourhood_Longwood                   -3.434e+01  2.007e+01  -1.711
neighbourhood_Lower.East.Side            -2.761e+01  3.506e+00  -7.874
neighbourhood_Manhattan.Beach            -2.713e+01  2.311e+01  -1.174
neighbourhood_Marble.Hill                -6.597e+00  2.007e+01  -0.329
neighbourhood_Mariners.Harbor            -1.214e+02  4.847e+01  -2.505
neighbourhood_Maspeth                    -1.848e+01  8.254e+00  -2.238
neighbourhood_Melrose                    -3.678e+01  2.714e+01  -1.355
neighbourhood_Middle.Village              1.888e+00  1.253e+01   0.151
neighbourhood_Midland.Beach              -1.122e+02  5.108e+01  -2.196
neighbourhood_Midtown                     4.213e+00  3.659e+00   1.151
neighbourhood_Midwood                    -3.579e+01  8.034e+00  -4.455
neighbourhood_Mill.Basin                 -1.424e+01  2.627e+01  -0.542
neighbourhood_Morningside.Heights        -3.059e+01  6.103e+00  -5.011
neighbourhood_Morris.Heights             -4.297e+01  2.479e+01  -1.733
neighbourhood_Morris.Park                -1.910e+01  2.399e+01  -0.796
neighbourhood_Morrisania                 -2.925e+01  2.880e+01  -1.016
neighbourhood_Mott.Haven                 -4.760e+01  2.035e+01  -2.340
neighbourhood_Mount.Eden                 -3.197e+01  3.626e+01  -0.882
neighbourhood_Mount.Hope                 -3.364e+01  2.302e+01  -1.461
neighbourhood_Murray.Hill                 5.650e+00  5.061e+00   1.117
neighbourhood_Navy.Yard                   3.137e+01  1.511e+01   2.075
neighbourhood_Neponsit                    9.128e+01  3.310e+01   2.757
neighbourhood_New.Brighton               -4.511e+01  5.421e+01  -0.832
neighbourhood_New.Dorp.Beach             -1.190e+02  6.255e+01  -1.902
neighbourhood_New.Springville            -1.103e+02  5.107e+01  -2.160
neighbourhood_NoHo                       -5.091e-01  9.212e+00  -0.055
neighbourhood_Nolita                     -9.957e+00  4.938e+00  -2.017
```

```
neighbourhood_North.Riverdale            -2.764e+01  2.856e+01  -0.968
neighbourhood_Norwood                    -2.020e+01  2.102e+01  -0.961
neighbourhood_Oakwood                    -1.180e+02  5.108e+01  -2.310
neighbourhood_Olinville                  -9.175e+00  3.610e+01  -0.254
neighbourhood_Ozone.Park                 -7.932e+00  1.065e+01  -0.745
neighbourhood_Park.Slope                  1.078e+01  6.125e+00   1.759
neighbourhood_Parkchester                -1.156e+01  2.030e+01  -0.569
neighbourhood_Pelham.Bay                 -7.929e+00  2.349e+01  -0.338
neighbourhood_Pelham.Gardens             -3.914e+01  2.168e+01  -1.806
neighbourhood_Port.Morris                -4.178e+01  2.059e+01  -2.030
neighbourhood_Port.Richmond              -1.208e+02  4.846e+01  -2.492
neighbourhood_Prince.s.Bay               -1.260e+02  5.435e+01  -2.318
neighbourhood_Prospect.Heights            9.075e+00  6.347e+00   1.430
neighbourhood_Prospect.Lefferts.Gardens  -1.927e+01  5.961e+00  -3.233
neighbourhood_Queens.Village              3.323e+01  1.188e+01   2.796
neighbourhood_Randall.Manor              -1.241e+02  4.604e+01  -2.694
neighbourhood_Red.Hook                   -2.115e+01  8.977e+00  -2.356
neighbourhood_Rego.Park                   3.625e+00  7.470e+00   0.485
neighbourhood_Richmond.Hill               1.719e+01  9.050e+00   1.900
neighbourhood_Richmondtown                       NA         NA      NA
neighbourhood_Ridgewood                  -9.597e+00  6.175e+00  -1.554
neighbourhood_Riverdale                   3.863e+00  2.859e+01   0.135
neighbourhood_Rockaway.Beach              2.111e+01  1.344e+01   1.571
neighbourhood_Roosevelt.Island           -2.763e+01  7.692e+00  -3.592
neighbourhood_Rosebank                   -5.010e+01  5.110e+01  -0.981
neighbourhood_Rosedale                    4.664e+01  1.247e+01   3.740
neighbourhood_Rossville                  -1.886e+02  6.264e+01  -3.011
neighbourhood_Schuylerville              -1.305e+01  2.579e+01  -0.506
neighbourhood_Sea.Gate                    4.610e+01  4.471e+01   1.031
neighbourhood_Sheepshead.Bay             -2.755e+01  7.996e+00  -3.445
neighbourhood_Shore.Acres                -9.950e+01  5.111e+01  -1.947
neighbourhood_Silver.Lake                        NA         NA      NA
neighbourhood_SoHo                       -6.454e+00  4.620e+00  -1.397
neighbourhood_Soundview                  -4.284e+01  2.488e+01  -1.722
neighbourhood_South.Beach                -5.744e+01  4.778e+01  -1.202
neighbourhood_South.Ozone.Park            2.116e+00  1.310e+01   0.162
neighbourhood_South.Slope                 7.611e+00  6.472e+00   1.176
neighbourhood_Springfield.Gardens         5.106e+01  1.120e+01   4.560
neighbourhood_Spuyten.Duyvil             -4.914e+01  3.616e+01  -1.359
neighbourhood_St..Albans                  4.612e+01  1.075e+01   4.289
neighbourhood_St..George                 -7.478e+01  4.611e+01  -1.622
neighbourhood_Stapleton                  -6.911e+01  4.625e+01  -1.494
neighbourhood_Stuyvesant.Town            -2.974e+01  1.425e+01  -2.088
                                         Pr(>|t|)
(Intercept)                              1.03e-11 ***
latitude                                 0.029930 *
longitude                                 < 2e-16 ***
minimum_nights                            < 2e-16 ***
number_of_reviews                        0.000344 ***
reviews_per_month                        0.949549
calculated_host_listings_count           0.000645 ***
availability_365                          < 2e-16 ***
neighbourhood_group_Bronx                0.331399
neighbourhood_group_Brooklyn             0.995893
neighbourhood_group_Manhattan            0.271655
neighbourhood_group_Queens               0.695052
```

```
neighbourhood_group_Staten.Island                 NA
neighbourhood_Allerton                      0.325965
neighbourhood_Arden.Heights                 0.003570 **
neighbourhood_Arrochar                      0.042064 *
neighbourhood_Arverne                       1.97e-05 ***
neighbourhood_Astoria                       0.716913
neighbourhood_Bath.Beach                    1.06e-05 ***
neighbourhood_Battery.Park.City             0.088345 .
neighbourhood_Bay.Ridge                     7.27e-08 ***
neighbourhood_Bay.Terrace                   0.007707 **
neighbourhood_Bay.Terrace..Staten.Island          NA
neighbourhood_Baychester                    0.337082
neighbourhood_Bayside                       6.02e-08 ***
neighbourhood_Bayswater                     0.052177 .
neighbourhood_Bedford.Stuyvesant            0.813037
neighbourhood_Belle.Harbor                  0.004943 **
neighbourhood_Bellerose                     6.19e-05 ***
neighbourhood_Belmont                       0.638440
neighbourhood_Bensonhurst                   6.81e-08 ***
neighbourhood_Bergen.Beach                  0.099763 .
neighbourhood_Boerum.Hill                   0.058612 .
neighbourhood_Borough.Park                  2.36e-11 ***
neighbourhood_Breezy.Point                  0.000247 ***
neighbourhood_Briarwood                     0.223661
neighbourhood_Brighton.Beach                0.002989 **
neighbourhood_Bronxdale                     0.057108 .
neighbourhood_Brooklyn.Heights              0.022624 *
neighbourhood_Brownsville                   0.902258
neighbourhood_Bull.s.Head                   0.014130 *
neighbourhood_Bushwick                      0.407861
neighbourhood_Cambria.Heights               6.48e-06 ***
neighbourhood_Canarsie                      0.913428
neighbourhood_Carroll.Gardens               0.645828
neighbourhood_Castle.Hill                   0.158071
neighbourhood_Castleton.Corners             0.363832
neighbourhood_Chelsea                       0.047455 *
neighbourhood_Chinatown                     3.67e-11 ***
neighbourhood_City.Island                   0.721774
neighbourhood_Civic.Center                  0.014059 *
neighbourhood_Claremont.Village             0.093003 .
neighbourhood_Clason.Point                  0.813904
neighbourhood_Clifton                       0.069839 .
neighbourhood_Clinton.Hill                  0.194872
neighbourhood_Co.op.City                    0.801918
neighbourhood_Cobble.Hill                   0.104655
neighbourhood_College.Point                 0.837053
neighbourhood_Columbia.St                   0.173946
neighbourhood_Concord                       0.027591 *
neighbourhood_Concourse                     0.016695 *
neighbourhood_Concourse.Village             0.053798 .
neighbourhood_Coney.Island                  0.052581 .
neighbourhood_Corona                        0.261391
neighbourhood_Crown.Heights                 0.542331
neighbourhood_Cypress.Hills                 0.504380
neighbourhood_DUMBO                         0.019783 *
neighbourhood_Ditmars.Steinway              0.651109
```

```
neighbourhood_Dongan.Hills               0.024803 *
neighbourhood_Douglaston                 0.085461 .
neighbourhood_Downtown.Brooklyn          0.026535 *
neighbourhood_Dyker.Heights              0.004099 **
neighbourhood_East.Elmhurst              0.876448
neighbourhood_East.Flatbush              0.015758 *
neighbourhood_East.Harlem                0.000355 ***
neighbourhood_East.Morrisania            0.439769
neighbourhood_East.New.York              0.257463
neighbourhood_East.Village               2.59e-10 ***
neighbourhood_Eastchester                0.713626
neighbourhood_Edenwald                   0.664417
neighbourhood_Edgemere                   0.278546
neighbourhood_Elmhurst                   0.251299
neighbourhood_Eltingville                      NA
neighbourhood_Emerson.Hill               0.006053 **
neighbourhood_Far.Rockaway               0.083666 .
neighbourhood_Fieldston                  0.188987
neighbourhood_Financial.District         7.71e-06 ***
neighbourhood_Flatbush                   1.26e-05 ***
neighbourhood_Flatiron.District          0.509513
neighbourhood_Flatlands                  0.248233
neighbourhood_Flushing                   4.69e-07 ***
neighbourhood_Fordham                    0.111305
neighbourhood_Forest.Hills               0.000881 ***
neighbourhood_Fort.Greene                0.032141 *
neighbourhood_Fort.Hamilton              9.77e-10 ***
neighbourhood_Fresh.Meadows              0.001402 **
neighbourhood_Glendale                   0.502610
neighbourhood_Gowanus                    0.460438
neighbourhood_Gramercy                   0.030715 *
neighbourhood_Graniteville               0.005545 **
neighbourhood_Grant.City                 0.005695 **
neighbourhood_Gravesend                  0.000111 ***
neighbourhood_Great.Kills                0.037374 *
neighbourhood_Greenpoint                 0.002319 **
neighbourhood_Greenwich.Village          0.144099
neighbourhood_Grymes.Hill                0.080888 .
neighbourhood_Harlem                     5.74e-06 ***
neighbourhood_Hell.s.Kitchen             0.057336 .
neighbourhood_Highbridge                 0.089465 .
neighbourhood_Hollis                     0.010319 *
neighbourhood_Holliswood                 5.28e-06 ***
neighbourhood_Howard.Beach               0.098260 .
neighbourhood_Howland.Hook               0.009125 **
neighbourhood_Huguenot                   0.000964 ***
neighbourhood_Hunts.Point                0.212185
neighbourhood_Inwood                     0.000910 ***
neighbourhood_Jackson.Heights            0.855915
neighbourhood_Jamaica                    0.003522 **
neighbourhood_Jamaica.Estates            0.295746
neighbourhood_Jamaica.Hills              0.294299
neighbourhood_Kensington                 2.68e-06 ***
neighbourhood_Kew.Gardens                0.361226
neighbourhood_Kew.Gardens.Hills          0.173784
neighbourhood_Kingsbridge                0.154006
```

```
neighbourhood_Kips.Bay                    3.17e-05 ***
neighbourhood_Laurelton                   0.022038 *
neighbourhood_Lighthouse.Hill             0.370136
neighbourhood_Little.Italy                1.34e-05 ***
neighbourhood_Little.Neck                 0.021907 *
neighbourhood_Long.Island.City            0.811985
neighbourhood_Longwood                    0.087068 .
neighbourhood_Lower.East.Side             3.60e-15 ***
neighbourhood_Manhattan.Beach             0.240281
neighbourhood_Marble.Hill                 0.742369
neighbourhood_Mariners.Harbor             0.012250 *
neighbourhood_Maspeth                     0.025203 *
neighbourhood_Melrose                     0.175432
neighbourhood_Middle.Village              0.880204
neighbourhood_Midland.Beach               0.028104 *
neighbourhood_Midtown                     0.249545
neighbourhood_Midwood                     8.43e-06 ***
neighbourhood_Mill.Basin                  0.587619
neighbourhood_Morningside.Heights         5.45e-07 ***
neighbourhood_Morris.Heights              0.083036 .
neighbourhood_Morris.Park                 0.426040
neighbourhood_Morrisania                  0.309870
neighbourhood_Mott.Haven                  0.019311 *
neighbourhood_Mount.Eden                  0.377928
neighbourhood_Mount.Hope                  0.143903
neighbourhood_Murray.Hill                 0.264202
neighbourhood_Navy.Yard                   0.037985 *
neighbourhood_Neponsit                    0.005833 **
neighbourhood_New.Brighton                0.405309
neighbourhood_New.Dorp.Beach              0.057207 .
neighbourhood_New.Springville             0.030815 *
neighbourhood_NoHo                        0.955928
neighbourhood_Nolita                      0.043747 *
neighbourhood_North.Riverdale             0.333262
neighbourhood_Norwood                     0.336697
neighbourhood_Oakwood                     0.020925 *
neighbourhood_Olinville                   0.799388
neighbourhood_Ozone.Park                  0.456316
neighbourhood_Park.Slope                  0.078565 .
neighbourhood_Parkchester                 0.569059
neighbourhood_Pelham.Bay                  0.735713
neighbourhood_Pelham.Gardens              0.070952 .
neighbourhood_Port.Morris                 0.042408 *
neighbourhood_Port.Richmond               0.012703 *
neighbourhood_Prince.s.Bay                0.020466 *
neighbourhood_Prospect.Heights            0.152799
neighbourhood_Prospect.Lefferts.Gardens   0.001226 **
neighbourhood_Queens.Village              0.005181 **
neighbourhood_Randall.Manor               0.007055 **
neighbourhood_Red.Hook                    0.018493 *
neighbourhood_Rego.Park                   0.627479
neighbourhood_Richmond.Hill               0.057464 .
neighbourhood_Richmondtown                      NA
neighbourhood_Ridgewood                   0.120156
neighbourhood_Riverdale                   0.892518
neighbourhood_Rockaway.Beach              0.116280
```

```
neighbourhood_Roosevelt.Island          0.000329 ***
neighbourhood_Rosebank                  0.326834
neighbourhood_Rosedale                  0.000184 ***
neighbourhood_Rossville                 0.002611 **
neighbourhood_Schuylerville             0.612908
neighbourhood_Sea.Gate                  0.302571
neighbourhood_Sheepshead.Bay            0.000572 ***
neighbourhood_Shore.Acres               0.051566 .
neighbourhood_Silver.Lake                     NA
neighbourhood_SoHo                      0.162421
neighbourhood_Soundview                 0.085068 .
neighbourhood_South.Beach               0.229306
neighbourhood_South.Ozone.Park          0.871644
neighbourhood_South.Slope               0.239594
neighbourhood_Springfield.Gardens       5.15e-06 ***
neighbourhood_Spuyten.Duyvil            0.174166
neighbourhood_St..Albans                1.81e-05 ***
neighbourhood_St..George                0.104863
neighbourhood_Stapleton                 0.135106
neighbourhood_Stuyvesant.Town           0.036825 *
 [ reached getOption("max.print") -- omitted 34 rows ]
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 44.2 on 21755 degrees of freedom
Multiple R-squared:  0.5529,    Adjusted R-squared:  0.5484
F-statistic: 121.2 on 222 and 21755 DF,  p-value: < 2.2e-16
```

Hide

```
# Calculate the Mean Squared Error (MSE)
MSE <- mean((Y.test - first_prediction)^2)

# Print the resulting MSE
print(paste0("The resulting MSE is: ", MSE))
```

```
[1] "The resulting MSE is: 2009.63774609866"
```

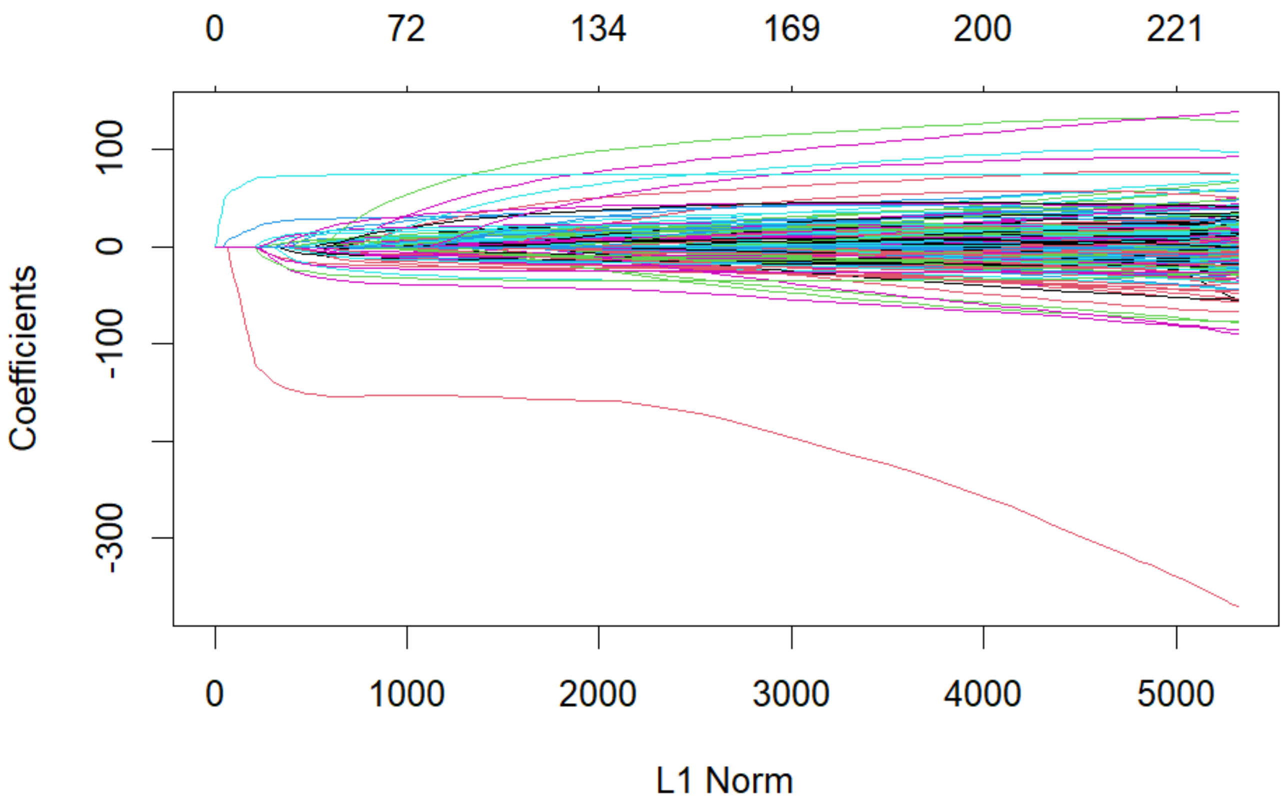
Hide

```
library(glmnet)
```

```
Loading required package: Matrix

Attaching package: ‘Matrix’

The following objects are masked from ‘package:tidyr’:

    expand, pack, unpack

Loaded glmnet 4.1-6
```

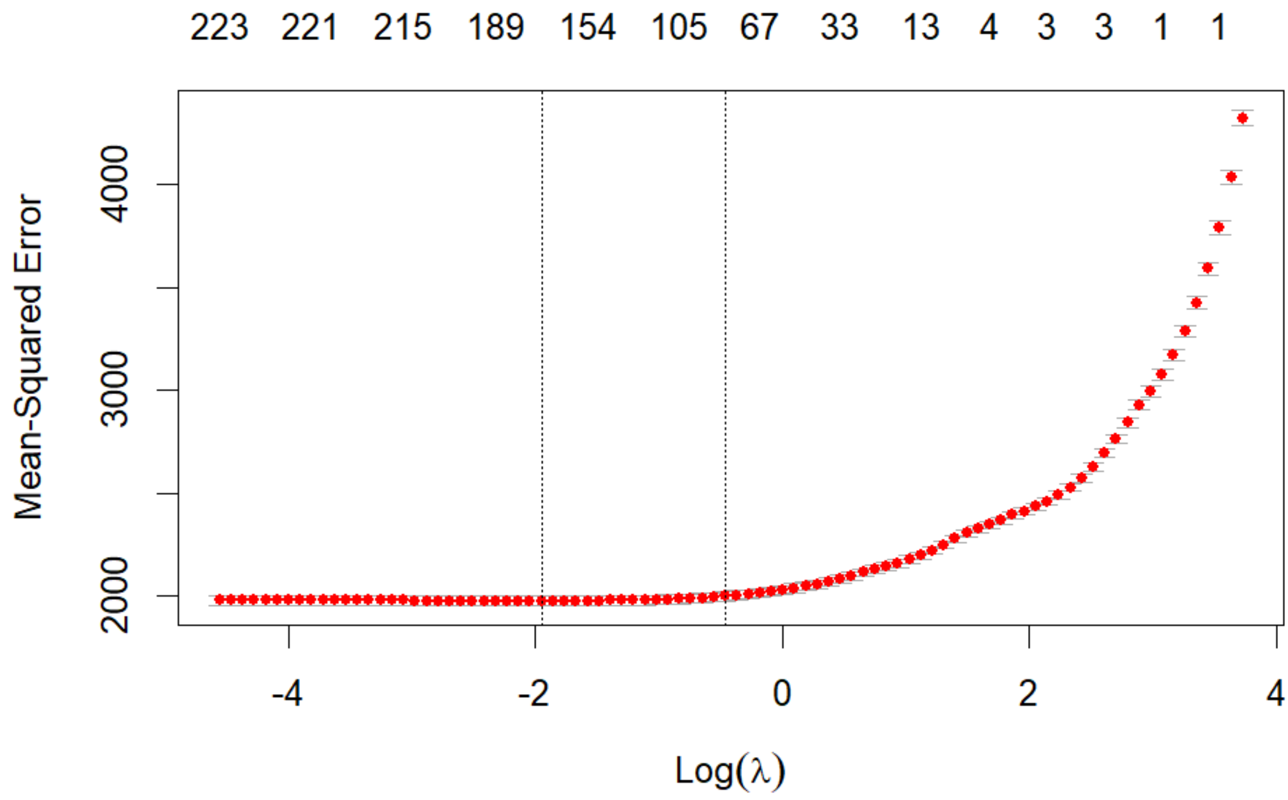
Hide

```
## Lasso
x=Airbnb_data_cleaned1[, -which(names(Airbnb_data_cleaned1) == "price")]
y=Airbnb_data_cleaned1$price

#Matrix Generation
X.train_M<-data.matrix(X.train)
Y.train_M<-data.matrix(Y.train)
X.test_M<-data.matrix(X.test)
Y.test_M<-data.matrix(Y.test)
# alpha=1 for Lasso
lasso.mod=glmnet(x=X.train_M,y=Y.train_M,alpha=1)
plot(lasso.mod)
```



Hide

```
# Use CV to calculate test error
set.seed(1)
cv.out=cv.glmnet(x=X.train_M,y=Y.train_M,alpha=1)
plot(cv.out)
```



Hide

```
bestlambda <- cv.out$lambda.min
lasso.pred=predict(lasso.mod,s=bestlambda ,newx=X.test_M)
```

Hide

```
# Several coefficients are exactly zero
out=glmnet(x=X.train_M,y=Y.train_M,alpha=1)
lasso.coef=predict(out,type="coefficients",s=bestlambda)
mse_lasso <- mean((lasso.pred - Y.test)^2)
# Print the resulting MSE
print(paste0("The resulting LASSO MSE is: ", mse_lasso))
```

```
[1] "The resulting LASSO MSE is: 2007.0501017308"
```

Hide

```
bestlambda <- cv.out$lambda.min
bestlambda
```

```
[1] 0.1421658
```

Hide

```
library(Metrics)
# Print the resulting R2
lasso_r2 <- cor(Y.test_M, lasso.pred)^2
print(paste0("The resulting LASSO R2 is: ", lasso_r2))
```

```
[1] "The resulting LASSO R2 is: 0.535457531068209"
```

Hide

```
# Calculate the adjusted R2
n_lasso <- nrow(X.test_M)
p_lasso <- ncol(X.test_M) - 1
lasso_adjR2 <- 1 - ((1 - lasso_r2) * (n_lasso - 1)) / (n_lasso - p_lasso - 1)
print(paste0("The resulting LASSO adjusted R2 is: ", lasso_adjR2))
```

```
[1] "The resulting LASSO adjusted R2 is: 0.514975993099343"
```

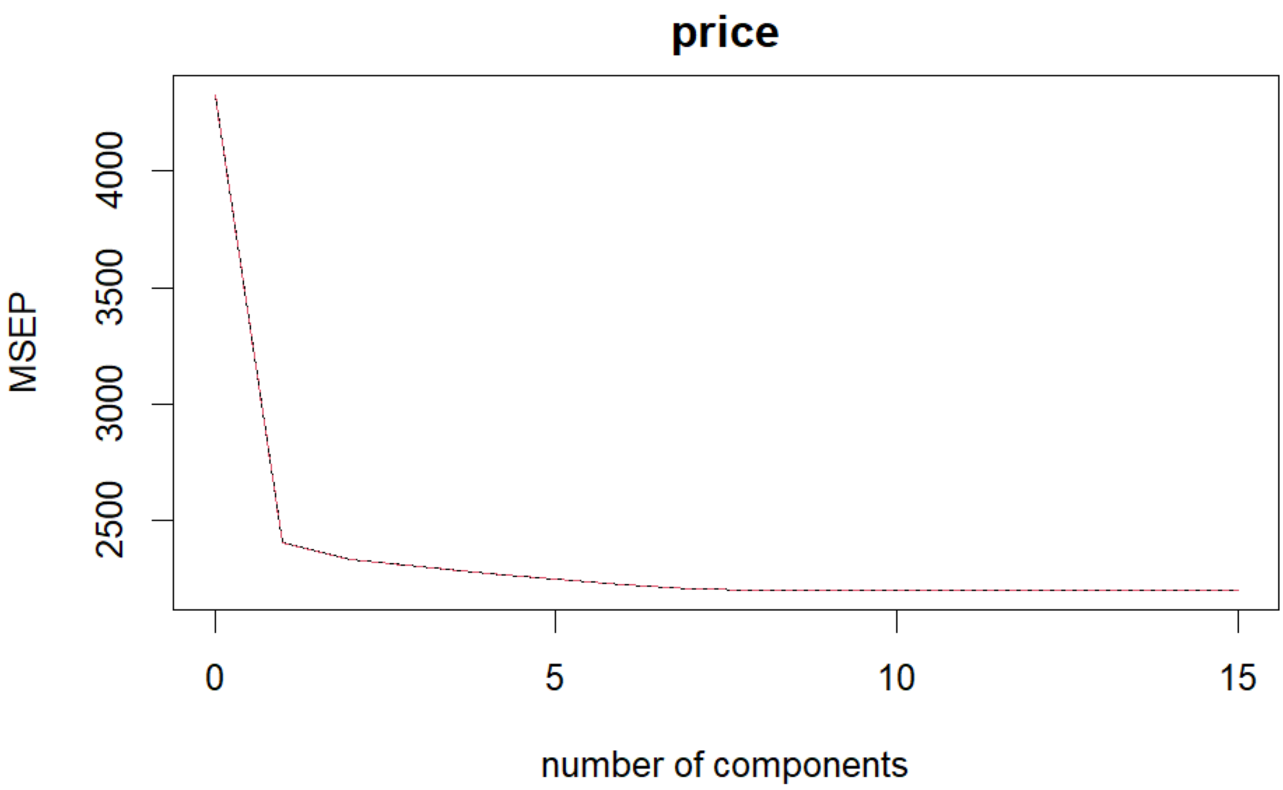
Hide

```
library(pls)

#a) PLS with cross-validation to optimize M
training_data_filtered <- training_data[, -nearZeroVar(training_data)]
set.seed(1)
pls.cv = plsr(price ~ ., data = training_data_filtered, scale = TRUE, validation = "CV")
summary(pls.cv)
```

```
Data:   X dimension: 21978 15
    Y dimension: 21978 1
Fit method: kernelpls
Number of components considered: 15

VALIDATION: RMSEP
Cross-validated using 10 random segments.
       (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps
CV           65.77    49.05     48.3    47.97     47.7    47.43
adjCV        65.77    49.05     48.3    47.97     47.7    47.43
       6 comps  7 comps  8 comps  9 comps  10 comps  11 comps
CV       47.16    46.96    46.93    46.92     46.91     46.91
adjCV    47.16    46.96    46.92    46.92     46.91     46.91
       12 comps  13 comps  14 comps  15 comps
CV        46.91     46.91     46.91     46.91
adjCV     46.91     46.91     46.91     46.91

TRAINING: % variance explained
       1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
X        15.88    30.63    40.24    48.28    53.74    57.39    60.81
price    44.41    46.10    46.85    47.47    48.07    48.65    49.08
       8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
X        66.57    73.31      76.2     81.12     86.55     91.06
price    49.16    49.18      49.2     49.20     49.20     49.20
       14 comps  15 comps
X         93.63     100.0
price     49.20      49.2
```

Hide

```
# Plot the validation curve and extract the optimal number of components (M)
validationplot(pls.cv, val.type = "MSEP")
```



Hide

```
optM <- 9
```

Hide

```
optM <- 9
testing_data_filtered <- testing_data[, -nearZeroVar(testing_data)]

pls.pred = predict(pls.cv, testing_data_filtered, ncomp = optM)
mse = mean((pls.pred - Y.test)^2)
cat("MSE using optimal M =", mse, "\n")
```

```
MSE using optimal M = 2210.67
```

Hide

```
# Refit the model using all the data and the optimal M
pls.fit = plsr(price ~ ., data = training_data_filtered, scale = TRUE, ncomp = optM)
summary(pls.fit)
```

```
Data:   X dimension: 21978 15
    Y dimension: 21978 1
Fit method: kernelpls
Number of components considered: 9
TRAINING: % variance explained
       1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
X        15.88    30.63    40.24    48.28    53.74    57.39    60.81
price    44.41    46.10    46.85    47.47    48.07    48.65    49.08
       8 comps  9 comps
X        66.57    73.31
price    49.16    49.18
```

Hide

```
# Calculate the R2
r2_pls_set <- R2(pls.fit)
r2_pls_set
```

```
(Intercept)      1 comps      2 comps      3 comps      4 comps
     0.0000       0.4441       0.4610       0.4685       0.4747
    5 comps      6 comps      7 comps      8 comps      9 comps
     0.4807       0.4865       0.4908       0.4916       0.4918
```

Hide

```
r2_pls<-0.4918
cat("PLS R2 =", r2_pls, "\n")
```

```
PLS R2 = 0.4918
```

Hide

```
# Calculate the adjusted R2
n_pls <- nrow(testing_data_filtered)
p_pls <- ncol(testing_data_filtered) - 1
adj_r2_pls <- 1 - ((1 - r2_pls) * (n_pls - 1)) / (n_pls - p_pls - 1)
cat("PLS Adjusted R2 =", adj_r2_pls, "\n")
```

```
PLS Adjusted R2 = 0.4904087
```

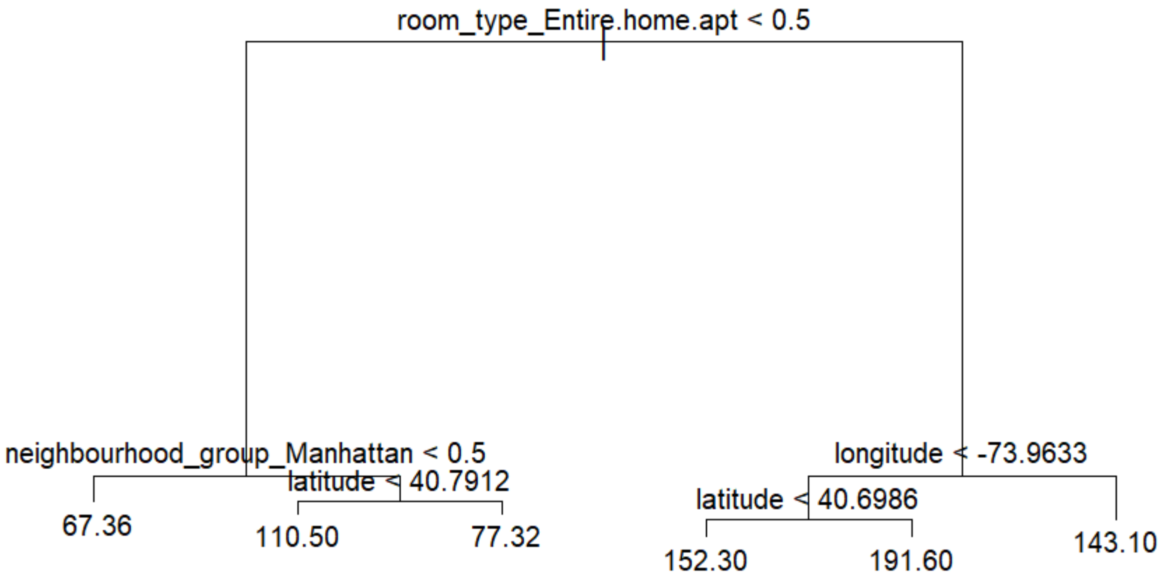
Hide

```
library(tree)
library(MASS)
set.seed(1)
tree.airbnb=tree(price~.,training_data)
# Only a few of the variables were used in constructing the tree
# lstat: percentage of individuals with lower socioeconomic status
summary(tree.airbnb)
```

```
Regression tree:
tree(formula = price ~ ., data = training_data)
Variables actually used in tree construction:
[1] "room_type_Entire.home.apt"     "neighbourhood_group_Manhattan"
[3] "latitude"                      "longitude"
Number of terminal nodes:  6
Residual mean deviance:  2221 = 48790000 / 21970
Distribution of residuals:
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-175.600  -27.360   -7.357    0.000   19.680  251.600
```

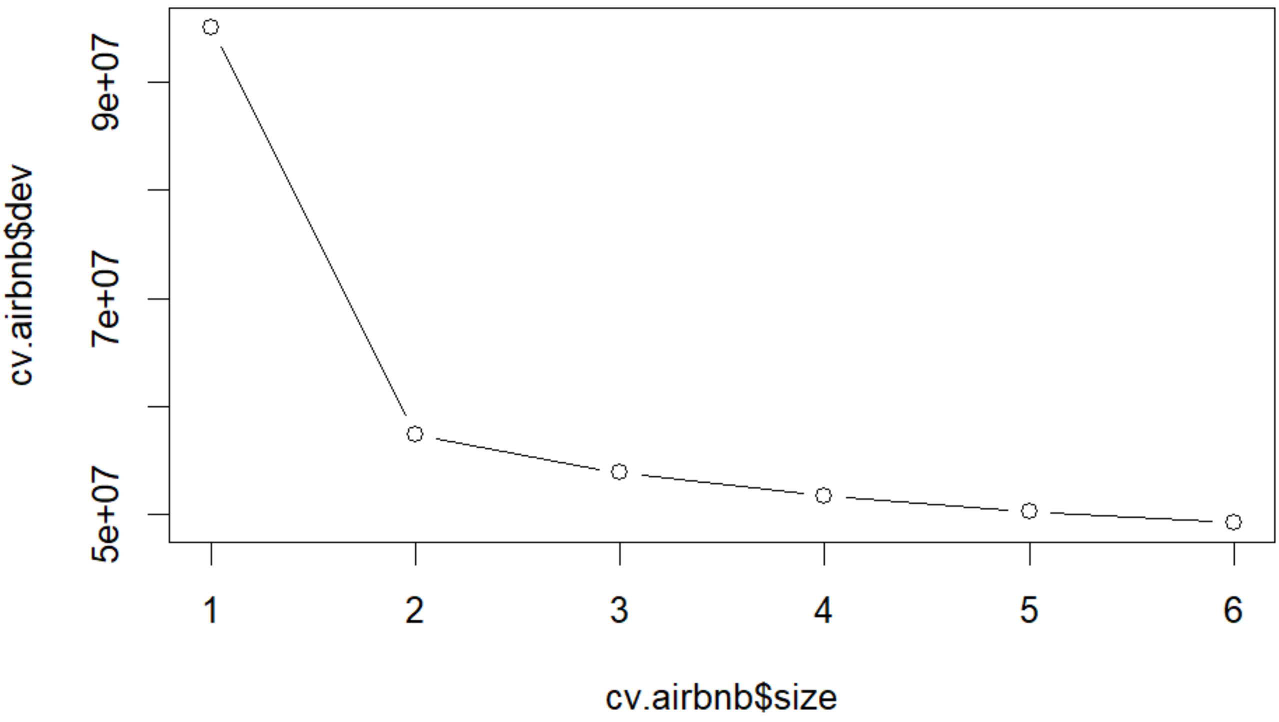
Hide

```
# Plot the tree
plot(tree.airbnb)
text(tree.airbnb,pretty=0,cex=0.75)
```



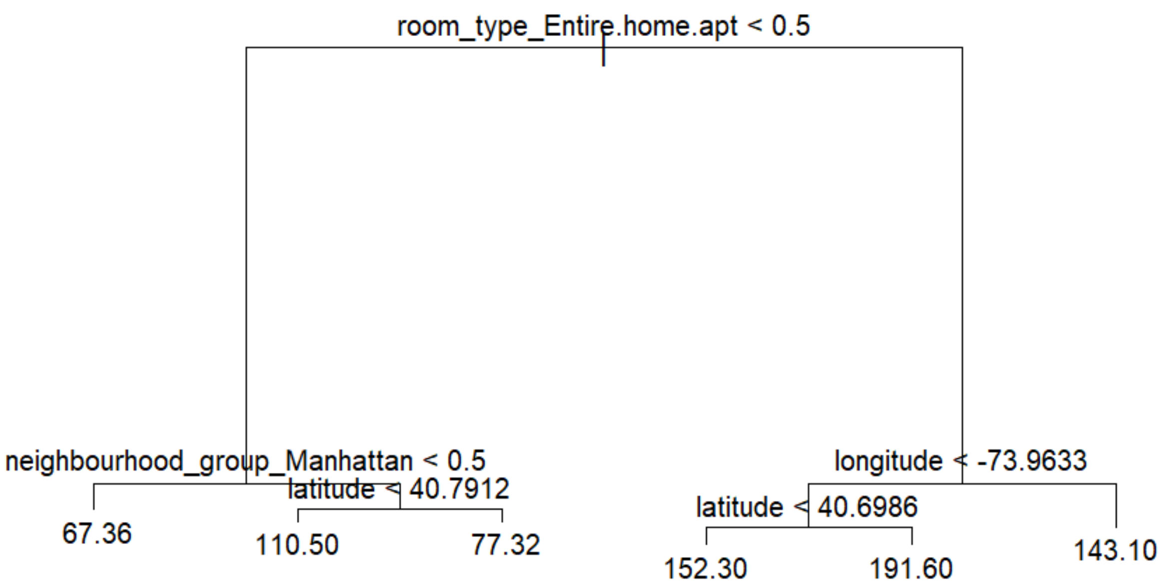
Hide

```
# cv.tree() to determine whether pruning improves performance
cv.airbnb=cv.tree(tree.airbnb)
# It doesn't seem to be the case
plot(cv.airbnb$size,cv.airbnb$dev,type="b")
```



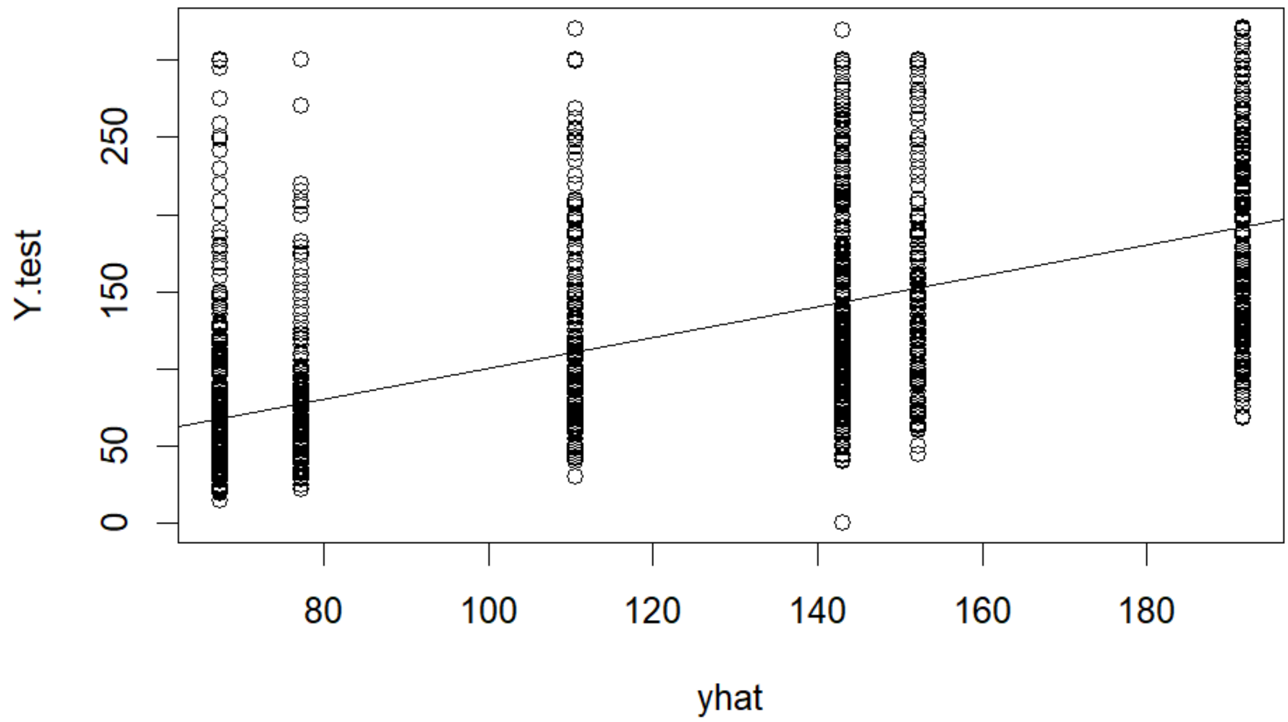
Hide

```
# prune.tree(): function to prune to be used in case we wanted to prune the tree
prune.airbnb=prune.tree(tree.airbnb,best=6)
plot(prune.airbnb)
text(prune.airbnb,pretty=0,cex=0.75)
```



Hide

```
# Predicting based on CV results (i.e., use the unpruned tree)
yhat=predict(tree.airbnb,testing_data)
plot(yhat,Y.test)
abline(0,1)
```



Hide

```
# Test error
mse_tree=mean((yhat-Y.test)^2)
mse_tree
```

```
[1] 2212.729
```

Hide

```
# Calculate the R2
ybar_tree <- mean(Y.test)

# Calculate the total sum of squares (SST)
SST_tree <- sum((Y.test - ybar_tree)^2)

# Calculate the residual sum of squares (SSE)
SSE_tree <- sum((Y.test - yhat)^2)

# Calculate the R-squared value
r2_tree <- 1 - SSE_tree/SST_tree

# Print the R-squared value
cat("R2_tree =", r2_tree, "\n")
```

```
R2_tree = 0.4877283
```

Hide

```
# Calculate the adjusted R2
n_tree <- nrow(testing_data)
p_tree <- ncol(testing_data) - 1
adj_r2_tree <- 1 - ((1 - r2_tree) * (n_tree - 1)) / (n_tree - p_tree - 1)
cat("Adjusted R2_tree =", adj_r2_tree, "\n")
```

```
Adjusted R2_tree = 0.4650407
```

Hide

```
##################
# Random Forests #
##################
library(randomForest)
```

```
randomForest 4.7-1.1
Type rfNews() to see new features/changes/bug fixes.
```

Hide

```
# By default randomForest() uses m=p/3 for regression and m=sqrt(p) for classification
# Let's try m=6
set.seed(1)
rf.airbnb=randomForest(price~.,training_data,mtry=233/3,importance =T)
yhat.rf = predict(rf.airbnb ,testing_data)

mse.rf<-mean((yhat.rf-Y.test)^2)
mse.rf
```

```
[1] 1838.123
```

Hide

```
# Calculate the R2
ybar_rf <- mean(Y.test)

# Calculate the total sum of squares (SST)
SST_rf <- sum((Y.test - ybar_rf)^2)

# Calculate the residual sum of squares (SSE)
SSE_rf <- sum((Y.test - yhat.rf)^2)

# Calculate the R-squared value
r2_rf <- 1 - SSE_rf/SST_rf
cat("RF R2 =", r2_rf, "\n")
```

```
RF R2 = 0.5744539
```

Hide

```
# Calculate the adjusted R2
n_rf <- nrow(testing_data)
p_rf <- ncol(testing_data) - 1
adj_r2_rf <- 1 - ((1 - r2_rf) * (n_rf - 1)) / (n_rf - p_rf - 1)
cat("Adjusted R2_rf =", adj_r2_rf, "\n")
```

```
Adjusted R2_rf = 0.5556073
```

Hide

```
# importance(): view the importance of each variable
# %IncMSE: mean decrease of accuracy in predictions on the OOB samples when a
# given variable is excluded from the model
# IncNodeImpurity: total decrease in node impurity that results from splits over
# that variable, averaged over all trees (RSS in regr. vs. deviance in class)
importance(rf.airbnb)
```

```
                                           %IncMSE IncNodePurity
latitude                               43.371334827  9.789892e+06
longitude                              80.975949091  1.256811e+07
minimum_nights                         24.095100138  2.435539e+06
number_of_reviews                      51.604432367  4.115502e+06
reviews_per_month                      59.334688230  5.273404e+06
calculated_host_listings_count         65.600834895  1.784999e+06
availability_365                      135.156486951  6.136025e+06
neighbourhood_group_Bronx               6.991506151  1.135749e+05
neighbourhood_group_Brooklyn           11.173457484  5.076947e+05
neighbourhood_group_Manhattan          19.616796938  2.573367e+06
neighbourhood_group_Queens             11.967844559  3.816479e+05
neighbourhood_group_Staten.Island       5.549771964  4.296883e+04
neighbourhood_Allerton                 -5.637417244  4.372362e+03
neighbourhood_Arden.Heights             1.001001503  1.450354e+02
neighbourhood_Arrochar                 -5.169360244  4.057268e+03
neighbourhood_Arverne                  -3.454671779  5.192061e+04
neighbourhood_Astoria                  -4.055116322  5.846700e+04
neighbourhood_Bath.Beach               -2.787340886  1.941727e+03
neighbourhood_Battery.Park.City        -0.825938437  1.099067e+04
neighbourhood_Bay.Ridge                -2.254343272  3.100933e+04
neighbourhood_Bay.Terrace               2.011228419  7.009908e+02
neighbourhood_Bay.Terrace..Staten.Island   0.000000000  0.000000e+00
neighbourhood_Baychester                0.000000000  8.547982e+01
neighbourhood_Bayside                   1.965940705  2.038447e+04
neighbourhood_Bayswater                -2.295603981  1.370739e+03
neighbourhood_Bedford.Stuyvesant       12.629750065  1.374646e+05
neighbourhood_Belle.Harbor              6.583837995  2.104765e+04
neighbourhood_Bellerose                -0.398309726  1.572619e+03
neighbourhood_Belmont                   5.248958072  2.848888e+04
neighbourhood_Bensonhurst               2.175066924  7.891183e+03
neighbourhood_Bergen.Beach             -4.491010712  5.157063e+03
neighbourhood_Boerum.Hill              -4.936941136  3.153001e+04
neighbourhood_Borough.Park             -2.535487317  2.014977e+04
neighbourhood_Breezy.Point              6.232410329  1.206842e+04
neighbourhood_Briarwood                 4.677909377  1.156359e+04
neighbourhood_Brighton.Beach           -2.333406570  1.370777e+04
neighbourhood_Bronxdale                 2.959498267  1.190121e+03
neighbourhood_Brooklyn.Heights          2.940599340  5.028936e+04
neighbourhood_Brownsville              -3.911768550  1.362858e+04
neighbourhood_Bull.s.Head              -0.102559661  2.754293e+02
neighbourhood_Bushwick                  8.845534222  8.812233e+04
neighbourhood_Cambria.Heights          -4.072238475  1.419263e+04
neighbourhood_Canarsie                 -5.229593399  2.360376e+04
neighbourhood_Carroll.Gardens           0.912784391  3.440540e+04
neighbourhood_Castle.Hill               0.000000000  2.288872e+02
neighbourhood_Castleton.Corners        -7.281181706  1.884914e+04
neighbourhood_Chelsea                   3.839679150  8.513572e+04
neighbourhood_Chinatown                -6.442144530  5.578035e+04
neighbourhood_City.Island              -1.337987832  1.802770e+03
neighbourhood_Civic.Center             -3.752141522  1.058427e+04
neighbourhood_Claremont.Village        -4.815186944  1.303202e+03
neighbourhood_Clason.Point             -6.442164266  2.071237e+04
neighbourhood_Clifton                  -2.528133830  4.092940e+03
neighbourhood_Clinton.Hill             16.832083747  1.029220e+05
```

```
neighbourhood_Co.op.City                 0.000000000  4.112921e+01
neighbourhood_Cobble.Hill               -3.245013889  2.197405e+04
neighbourhood_College.Point              0.580702445  1.939692e+03
neighbourhood_Columbia.St               -2.158627763  5.131281e+03
neighbourhood_Concord                   -5.298936913  4.521228e+03
neighbourhood_Concourse                  1.367437708  3.054399e+03
neighbourhood_Concourse.Village         -4.193336460  1.422398e+03
neighbourhood_Coney.Island              -2.601766497  2.062621e+04
neighbourhood_Corona                     1.520945632  6.052942e+03
neighbourhood_Crown.Heights             13.012628998  1.111155e+05
neighbourhood_Cypress.Hills             -1.830010556  1.238458e+04
neighbourhood_DUMBO                     -1.081546973  2.145851e+04
neighbourhood_Ditmars.Steinway          -1.200651099  2.569485e+04
neighbourhood_Dongan.Hills              -2.805887305  1.090341e+03
neighbourhood_Douglaston                 2.351055354  1.178140e+02
neighbourhood_Downtown.Brooklyn          0.649120515  1.358159e+04
neighbourhood_Dyker.Heights             -6.965768680  5.016860e+03
neighbourhood_East.Elmhurst             -1.575152666  1.146349e+04
neighbourhood_East.Flatbush             -0.622906781  5.089913e+04
neighbourhood_East.Harlem                9.407898685  7.296049e+04
neighbourhood_East.Morrisania           -9.667135132  6.448788e+03
neighbourhood_East.New.York             -7.519974095  4.636281e+04
neighbourhood_East.Village               8.995428279  1.159905e+05
neighbourhood_Eastchester               -4.351116005  3.773170e+03
neighbourhood_Edenwald                  -0.397107319  4.351703e+03
neighbourhood_Edgemere                  -0.307473972  1.428194e+03
neighbourhood_Elmhurst                  -8.024871436  4.486162e+04
neighbourhood_Eltingville                0.000000000  0.000000e+00
neighbourhood_Emerson.Hill               0.000000000  8.633557e+02
neighbourhood_Far.Rockaway              -5.824526207  6.705426e+03
neighbourhood_Fieldston                 -1.225679900  1.440988e+03
neighbourhood_Financial.District        11.619740972  5.329945e+04
neighbourhood_Flatbush                   6.801365491  7.555393e+04
neighbourhood_Flatiron.District          5.206655242  2.656291e+04
neighbourhood_Flatlands                 -6.834095518  3.097051e+04
neighbourhood_Flushing                  -1.388287222  3.035412e+04
neighbourhood_Fordham                    3.237608569  1.977200e+03
neighbourhood_Forest.Hills              -5.035936106  2.427481e+04
neighbourhood_Fort.Greene                0.985207145  4.615962e+04
neighbourhood_Fort.Hamilton              5.096064068  4.541910e+03
neighbourhood_Fresh.Meadows              1.533570059  3.135967e+04
neighbourhood_Glendale                  -5.054541896  2.052097e+04
neighbourhood_Gowanus                   -1.106720011  4.281435e+04
neighbourhood_Gramercy                  -5.273772478  6.335300e+04
neighbourhood_Graniteville              -1.410846129  1.263776e+02
neighbourhood_Grant.City                 1.168142577  7.703026e+02
neighbourhood_Gravesend                 -2.797479769  7.222252e+03
neighbourhood_Great.Kills               -4.036925931  8.150601e+03
neighbourhood_Greenpoint                11.483826972  7.604007e+04
neighbourhood_Greenwich.Village          2.700788030  4.461949e+04
neighbourhood_Grymes.Hill                2.567853379  4.078487e+02
neighbourhood_Harlem                     9.805603971  1.305609e+05
neighbourhood_Hell.s.Kitchen             6.828012920  1.361922e+05
neighbourhood_Highbridge                -3.584000975  5.979910e+03
neighbourhood_Hollis                    -1.306615435  3.234844e+02
neighbourhood_Holliswood                -9.337230568  2.134525e+04
```

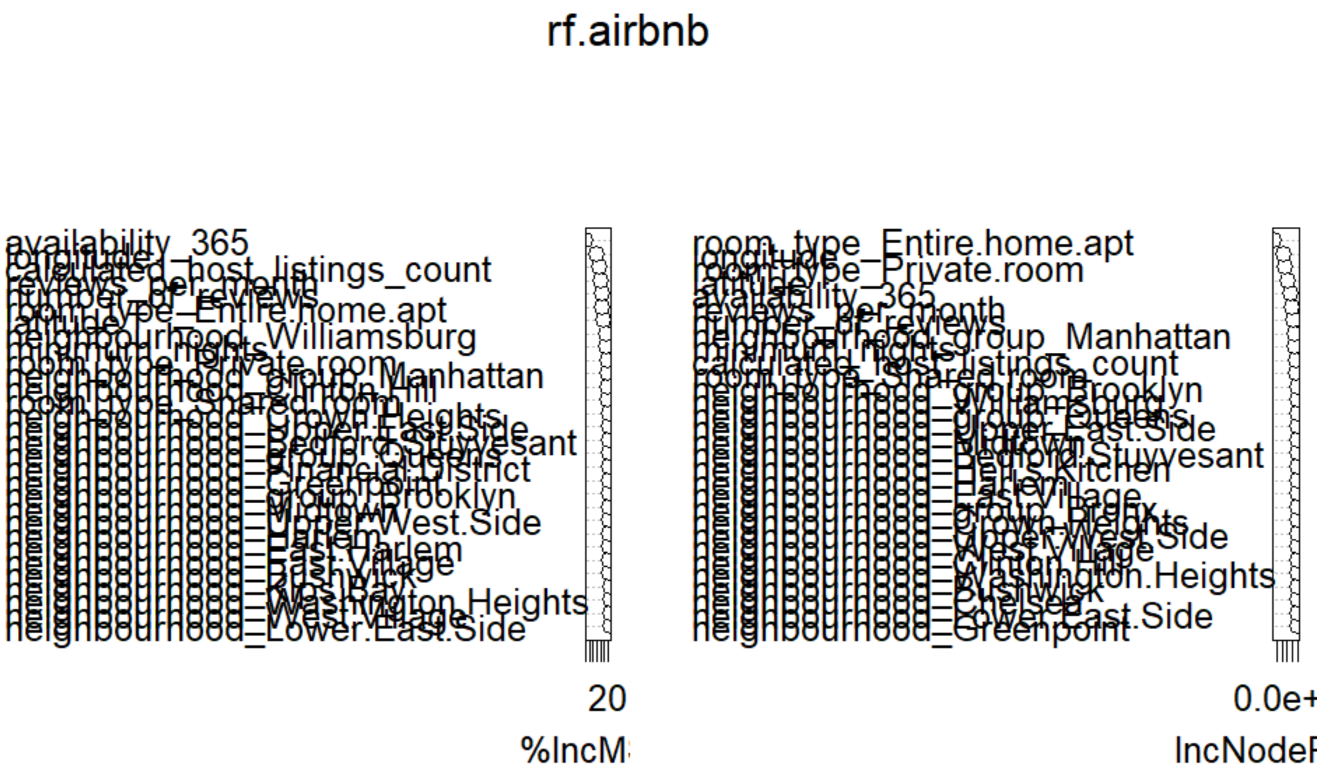
```
neighbourhood_Howard.Beach              3.761583951  8.196208e+03
neighbourhood_Howland.Hook              0.000000000  8.004673e+01
neighbourhood_Huguenot                  1.015054243  3.137673e+02
neighbourhood_Hunts.Point               3.412853934  1.302006e+03
neighbourhood_Inwood                    2.817692672  2.563136e+04
neighbourhood_Jackson.Heights          -3.688757712  2.142309e+04
neighbourhood_Jamaica                  -6.147675257  3.834141e+04
neighbourhood_Jamaica.Estates          -1.044723097  9.177589e+02
neighbourhood_Jamaica.Hills             0.000000000  1.383895e+02
neighbourhood_Kensington                0.919436524  2.287026e+04
neighbourhood_Kew.Gardens               1.179273953  7.802866e+02
neighbourhood_Kew.Gardens.Hills        -5.048848508  6.714312e+03
neighbourhood_Kingsbridge              -1.189635398  2.759990e+03
neighbourhood_Kips.Bay                  8.104145421  5.459797e+04
neighbourhood_Laurelton                -5.252278826  1.178209e+04
neighbourhood_Lighthouse.Hill           0.000000000  2.460713e+03
neighbourhood_Little.Italy              5.229057745  1.227892e+04
neighbourhood_Little.Neck               1.221713236  1.216344e+02
neighbourhood_Long.Island.City          4.046050557  4.743894e+04
neighbourhood_Longwood                 -5.658805628  5.595665e+03
neighbourhood_Lower.East.Side           7.093465204  7.847214e+04
neighbourhood_Manhattan.Beach           0.950917875  2.238484e+03
neighbourhood_Marble.Hill              -6.029893868  1.351154e+04
neighbourhood_Mariners.Harbor          -0.999072763  9.313883e+02
neighbourhood_Maspeth                  -0.918253424  9.072717e+03
neighbourhood_Melrose                  -0.570190981  1.227200e+03
neighbourhood_Middle.Village           -2.351364795  4.418533e+03
neighbourhood_Midland.Beach            -2.086263548  2.672433e+02
neighbourhood_Midtown                  10.172159811  2.118463e+05
neighbourhood_Midwood                   1.359935891  2.652556e+04
neighbourhood_Mill.Basin               -3.153854468  2.718585e+03
neighbourhood_Morningside.Heights       6.445455652  3.372579e+04
neighbourhood_Morris.Heights           -5.977787731  2.269668e+03
neighbourhood_Morris.Park              -1.195036926  1.229374e+03
neighbourhood_Morrisania                1.001001503  6.635737e+02
neighbourhood_Mott.Haven               -0.977454080  2.841520e+03
neighbourhood_Mount.Eden                0.000000000  1.280282e+03
neighbourhood_Mount.Hope               -4.152237215  9.499280e+03
neighbourhood_Murray.Hill               5.872561428  5.002112e+04
neighbourhood_Navy.Yard                -0.841556219  1.471294e+04
neighbourhood_Neponsit                  1.292441981  6.485104e+03
neighbourhood_New.Brighton             -3.696479732  1.252145e+03
neighbourhood_New.Dorp.Beach            0.000000000  1.792919e+02
neighbourhood_New.Springville           2.664124394  4.880599e+01
neighbourhood_NoHo                     -5.301618194  1.595421e+04
neighbourhood_Nolita                   -0.000754385  5.247086e+04
neighbourhood_North.Riverdale           1.865974443  2.669723e+02
neighbourhood_Norwood                  -8.022577043  6.226054e+03
neighbourhood_Oakwood                  -1.226158808  4.019629e+02
neighbourhood_Olinville                -1.394029568  4.979036e+02
neighbourhood_Ozone.Park                0.636497093  6.163472e+03
neighbourhood_Park.Slope                3.072006049  5.488341e+04
neighbourhood_Parkchester              -6.838175488  1.662043e+04
neighbourhood_Pelham.Bay               -6.012470612  1.184673e+04
neighbourhood_Pelham.Gardens           -0.971018288  2.581186e+03
neighbourhood_Port.Morris               0.716500902  5.078125e+03
```

```
neighbourhood_Port.Richmond                3.595228218  2.426064e+02
neighbourhood_Prince.s.Bay                -3.454422030  1.696477e+03
neighbourhood_Prospect.Heights             0.189263446  4.428931e+04
neighbourhood_Prospect.Lefferts.Gardens    5.506846778  4.239749e+04
neighbourhood_Queens.Village               1.109922439  7.701828e+03
neighbourhood_Randall.Manor                0.338126015  1.619472e+03
neighbourhood_Red.Hook                    -0.765219710  1.040102e+04
neighbourhood_Rego.Park                   -3.902468409  2.246565e+04
neighbourhood_Richmond.Hill               -3.116216449  2.765620e+04
neighbourhood_Richmondtown                 0.000000000  0.000000e+00
neighbourhood_Ridgewood                   -2.668229557  2.677286e+04
neighbourhood_Riverdale                   -4.676117767  6.717082e+03
neighbourhood_Rockaway.Beach               5.996247140  1.255300e+04
neighbourhood_Roosevelt.Island            -4.968080476  2.218410e+04
neighbourhood_Rosebank                    -2.203190021  2.187771e+03
neighbourhood_Rosedale                    -2.567427365  8.007200e+03
neighbourhood_Rossville                    0.000000000  3.855308e+01
neighbourhood_Schuylerville                2.101277542  1.914311e+02
neighbourhood_Sea.Gate                     0.000000000  6.544964e+03
neighbourhood_Sheepshead.Bay              -2.335584034  1.695365e+04
neighbourhood_Shore.Acres                  2.694128756  2.545424e+03
neighbourhood_Silver.Lake                  0.000000000  0.000000e+00
neighbourhood_SoHo                        -8.664219372  6.204991e+04
neighbourhood_Soundview                    1.803733490  1.516607e+03
neighbourhood_South.Beach                 -3.195100635  1.329018e+04
neighbourhood_South.Ozone.Park             2.191312035  1.580247e+03
neighbourhood_South.Slope                  1.901578861  4.725400e+04
neighbourhood_Springfield.Gardens         -3.240673437  2.195683e+04
neighbourhood_Spuyten.Duyvil               0.000000000  4.944998e+01
neighbourhood_St..Albans                  -6.981006864  3.053556e+04
neighbourhood_St..George                   1.248998082  2.231889e+03
neighbourhood_Stapleton                   -5.730408231  6.730417e+03
neighbourhood_Stuyvesant.Town              1.548348620  2.385757e+03
neighbourhood_Sunnyside                    3.616061981  2.137268e+04
neighbourhood_Sunset.Park                  3.221707620  5.564323e+04
neighbourhood_Theater.District            -6.238403593  4.096432e+04
neighbourhood_Throgs.Neck                 -7.406739968  3.790412e+03
neighbourhood_Todt.Hill                    0.000000000  3.538381e+02
neighbourhood_Tompkinsville               -5.972013352  5.379898e+03
neighbourhood_Tottenville                 -0.962053335  1.094009e+04
neighbourhood_Tremont                     -3.114718252  2.158044e+03
neighbourhood_Tribeca                      6.517903983  5.296349e+04
neighbourhood_Two.Bridges                 -4.217942661  1.985577e+04
neighbourhood_Unionport                    0.000000000  3.767160e+01
neighbourhood_University.Heights          -1.292699986  2.099234e+03
neighbourhood_Upper.East.Side             12.939810455  2.187377e+05
neighbourhood_Upper.West.Side              9.916200872  1.088313e+05
neighbourhood_Van.Nest                     1.293499213  8.695216e+02
neighbourhood_Vinegar.Hill                 0.592898548  2.608052e+04
neighbourhood_Wakefield                   -3.768203882  5.405440e+03
neighbourhood_Washington.Heights           7.952588879  1.028577e+05
neighbourhood_West.Brighton                0.866542207  1.010379e+03
neighbourhood_West.Farms                   0.000000000  9.673679e+02
neighbourhood_West.Village                 7.884719574  1.048350e+05
neighbourhood_Westchester.Square          -2.224952102  9.042705e+02
neighbourhood_Westerleigh                  0.000000000  1.185382e+02
```

```
neighbourhood_Whitestone                -4.202455008  5.762306e+03
neighbourhood_Williamsbridge            -4.737982778  6.851476e+03
neighbourhood_Williamsburg              24.615430245  4.670360e+05
neighbourhood_Willowbrook                0.000000000  7.008329e+03
neighbourhood_Windsor.Terrace           -2.349563828  3.754243e+04
neighbourhood_Woodhaven                  0.388175304  8.182715e+03
neighbourhood_Woodlawn                   1.008410122  6.321240e+02
neighbourhood_Woodside                  -5.723326149  2.978485e+04
room_type_Entire.home.apt               46.382082604  2.228962e+07
room_type_Private.room                  20.969286208  1.228874e+07
room_type_Shared.room                   16.807852550  1.161151e+06
```

Hide

```
# varImpPlot(): Variance importance plot
varImpPlot(rf.airbnb)
```



Hide

```
# gbm: library for boosting
library(gbm)
```

```
Warning: package ‘gbm’ was built under R version 4.2.3Loaded gbm 2.1.8.1
```

Hide

```
set.seed(1)
# Since this is a regression problem, we set the distribution to "gaussian"
# For binary classification, we would use "bernoulli"
# n.trees: number of trees we want
# interaction.depth: limits the depth of each tree
boost.airbnb=gbm(price~.,data=training_data,distribution="gaussian",n.trees=5000, interactio
n.depth=4)
```

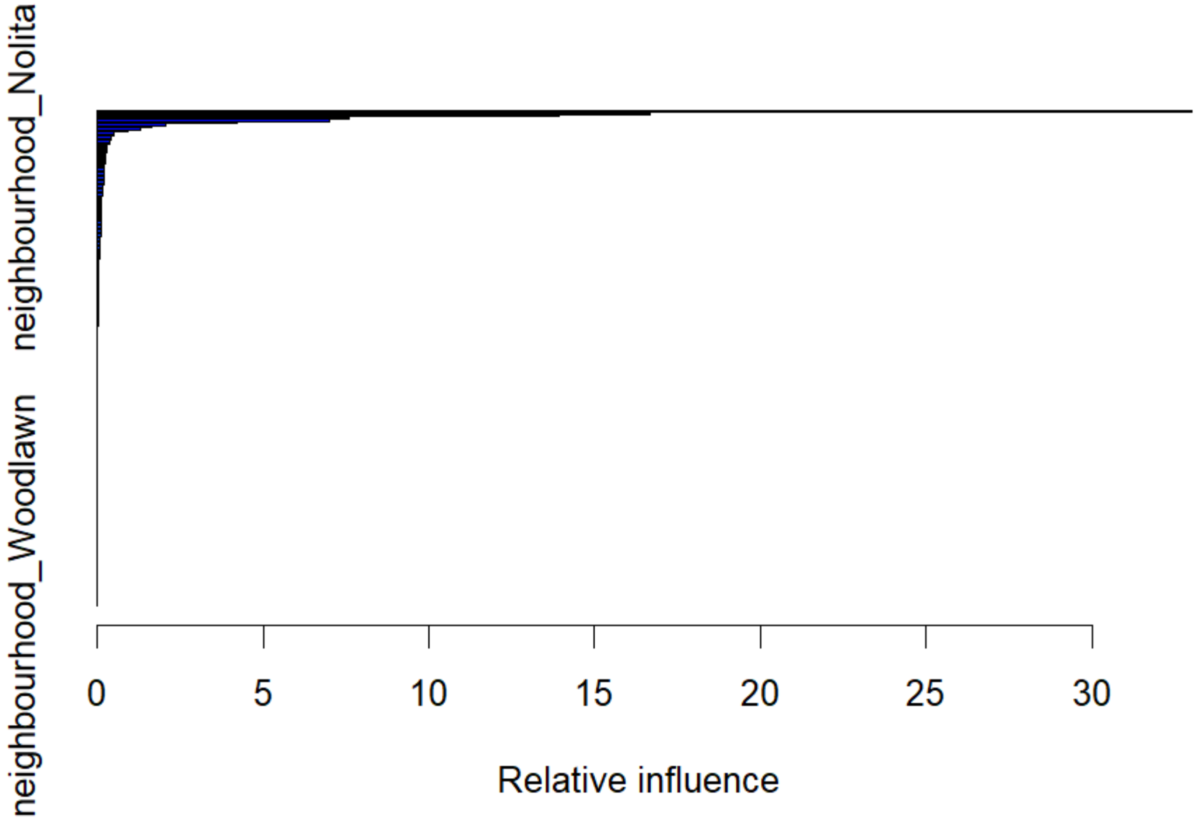
```
Warning: variable 22: neighbourhood_Bay.Terrace..Staten.Island has no variation.Warning: vari
able 82: neighbourhood_Eltingville has no variation.Warning: variable 176: neighbourhood_Rich
mondtown has no variation.Warning: variable 188: neighbourhood_Silver.Lake has no variation.
```

Hide

```
# In this case, summary() produces the relative influence plot and outputs
# the relative influence statistics
summary(boost.airbnb)
```

	var <chr>
room_type_Entire.home.apt	room_type_Entire.home.apt
longitude	longitude
latitude	latitude
availability_365	availability_365
reviews_per_month	reviews_per_month
number_of_reviews	number_of_reviews
minimum_nights	minimum_nights
neighbourhood_group_Manhattan	neighbourhood_group_Manhattan
calculated_host_listings_count	calculated_host_listings_count
room_type_Private.room	room_type_Private.room

1-10 of 233 rows Previous 1 2 3 4 5 6 ... 24 Next



Hide

```
#MSE Test
# Performance on the test set
yhat.boost=predict(boost.airbnb,testing_data,n.trees=5000)
mean((yhat.boost -Y.test)^2)
```

```
[1] 1957.024
```

Hide

```
# Calculate the R2
ybar_boost <- mean(Y.test)

# Calculate the total sum of squares (SST)
SST_boost <- sum((Y.test - ybar_boost)^2)

# Calculate the residual sum of squares (SSE)
SSE_boost <- sum((Y.test - yhat.boost)^2)

# Calculate the R-squared value
r2_boost <- 1 - SSE_boost/SST_boost
cat("Boosting R2 =", r2_boost, "\n")
```

```
Boosting R2 = 0.5469268
```

Hide

```
# Calculate the adjusted R2
n_boost <- nrow(testing_data)
p_boost <- ncol(testing_data) - 1
adj_r2_boost <- 1 - ((1 - r2_boost) * (n_boost - 1)) / (n_boost - p_boost - 1)
cat("Adjusted R2_boost =", adj_r2_boost, "\n")
```

```
Adjusted R2_boost = 0.5268611
```