GROUP ASSIGNMENT

TECHNOLOGY PARK MALAYSIA

CT127-3-2-PFDA

PROGRAMMING FOR DATA ANALYSIS

HAND OUT DATE: 06 JULY 2023

HAND IN DATE: 1 SEPTEMBER 2023

WEIGHTAGE: 50%

INSTRUCTIONS TO CANDIDATES:

1 Students are advised to underpin their answers with the use of references (cited using the American Psychological Association (APA) Referencing).

2 Late submission will be awarded zero (0) unless Extenuating Circumstances (EC) are upheld.

3 Cases of plagiarism will be penalized.

4 Submit the assignment to APU Learning Management System.

5 You must obtain 50% overall to pass this module.

Name	TP Number
Puah Yi Kai	TP069207
Wong Shi Wei	TP063736
William Hamilton Otieno Odhiambo	TP063033

Contents

1.0 Abstract

This report discusses about the analysis of a House Rent dataset using R programming. Using R language, we completed the full procedure of data analysis including data import, data exploration, data analysis, data visualisation, etc. In light of that, we introduced some data analysis techniques, and some data visualisation graphs and packages in R Studio. This report presents a data analysis of a house rent dataset in India using R programming. The dataset contains various features of houses such as size, BHK, furnishing status, area type, etc. and their corresponding rent prices in six different cities. The main objectives of this report are to explore the relationship between rent and other factors, and to build and evaluate different predictive models such as linear regression, decision tree, and random forest. The report uses various data analysis techniques and data visualisation methods to answer the research questions and demonstrate the skills and knowledge in using R programming for data analysis and visualisation. The report also discusses the findings, limitations, and recommendations based on the analysis results.

2.0 Introduction and Assumption

2.1 Introduction

The dataset we are focussing provides the rental of house in India. It includes several factors or attributes that might or might not affect the final rental price of the houses. These attributes consist of size, BHK (Bedroom, Hall, and Kitchen), tenant preferred, bathroom number, area type, etc. This project will provide a clear view of how the house rent price in India will be affected by BHK, house size, furnishing status, and other factors in each city mentioned in the dataset.

2.2 Assumption

As we mentioned in introduction part, this dataset consists of several variables that might affect rental price. Hence, we made our hypothesis that we assume BHK, house size and house furnishing status will affect the house rental more than the other variable. To prove our hypothesis accuracy, we will be introduced several analysis techniques such as descriptive analysis, dispersion analysis, regression analysis, etc.

3.0 Libraries Used

```
library(dplyr)
library(ggplot2)
library(GGally)
library(plotly)
library(corrplot)
library(ggthemes)
library(randomForest)
library(caret)
library(glmnet)
library(rpart)
library(rpart.plot)
```

Above shows the additional library and packages we use to various aspects of data analysis, visualisation, and machine learning in R, enhancing the capabilities of the language and making it easier to perform complex tasks.

1. **readxl**: A library that facilitates importing datasets from Excel files into R.
2. **dplyr**: Enables data manipulation tasks like filtering, summarizing, and transforming in R.
3. **ggplot2**: Allows the creation of sophisticated and customizable graphs and charts in R.
4. **GGally**: Extends **ggplot2** to provide functions for creating pairs plots and visualizing multivariate data.
5. **plotly**: Converts **ggplot2** plots into interactive charts that can be explored and analysed.
6. **corrplot**: Offers tools for creating informative correlation matrix heatmaps in R.
7. **randomForest**: Provides functions for building and analysing random forest models, an ensemble learning technique.
8. **caret**: Assists in training and evaluating machine learning models with a unified interface.
9. **glmnet**: Implements Lasso and Elastic-Net regularization for linear regression and classification models.
10. **rpart**: Supports building decision tree models using Recursive Partitioning and Regression Trees.
11. **rpart. plot**: Generates visually appealing plots of decision trees built with the **rpart** package.

4.0 Data Import

```
#Import dataset
HouseRent = read.csv("C:\\Users\\YC PUAH\\OneDrive - Asia Pacific University\\L2S1\\PFDA\\Assignment\\House_Rent_Dataset.csv", header = TRUE)
```

Above code shows the process of importing the dataset. The function used is called “read.csv”, which will read a .csv file in table format and creates a data frame from it. This data frame was stored in variable called HouseRent. The argument passed in the function is Boolean variable called header. This Boolean argument determine if the data frame read the first row of the .csv

file as variable label. In our case, we pass in TRUE value, means the data frame will have the first row of value as the column name. If the value passed is false, system will read the first row of data as normal value. However, the default passing value is TRUE.

5.0 Data Cleaning

Data cleaning means to clean out outlier, empty data, which might affect the result we get, hence affect our prediction accuracy negatively. According to Chu X. et al. (2016), identifying and repairing dirty data is the eternal challenge in data analytics, which failure to do so will cause unreliable prediction. We separate our cleaning into three parts, null/empty value, duplicate value, and outlier.

5.1 Null value

```
#Check null value
colSums(is.na(HouseRent))
```

```
> colSums(is.na(HouseRent))
       Posted.On              BHK             Rent             Size            Floor
               0                0                0                0                0
       Area.Type    Area.Locality             City Furnishing.Status  Tenant.Preferred
               0                0                0                0                0
        Bathroom Point.of.Contact
               0                0
```

For null value detection, we use combination of is.na() and colSums(), where is.na() function provides Boolean value of null value and colSums() provide the total number of TRUE values based on the columns. As a result, there is no null value in every column.

5.2 Duplicate value

```
#Check duplicate row
sum(duplicated(HouseRent))
```

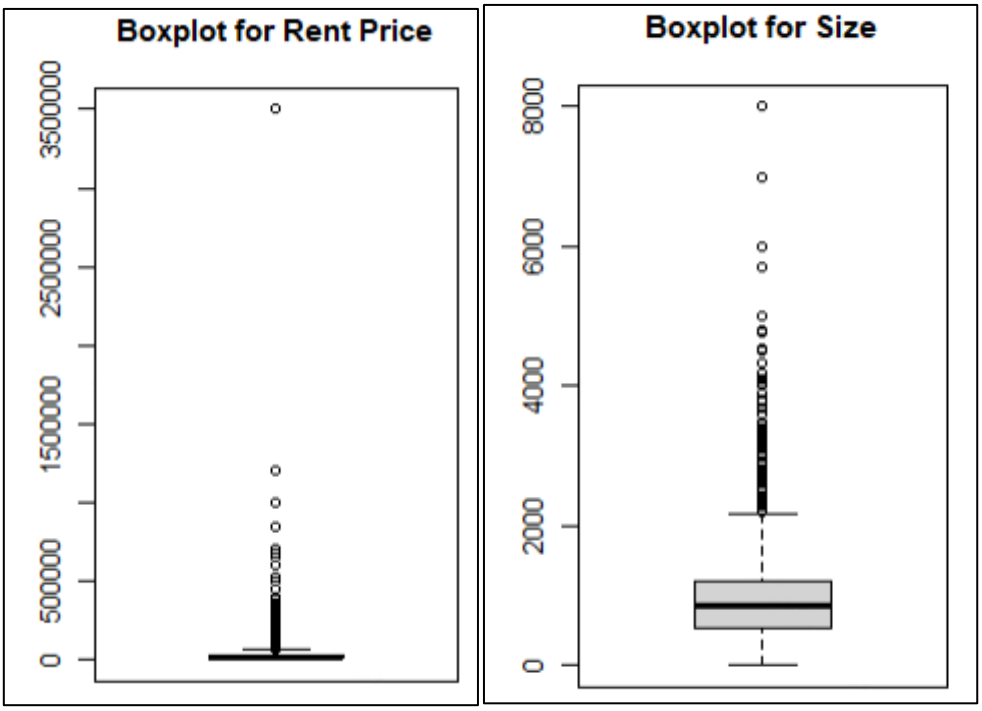
```
> sum(duplicated(HouseRent))
[1] 0
```

For duplicate value detection, we use duplicate() function to search duplicate row value, and use sum to sum up the TRUE result. As shown above, there is no duplicate value in the dataset.

5.3 Outlier

Based on our investigation for this dataset, we found out that only 'Rent' and 'Size' variable act as continuous value, other variables are categorical value. Hence, the outlier detection will only apply in these two variables.

```
#Check for outlier
boxplot(HouseRent$Rent)
title("Boxplot for Rent Price")
boxplot(HouseRent$Size)
title("Boxplot for Size")
```



Above figures, show the outliers for 'Rent' and 'Size' attribute using boxplot. Based on the boxplot, some outliers need to be removed for both variables. For accuracy purpose, our group decided to clean the 'Rent' and 'Size' variable based on city. This is to prevent any unnecessary data to be deleted based on the different average rent price among cities.

```
unique_cities = unique(HouseRent$City)

city_subsets <- setNames(replicate(length(unique_cities), NULL), unique_cities)

for(i in unique_cities){
  city_subset = HouseRent %>% filter(City == i)
  city_subsets[[i]] = city_subset
}

city_subsets_unclean = city_subsets
```

Above code aimed to disperse the dataset based on city and stored in a list called city_subsets. Unique_cities store the different cities, and city_subsets_unclean created for comparison.

5.3.1 Rent Cleaning in Kolkata City

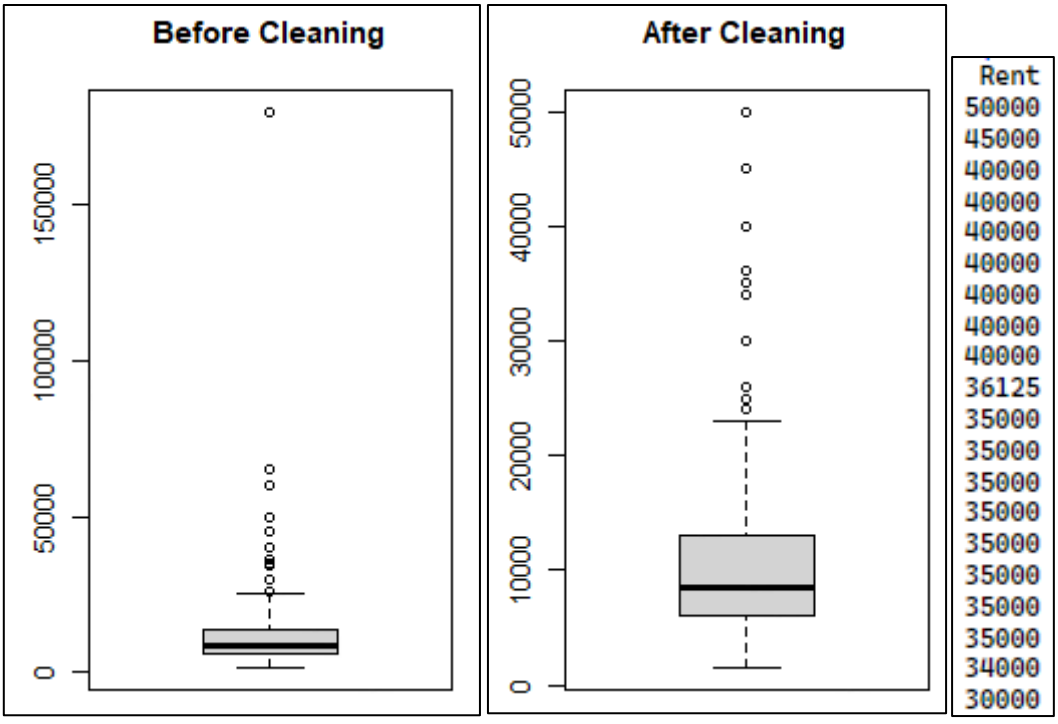
```
#Cleaning part for rent
{#Cleaning for Kolkata Rent
max(city_subsets$Kolkata$Rent)
city_subsets$Kolkata = city_subsets$Kolkata[city_subsets$Kolkata$Rent !=180000, ]
city_subsets$Kolkata = city_subsets$Kolkata[city_subsets$Kolkata$Rent !=65000, ]
city_subsets$Kolkata = city_subsets$Kolkata[city_subsets$Kolkata$Rent !=60000, ]

#Comparison before and after cleaning
boxplot(city_subsets_unclean$Kolkata$Rent)
title("Before Cleaning")
boxplot(city_subsets$Kolkata$Rent)
title("After Cleaning")

head(city_subsets$Kolkata[order(city_subsets$Kolkata$Rent, decreasing = TRUE), ], 20)
```

Here is the code for removing outliers for ‘Rent’ attribute in Kolkata city.

The actual procedure is to look at boxplot and personally identify the distance between plots. If the distance is not continuous, remove it using ‘!=’ logic.

The box plot holds a purpose to do a comparison before and after cleaning.

The last line of code provides the first 20 row of data with a descending ‘Rent’ price, to show to actual numeric result of the ‘Rent’ price.



By repeating the same code for each city, the outlier will be successfully removed to increase the accuracy of our result. Is important to take notes that the outlier we mention is the data that is not visually continuous. Please refer to Appendix 3.0 for other cities’ coding and comparison part, for both Rent and Size cleaning.

5.3.2 Comparison of Rent Before and After Cleaning in Each City

This section is to provide a better view of 'Rent' attribute in each city before and after cleaning process.

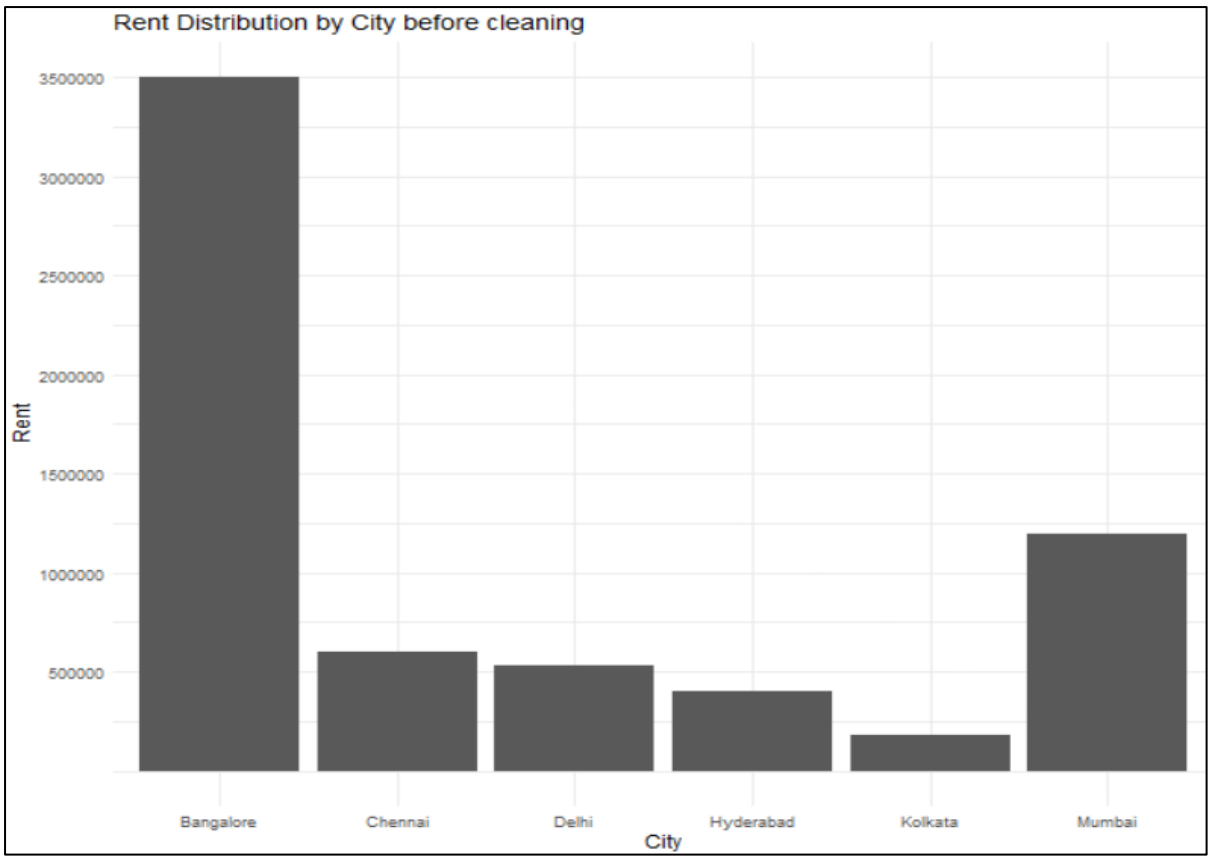
```
#Comparison before cleaning and after cleaning
custom_breaks <- seq(500000, max(HouseRent$Rent), by = 500000)

ggplot(HouseRent, aes(x = City, y = Rent)) +
  geom_bar(stat = "identity", position = "dodge") +
  ggtitle("Rent Distribution by City before cleaning") +
  labs(x = "City", y = "Rent") +
  scale_y_continuous(breaks = custom_breaks) +
  theme_minimal()
```

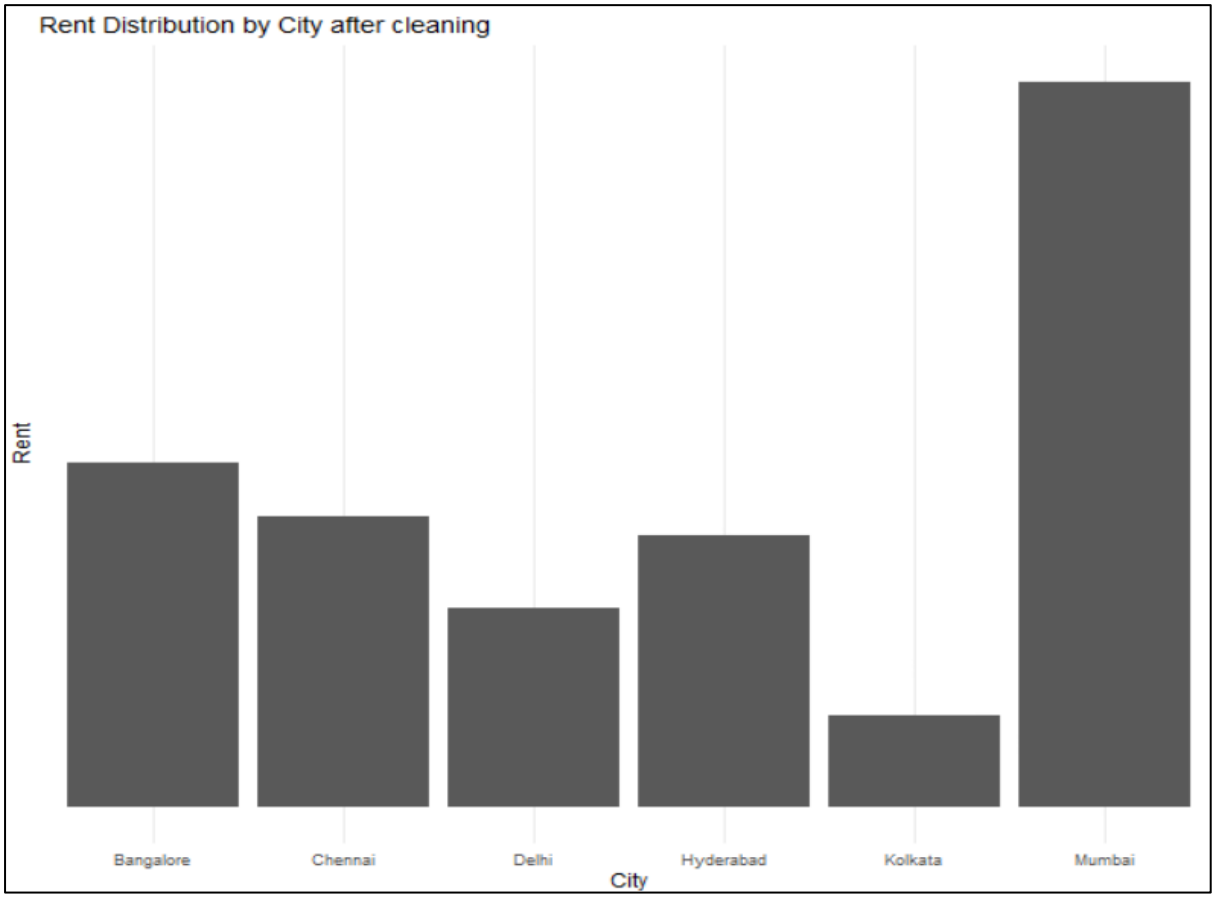
```
HouseRent_Cleaned = do.call(rbind, city_subsets)

ggplot(HouseRent_Cleaned, aes(x = City, y = Rent)) +
  geom_bar(stat = "identity", position = "dodge") +
  ggtitle("Rent Distribution by City after cleaning") +
  labs(x = "City", y = "Rent") +
  scale_y_continuous(breaks = (seq(100000, max(HouseRent_Cleaned$Rent), by = 100000))) +
  theme_minimal()
```

Images above show the code of creating the bar chart to achieve the comparison in 'Rent' attribute.

The function seq() is to create a list of sequenced numeric value which starts at 50000 and ends at the max of "Rent' value. This function benefits in creating bar chart to have a sequence in y-axis.

Do.call function helps in binding a list of data back to data frame. As our procedure, we have dispersed the original dataset into a list; meanwhile the do.call function is to combine the list back to data frame, to create valuable visualisation.


Rent Distribution by City before cleaning
3500000
3000000
2500000
2000000
1500000
1000000
500000
Rent
Bangalore
Chennai
Delhi
Hyderabad
Kolkata
Mumbai
City

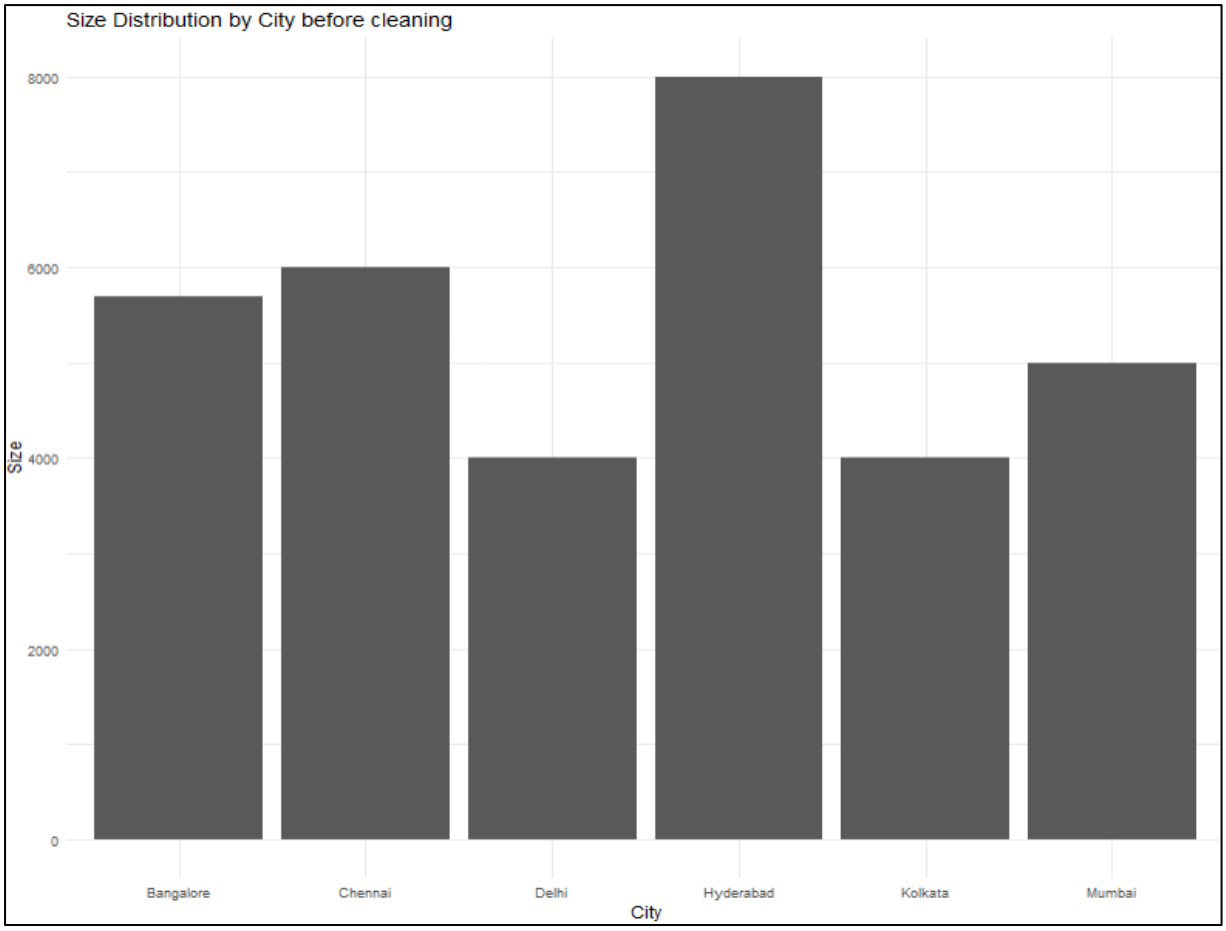

Rent Distribution by City after cleaning
Rent
Bangalore
Chennai
Delhi
Hyderabad
Kolkata
Mumbai
City

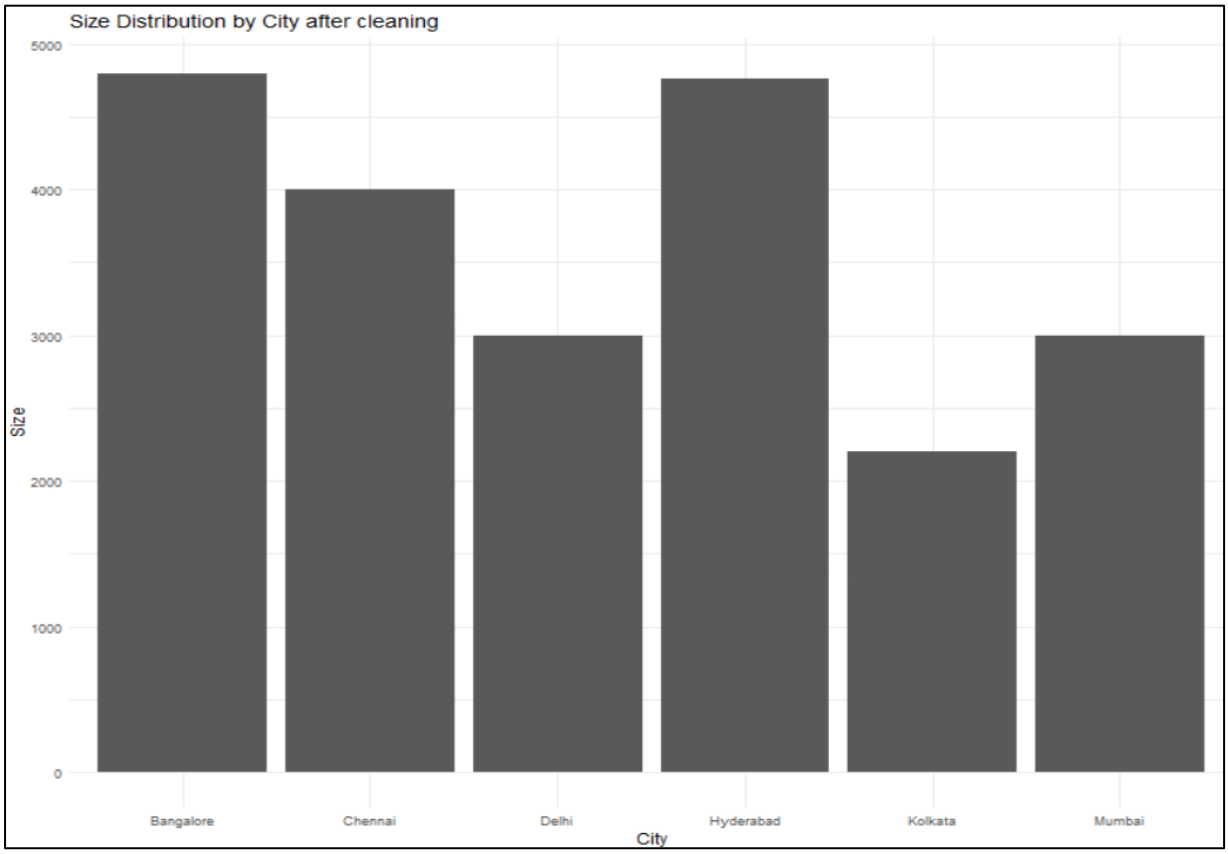
5.3.3 Comparison in Size Before and After Cleaning in Each City

```
ggplot(HouseRent, aes(x = City, y = Size)) +
  geom_bar(stat = "identity", position = "dodge") +
  ggtitle("Size Distribution by City before cleaning") +
  labs(x = "City", y = "Size") +
  theme_minimal()
```

```
ggplot(HouseRent_Cleaned, aes(x = City, y = Size)) +
  geom_bar(stat = "identity", position = "dodge") +
  ggtitle("Size Distribution by City after cleaning") +
  labs(x = "City", y = "Size") +
  theme_minimal()
```

Code above use the same procedure as ‘Rent’ to create a bar chart of ‘Size’ for each city.


Size Distribution by City before cleaning
8000
6000
4000
2000
0
Size
Bangalore
Chennai
Delhi
Hyderabad
Kolkata
Mumbai
City


Size Distribution by City after cleaning
5000
4000
3000
2000
1000
0
Size
Bangalore
Chennai
Delhi
Hyderabad
Kolkata
Mumbai
City

6.0 Data Pre-processing

According to Suad A. A. & Wesam S. B. (2017), data preprocessing is one of the vital processes, which consist of preparation and transformation, which make data more valuable in data mining. In our case, we encode some categorical attributes, to do modelling, testing, and correlation identifying. Furthermore, we also included data reduction to exclude the column that we assume have less relation with the rental price.

6.1 Category Encoding

```
HouseRent_Cleaned_Preprocessed = HouseRent_Cleaned

HouseRent_Cleaned_Preprocessed <- HouseRent_Cleaned_Preprocessed %>%
  mutate(Area.Type = case_when(
    Area.Type == "Super Area" ~ 0,
    Area.Type == "Carpet Area" ~ 1,
    Area.Type == "Built Area" ~ 2
  ),
  Furnishing.Status = case_when(
    Furnishing.Status == "Unfurnished" ~ 0,
    Furnishing.Status == "Semi-Furnished" ~ 1,
    Furnishing.Status == "Furnished" ~ 2
  ),
  Point.of.Contact = case_when(
    Point.of.Contact == "Contact Owner" ~ 0,
    Point.of.Contact == "Contact Agent" ~ 1,
    Point.of.Contact == "Contact Builder" ~ 2
  ),
  Tenant.Preferred = case_when(
    Tenant.Preferred == "Family" ~ 0,
    Tenant.Preferred == "Bachelors" ~ 1,
    Tenant.Preferred == "Bachelors/Family" ~ 2
  ))
```

Above shows the code of transforming categorical data from character to numeric data. The aim of this transforming is to achieve and enhance the quality of modelling and testing. In the code, we use mutate() function, which will modify the value of specify column.

6.2 Data Reduction

```
HouseRent_Cleaned_Preprocessed <- select(HouseRent_Cleaned_Preprocessed, -Posted.On, -Floor, -Area.Locality)
```

Above shows the code of column reduction. In this case, we deducted the columns ‘Posted On’, ‘Floor’ and ‘Area Locality’, which we assume have less impact with the rental price.

7.0 Data Exploration

Data Exploration aimed to provide some basic information of the dataset.

```
# Show all Column name
names(HouseRent)
```

```
> names(HouseRent)
 [1] "Posted.On"         "BHK"               "Rent"              "Size"
 [5] "Floor"             "Area.Type"         "Area.Locality"     "City"
 [9] "Furnishing.Status" "Tenant.Preferred"  "Bathroom"          "Point.of.Contact"
```

Above code shows all the column name in the dataset.

```
#Total column
ncol(HouseRent)
```

```
> ncol(HouseRent)
[1] 12
```

Above shows the total number of columns in this dataset, which is 12 columns.

```
#Total row
nrow(HouseRent)
```

```
> nrow(HouseRent)
[1] 4746
```

Here shows the total row of value in the dataset, which have 4746 rows of value.

```
#Structure of the dataset
str(HouseRent)
```

```
> str(HouseRent)
'data.frame':   4746 obs. of  12 variables:
 $ Posted.On        : chr  "5/18/2022" "5/13/2022" "5/16/2022" "7/4/2022" ...
 $ BHK              : int  2 2 2 2 2 2 1 2 2 ...
 $ Rent             : int  10000 20000 17000 10000 7500 7000 10000 5000 26000 10000 ...
 $ Size             : int  1100 800 1000 800 850 600 700 250 800 1000 ...
 $ Floor            : chr  "Ground out of 2" "1 out of 3" "1 out of 3" "1 out of 2" ...
 $ Area.Type        : chr  "Super Area" "Super Area" "Super Area" "Super Area" ...
 $ Area.Locality    : chr  "Bandel" "Phool Bagan, Kankurgachi" "Salt Lake City Sector 2" "Dumdum Park"
...
 $ City             : chr  "Kolkata" "Kolkata" "Kolkata" "Kolkata" ...
 $ Furnishing.Status: chr  "Unfurnished" "Semi-Furnished" "Semi-Furnished" "Unfurnished" ...
 $ Tenant.Preferred : chr  "Bachelors/Family" "Bachelors/Family" "Bachelors/Family" "Bachelors/Family"
...
 $ Bathroom         : int  2 1 1 1 1 2 2 1 2 2 ...
 $ Point.of.Contact : chr  "Contact Owner" "Contact Owner" "Contact Owner" "Contact Owner" ...
```

The function str() provides us the internal structure of the dataset, which included all the attributes data types and some sample value in the dataset.

7.1 Data exploration for Categorical Variable

In this section, we will use frequency to show the distribution of categorical attribute.

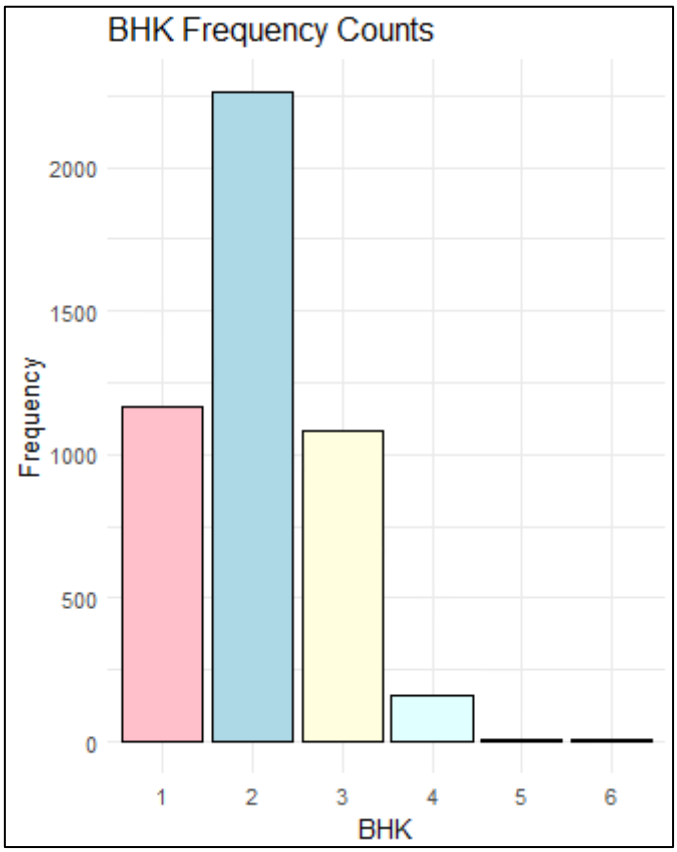
7.1.1 Data Exploration for BHK

```
BHK_Count = as.data.frame(table(HouseRent_Cleaned$BHK))
names(BHK_Count)[1] = "BHK"
print(BHK_Count, row.names = FALSE)
```

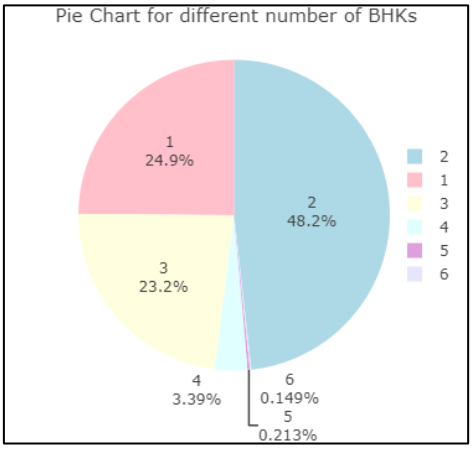
```
 BHK Freq
   1 1166
   2 2262
   3 1086
   4  159
   5   10
   6    7
```

Based on the result from the code shown above, we can understand in this dataset, 2 BHK hold most of the dataset, and 6 BHK only have 7 rows of value.

```
ggplot(BHK_Count, aes(x = BHK, y = Freq)) +
  geom_bar(stat = "identity", fill = c("pink", "lightblue","lightyellow", "lightcyan", "#DDA0DD", "lavender")
           , color = "black") +
  labs(title = "BHK Frequency Counts",
       x = "BHK",
       y = "Frequency") +
  theme_minimal()
```

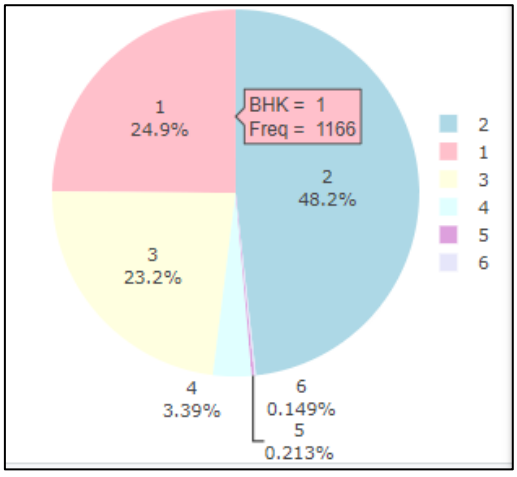


```
plot_ly(BHK_Count, labels =~BHK, values = ~Freq, type = "pie",
        marker = list(colors = c("pink", "lightblue","lightyellow", "lightcyan", "#DDA0DD", "lavender")),
        textinfo = "label+percent",
        hoverinfo = 'text',
        text = ~paste('BHK = ',BHK, '\nFreq = ', Freq))%>%
  layout(showlegend = TRUE,
         legend = list(orientation = 'v', x = 1, y = 0.5),
         title = list(text = "Pie Chart for different number of BHKs", font = list(size = 15)))
```



The first bar chart provides a better view for the distribution of BHK number, by using ggplot2 library.

In addition, the second pie chart also shows the percentage hold by each value of BHK. Do take note that we use plotly library for the pie chart. Hence, user is able to interact with the pie chart in the R Studio as shown below.



Do take note that for every other categorical variable, we will show the result of data exploration in appendix 5.0.

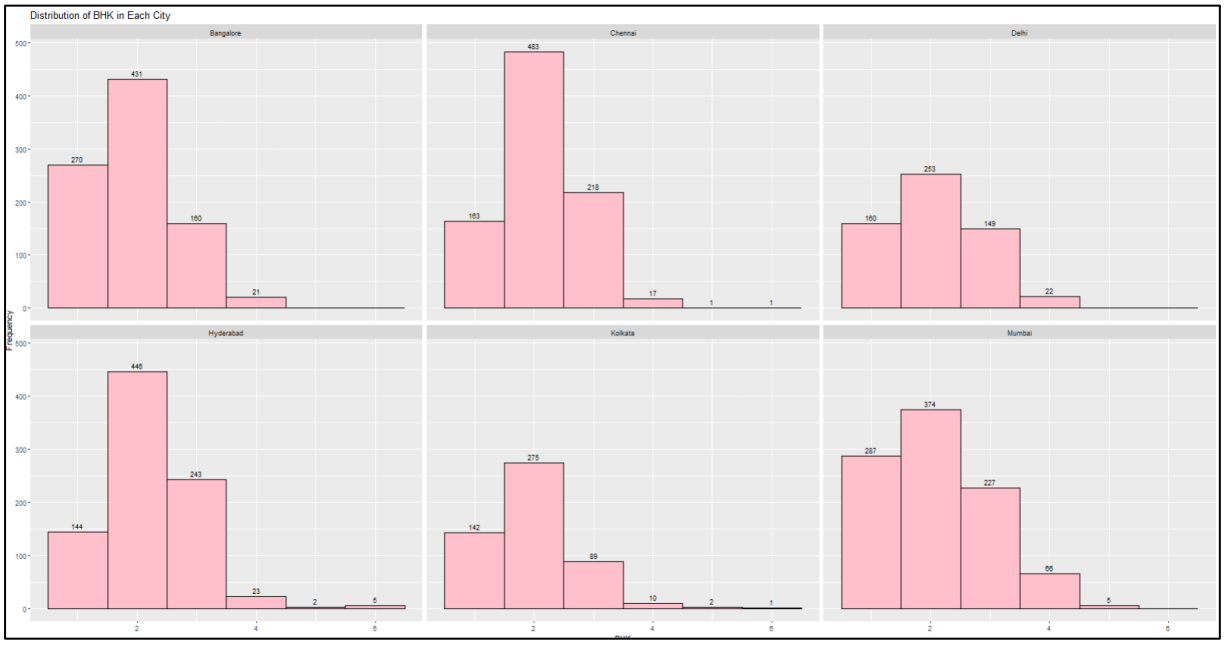
8.0 Objectives, Question and Analysis

8.1 Objective 1: Examine the impact from BHK to Rent price in Each City (Puah Yi Kai TP069207)

8.1.1 Question 1: How is the distribution of Rent and BHK in each city?

BHK

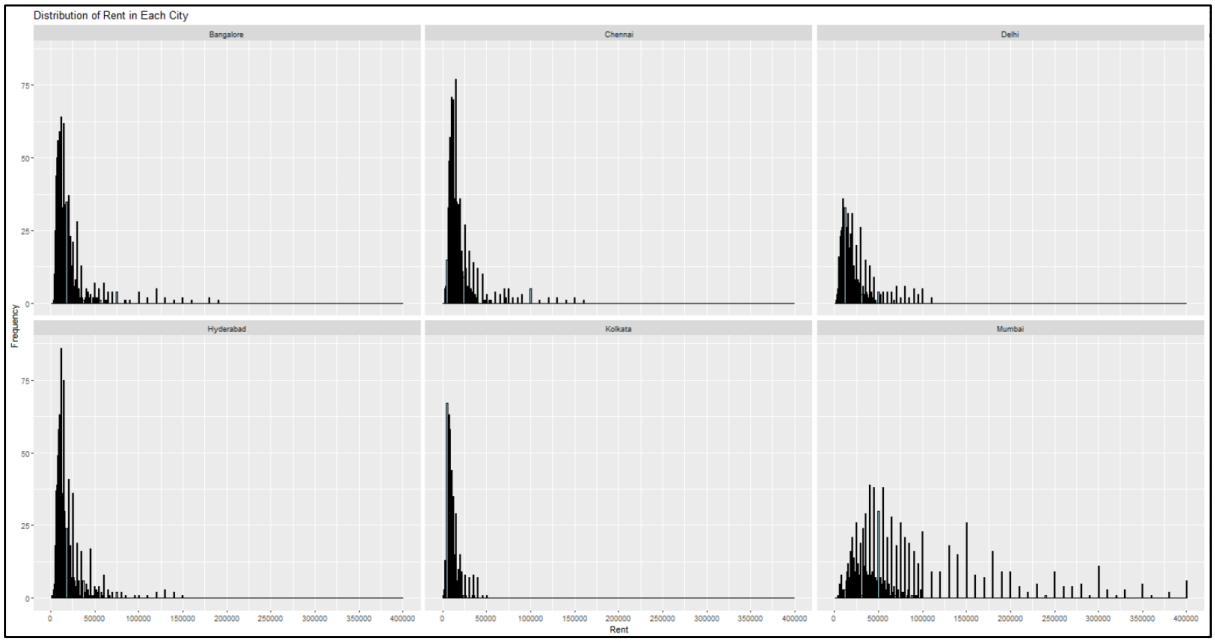
```
#Question 1 How is the distribution of Rent and BHK in each city?
ggplot(data = HouseRent_Cleaned, aes(x = BHK)) +
  geom_histogram(binwidth = 1, color = "black", fill= "pink") +
  facet_wrap(~City) +
  labs(title = "Distribution of BHK in Each City",
       x = "BHK",
       y = "Frequency")+
  geom_text(aes(label = ..count..),
            stat = "count",
            vjust = -0.5,  # Adjust the vertical position of labels
            color = "black",
            size = 3)
```



According to the distribution of 'BHK' in each city, we can understand that in all cities, houses that have 2 BHK always come first. In addition, houses that own 5 and 6 BHK always be the lowest among all the cities.

Rent

```
ggplot(HouseRent_Cleaned, aes(x = Rent)) +
  geom_histogram(binwidth = 1000, color = "black", fill = "lightblue") +
  labs(title = "Distribution of Rent in Each City",
       x = "Rent",
       y = "Frequency")+
  facet_wrap(~City)+
  scale_x_continuous(breaks = seq(0, max(HouseRent_Cleaned$Rent), by = 50000))
```

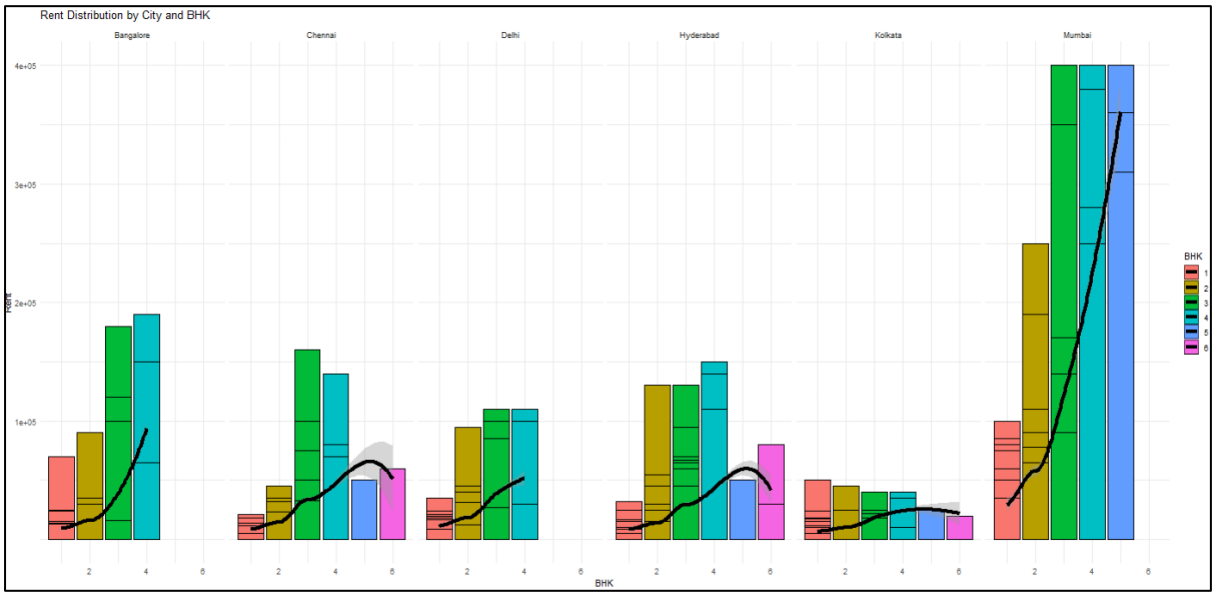


Based on the histogram of Rent in each city, except for Mumbai City, other cities Rent price often falls below than 50000. Only for Mumbai City, the distribution of Rent price is more symmetrical distributed, resulting at having frequencies in high Rent price.

8.1.2 Question 2: How does the Rent price changes with different BHK in each city?

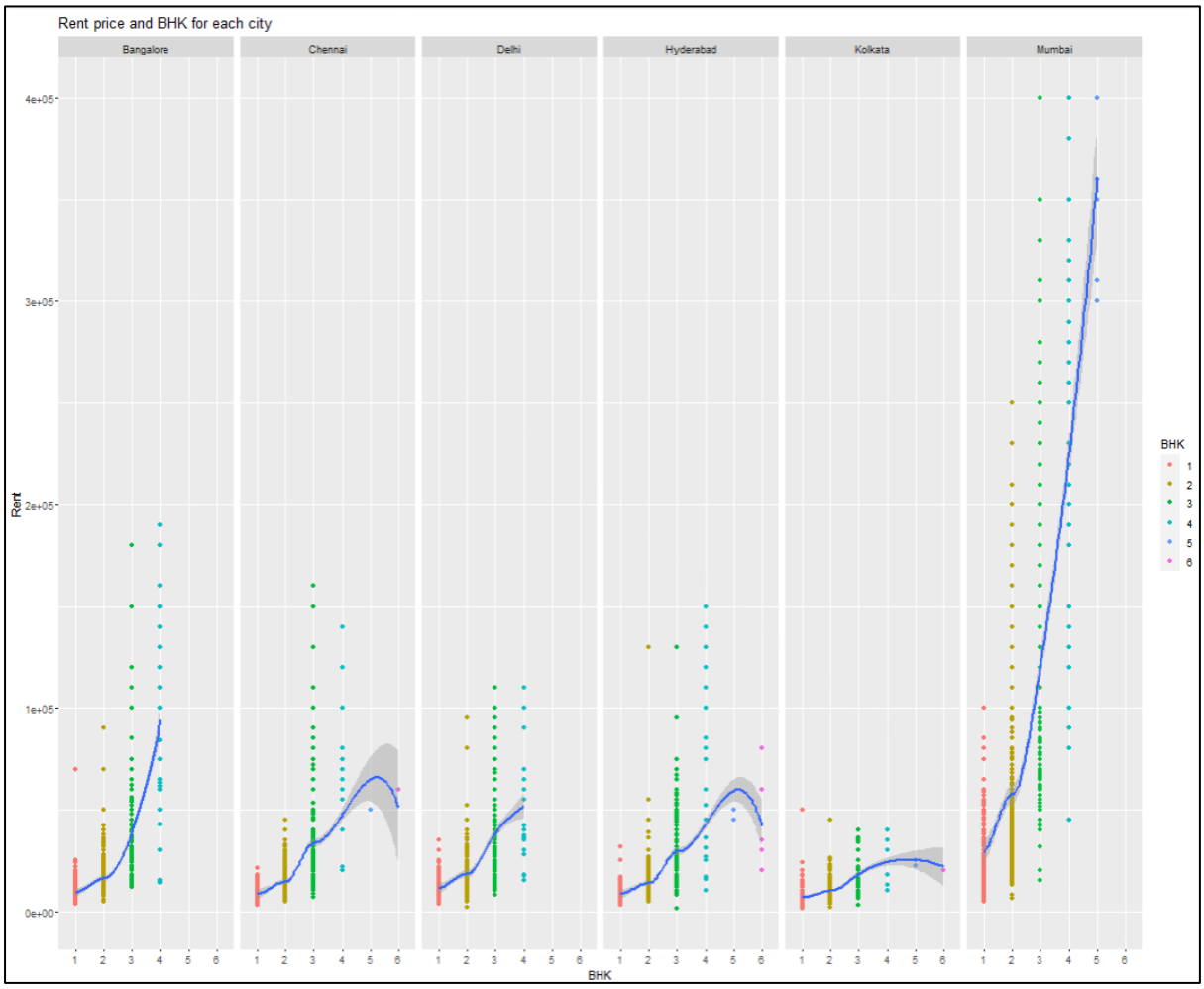
Bar Chart

```
ggplot(HouseRent_Cleaned, aes(x = BHK, y = Rent, fill = factor(BHK))) +
  geom_bar(stat = "identity", position = "dodge", color="Black") +
  ggtitle("Rent Distribution by City and BHK") +
  labs(x = "BHK", y = "Rent") +
  scale_fill_discrete(name = "BHK") +
  scale_y_continuous(breaks = (seq(100000, max(HouseRent_Cleaned$Rent), by = 100000))) +
  geom_smooth(aes(group = City), method = loess, color = 'black', linewidth = 1.5)+
  facet_grid(~City)+
  theme_minimal()
```



Scatter Chart

```
ggplot(HouseRent_Cleaned_Preprocessed, aes(x = factor(BHK), y = Rent)) +
  geom_point(aes(color = factor(BHK)))+
  labs(x = "BHK", y = "Rent") +
  scale_color_discrete(name = "BHK") +
  ggtitle("Rent price and BHK for each city")+
  geom_smooth(aes(group = City),method = loess)+
  facet_grid(~City)
```



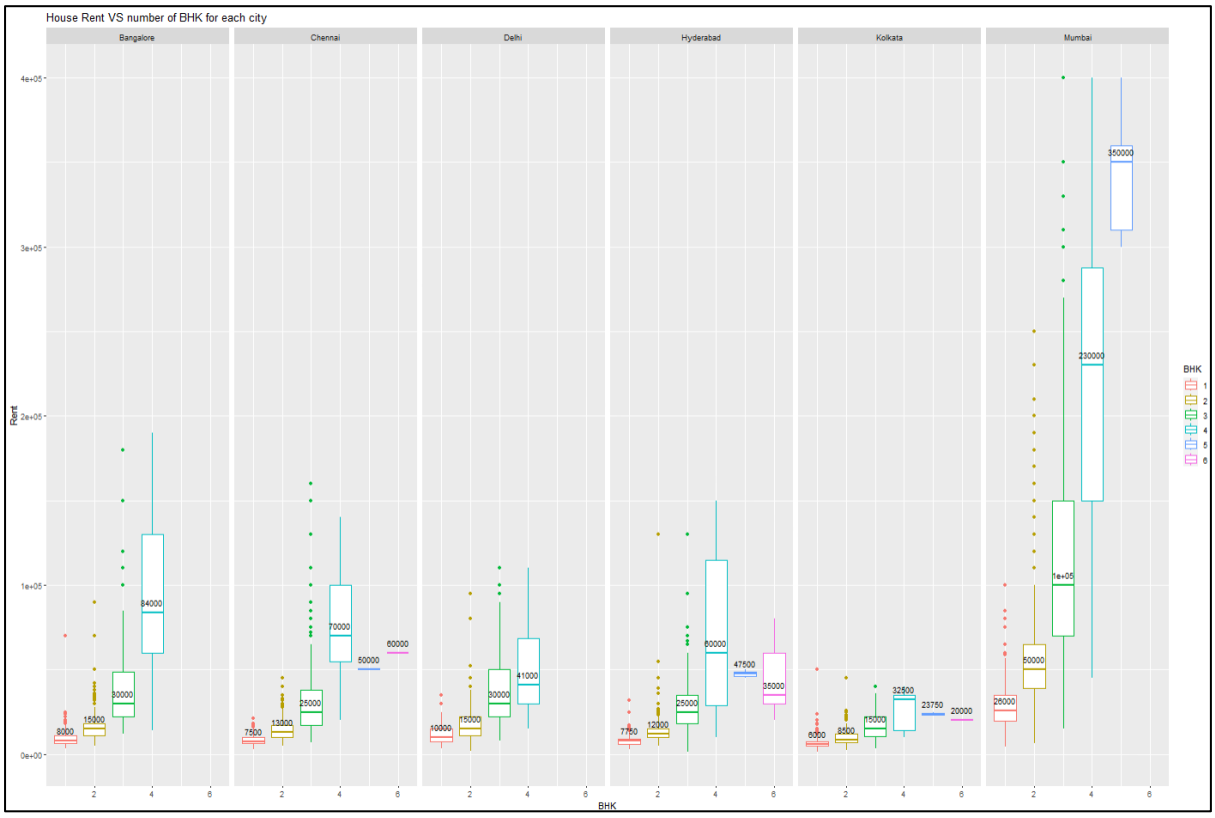
In above bar chart and scatter plot, we show how the Rent price will differ with the changes of BHK for each city. As the result, in Bangalore, Delhi and Mumbai City, Rent and BHK have positive relation, means the Rent price increase when the BHK increases. For Chennai and Hyderabad City, rent price follows the BHK increase until reach the 5 BHK. However, in Kolkata City, the increase rate of Rent price is low, hence the line will visualise flatter.

Furthermore, we can easily see that Mumbai city have the highest average rent compared to the other cities. On the other hand, Kolkata have the lowest average rent.

8.1.3 Question 3: Are they any significant difference in Rent distribution based on different BHK?

Box Plot

```
#Question 3:Are they any significant difference in Rent distribution based on different BHK?
ggplot(HouseRent_Cleaned_Preprocessed, aes(x = BHK, y = Rent)) +
  geom_boxplot(aes(color = factor(BHK)))+
  stat_summary(
    fun = median,
    geom = "text",
    aes(label = round(..y.., 2)),
    position = position_dodge(width = 0.75),
    vjust = -1,  # Adjust vertical position of text
    color = "black",
    size = 3
  )+
  ggtitle("House Rent VS number of BHK for each city")+
  facet_grid(~City)+
  labs(x = "BHK", y = "Rent") +
  scale_color_discrete(name = "BHK")
```



Above graph shows the box plot and the median value for each box plot. As a result, in Bangalore, Mumbai, Chennai and Hyderabad cities, there is a significance difference of Rent based on the changes of BHK. In Bangalore, 1 BHK houses have a median of 8000 where 4 BHK houses have a median of 84000. Same as Mumbai city, 1 BHK houses have 26000 median and 5 BHK houses have a median of 350000.

However, in Delhi and Kolkata city, the difference of Rent based on BHK is not as apparent as other cities. For example, in Kolkata city, 1 BHK houses have 6000 as median and 4 BHK houses have 32500 as median.

8.1.4 Question 4: Is there a correlation between BHK and Rent?

```
#Question 4: Is there a correlation between BHK and Rent?
HouseRent_Cleaned_Preprocessed %>%
  group_by(City) %>%
  summarize(correlation = cor(BHK, Rent))
```

```
> HouseRent_Cleaned_Preprocessed %>%
+   group_by(City) %>%
+   summarize(correlation = cor(BHK, Rent))
# A tibble: 6 × 2
  City      correlation
  <chr>           <dbl>
1 Bangalore       0.620
2 Chennai         0.583
3 Delhi           0.588
4 Hyderabad       0.593
5 Kolkata         0.554
6 Mumbai          0.733
```

Figures above show the relation of BHK and Rent in each city based on the correlation computation. As the result shows, all the cities have a positive correlation coefficient over than 0.5. This result suggests that when the BHK increases, Rent will also increase.

However, with only this result does not provide the impact of BHK is significant compared to other attributes. Hence, we did a heat map of all attributes for each city in Section 9. Do take note that for more detailed and compared correlation analysis will be shown in Appendix section.

8.1.5 Question 5: Can BHK predict Rent price in each city?

```
#Can BHK predict Rent price in each city?
HouseRent_Cleaned_Preprocessed %>%
  group_by(City) %>%
  do(anova(lm(Rent ~ BHK, data = .)))
```

```
> #Can BHK predict Rent price in each city?
> HouseRent_Cleaned_Preprocessed %>%
+   group_by(City) %>%
+   do(anova(lm(Rent ~ BHK, data = .)))
# A tibble: 12 × 6
# Groups:   City [6]
   City         Df `Sum Sq` `Mean Sq` `F value`   `Pr(>F)`
   <chr>     <int>    <dbl>     <dbl>     <dbl>      <dbl>
 1 Bangalore     1  1.61e11   1.61e11      550.  6.43e- 95
 2 Bangalore   880  2.57e11   2.92e 8       NA  NA
 3 Chennai       1  1.06e11   1.06e11      454.  1.46e- 81
 4 Chennai     881  2.05e11   2.33e 8       NA  NA
 5 Delhi         1  7.62e10   7.62e10      307.  1.56e- 55
 6 Delhi       582  1.44e11   2.48e 8       NA  NA
 7 Hyderabad     1  9.31e10   9.31e10      466.  5.85e- 83
 8 Hyderabad   861  1.72e11   2.00e 8       NA  NA
 9 Kolkata       1  9.21e 9   9.21e 9      229.  4.46e- 43
10 Kolkata     517  2.08e10   4.02e 7       NA  NA
11 Mumbai        1  2.82e12   2.82e12     1112.  2.12e-162
12 Mumbai      957  2.43e12   2.54e 9       NA  NA
```

Above image shows the ANOVA testing for BHK and Rent in each city. Based on the results, in each city, the p-value of the main effect of BHK is very close to 0, meaning that BHK and Rent have a highly significant relationship. In summary, the ANOVA testing suggested that the impact of BHK towards Rent price is significantly high. However, same as the correlation, we also have a full ANOVA testing for each attribute towards Rent in Section 10, in order to further prove our hypothesis.

8.1.6 Result in Observation Proving

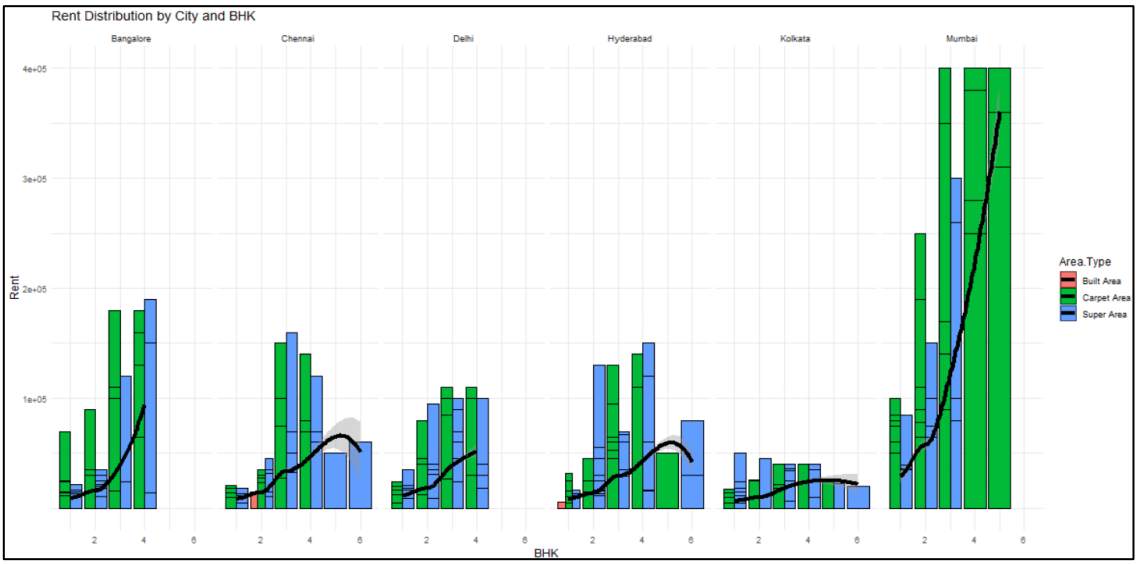
Above 5 questions and analysis investigated the impact of BHK to Rent price for each City. As our question 1,2 and 3 results, we proved that BHK did have a significance impact towards Rent Price. For question 4 and 5, we did Correlation and ANOVA testing which also provides the same result, which is BHK do have a correlation it is significance in affecting Rent price. Hence, my observation, which is BHK have a significance impact towards Rent, have proven true.

In order to further prove my observation, following question will introduce other attributes to examine if the other attributes have a higher relation with Rent.

8.1.7 Question 6: How is the relationship between BHK and Rent with a different Area Type in each city.

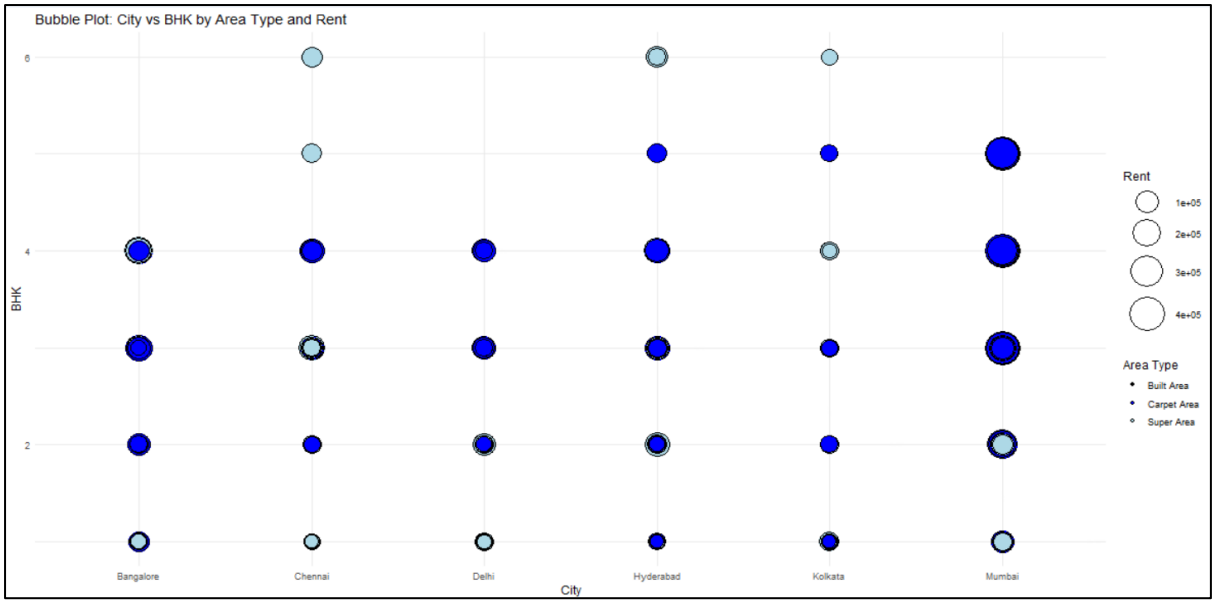
Bar Chart

```
ggplot(HouseRent_Cleaned, aes(x = BHK, y = Rent, fill = factor(Area.Type))) +
  geom_bar(stat = "identity", position = "dodge", color="Black") +
  ggtitle("Rent Distribution by City and BHK") +
  labs(x = "BHK", y = "Rent") +
  scale_fill_discrete(name = "Area.Type") +
  scale_y_continuous(breaks = (seq(100000, max(HouseRent_Cleaned$Rent), by = 100000))) +
  geom_smooth(aes(group = City), method = loess, color = 'black', linewidth = 1.5)+
  facet_grid(~City)+
  theme_minimal()
```



Bubble Plot

```
ggplot(HouseRent_Cleaned, aes(x = City, y = BHK, fill = factor(Area.Type))) +
  geom_point(aes(size = Rent), shape = 21, color = "black") +
  scale_fill_manual(values = c("Super Area" = "lightblue", "Carpet Area" = "blue", "Built Area" = "black")) +
  scale_size_continuous(range = c(5, 15)) +  # Adjust the range for bubble sizes
  labs(x = "City", y = "BHK", fill = "Area Type", size = "Rent") +
  ggtitle("Bubble Plot: City vs BHK by Area Type and Rent") +
  theme_minimal()
```



Correlation between Area Type and Rent in each city.

```
HouseRent_Cleaned_Preprocessed %>%
  group_by(City) %>%
  summarize(correlation = cor(Area.Type, Rent))
```

```
# A tibble: 6 × 2
  City       correlation
  <chr>            <dbl>
1 Bangalore        0.254
2 Chennai          0.181
3 Delhi            0.224
4 Hyderabad        0.226
5 Kolkata          0.0266
6 Mumbai           0.216
```

ANOVA Testing

```
HouseRent_Cleaned_Preprocessed %>%
  group_by(City) %>%
  do(anova(lm(Rent ~ Area.Type, data = .)))
```

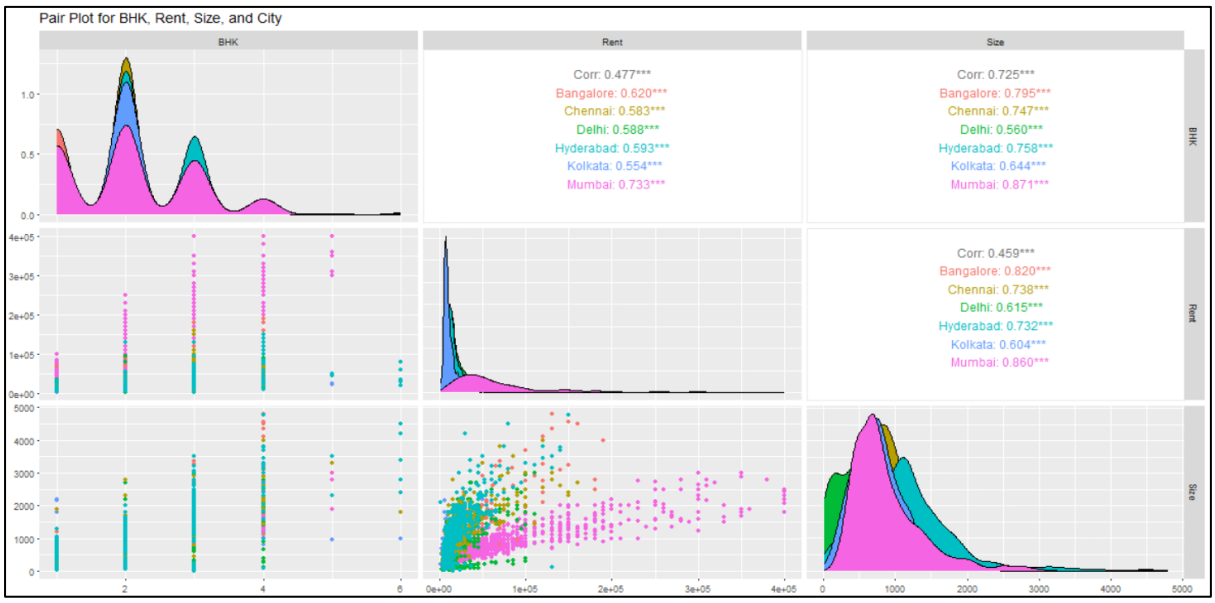
```
> HouseRent_Cleaned_Preprocessed %>%
+   group_by(City) %>%
+   do(anova(lm(Rent ~ Area.Type, data = .)))
# A tibble: 12 × 6
# Groups:   City [6]
   City          Df `Sum Sq`     `Mean Sq` `F value`  `Pr(>F)`
   <chr>      <int>    <dbl>         <dbl>     <dbl>     <dbl>
 1 Bangalore      1  2.70e10  26970329556.     60.8    1.78e-14
 2 Bangalore    880  3.90e11    443533472.     NA      NA
 3 Chennai        1  1.02e10  10161003782.     29.8    6.39e- 8
 4 Chennai      881  3.01e11    341541191.     NA      NA
 5 Delhi          1  1.11e10  11091411745.     30.8    4.29e- 8
 6 Delhi        582  2.09e11    359770800.     NA      NA
 7 Hyderabad      1  1.35e10  13537434836.     46.4    1.85e-11
 8 Hyderabad    861  2.51e11    292065705.     NA      NA
 9 Kolkata        1  2.13e 7     21253417.      0.366  5.45e- 1
10 Kolkata      517  3.00e10     58009927.     NA      NA
11 Mumbai         1  2.46e11 245704374694.     47.0    1.26e-11
12 Mumbai       957  5.00e12   5225130831.     NA      NA
```

Above charts shows the relationship between BHK and Rent in different Area Type. As a result, Area Type does not form a significance difference with Rent. We can observed the result as for example, Mumbai city have carpet area as higher rent and other city are more likely to have built area owns higher rent. In addition, the correlation results also provide a low correlation between Area Type and Rent. For ANOVA testing, although in each city the p-value is lower than 0.05, however, there is still a big difference compared to the p-value of BHK and Rent.

8.1.8 Question 7: How is the relationship between BHK and Rent in different Size across Cities

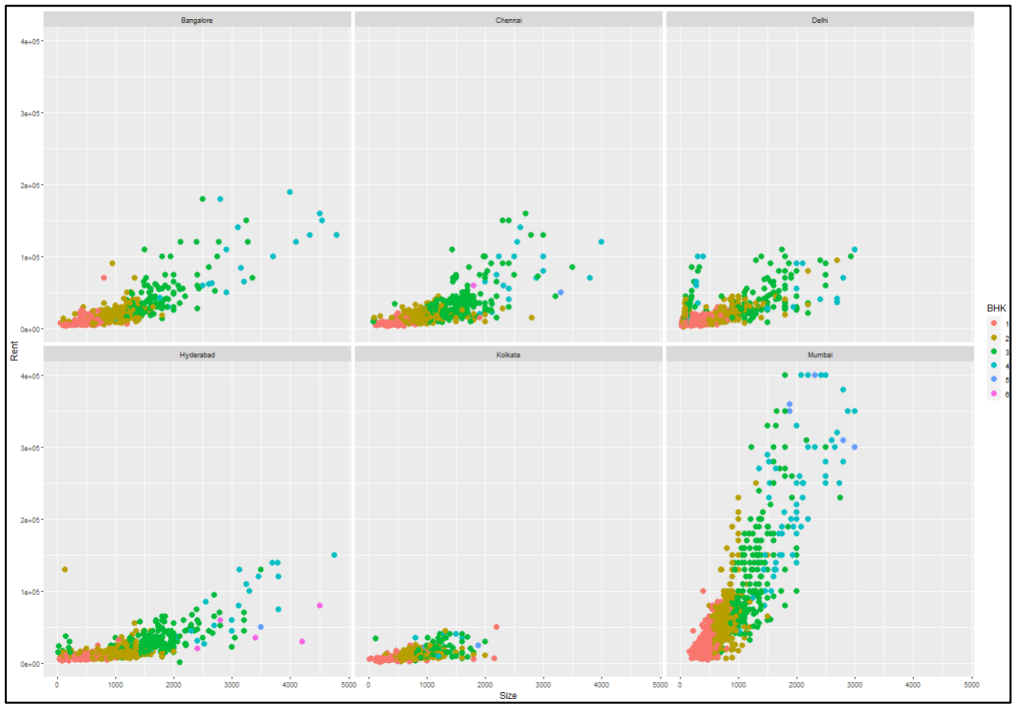
Pair Plot

```
ggpairs(data = HouseRent_Cleaned,
        columns = c("BHK", "Rent", "Size"),
        aes(color = City),
        title = "Pair Plot for BHK, Rent, Size, and City")
```



Scatter Plot

```
ggplot(HouseRent_Cleaned_Preprocessed, aes(x = Size, y = Rent, z = factor(BHK), color = factor(BHK))) +
  geom_point(size = 3) +
  scale_color_discrete(name = "BHK") +
  labs(x = "Size", y = "Rent", z = "BHK") +
  facet_wrap(~City)+
```



Correlation

```
HouseRent_Cleaned_Preprocessed %>%
  group_by(City) %>%
  summarize(correlation = cor(Size, Rent))
```

```
> HouseRent_Cleaned_Preprocessed %>%
+   group_by(City) %>%
+   summarize(correlation = cor(Size, Rent))
# A tibble: 6 × 2
  City      correlation
  <chr>           <dbl>
1 Bangalore       0.820
2 Chennai         0.738
3 Delhi           0.615
4 Hyderabad       0.732
5 Kolkata         0.604
6 Mumbai          0.860
```

ANOVA testing

```
HouseRent_Cleaned_Preprocessed %>%
  group_by(City) %>%
  do(anova(lm(Rent ~ Size, data = .)))
```

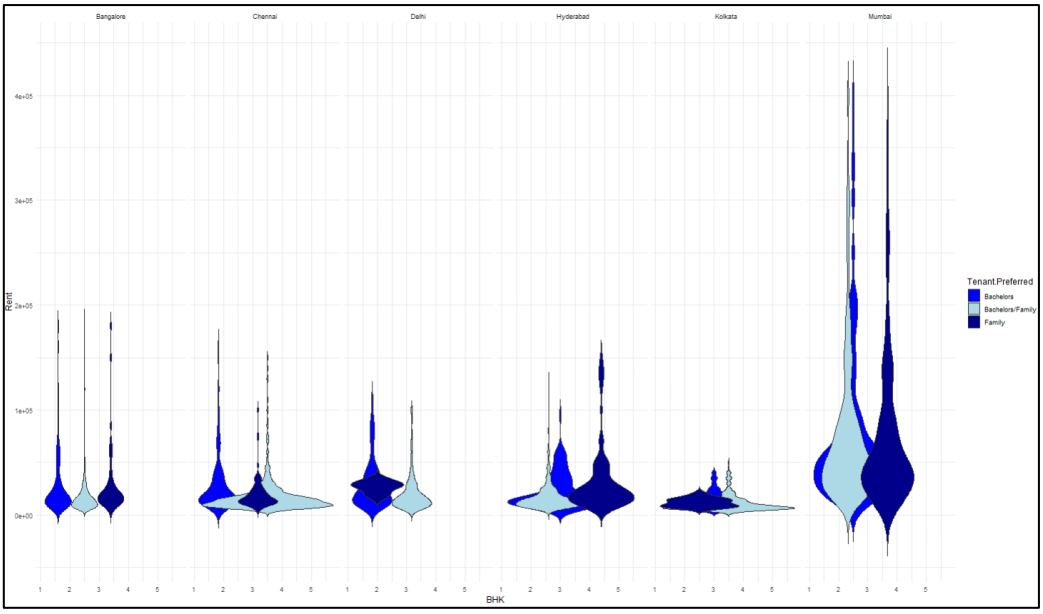
```
> HouseRent_Cleaned_Preprocessed %>%
+   group_by(City) %>%
+   do(anova(lm(Rent ~ Size, data = .)))
# A tibble: 12 × 6
# Groups:   City [6]
   City          Df 'Sum Sq' 'Mean Sq' 'F value'   'Pr(>F)'
   <chr>      <int>    <dbl>     <dbl>     <dbl>      <dbl>
 1 Bangalore      1  2.81e11   2.81e11     1809.  1.16e-215
 2 Bangalore    880  1.37e11   1.55e 8       NA  NA
 3 Chennai        1  1.69e11   1.69e11     1054.  1.10e-152
 4 Chennai      881  1.42e11   1.61e 8       NA  NA
 5 Delhi          1  8.35e10   8.35e10      355.  3.61e- 62
 6 Delhi        582  1.37e11   2.35e 8       NA  NA
 7 Hyderabad      1  1.42e11   1.42e11      995.  8.57e-146
 8 Hyderabad    861  1.23e11   1.43e 8       NA  NA
 9 Kolkata        1  1.09e10   1.09e10      297.  6.62e- 53
10 Kolkata      517  1.91e10   3.69e 7       NA  NA
11 Mumbai         1  3.88e12   3.88e12     2719.  6.72e-282
12 Mumbai       957  1.37e12   1.43e 9       NA  NA
```

Based on the results shows above, Size attribute do have a higher impact towards Rent compared to BHK. From the Scatter plot, we can observed that there is a positive relationship within size and Rent price without the effect of BHK, which will also provide the prove that Size and Rent have a high correlation and extremely low p-value.

8.1.9 Question 8: How is the relationship between BHK and Rent in different Tenant Preferred across Cities

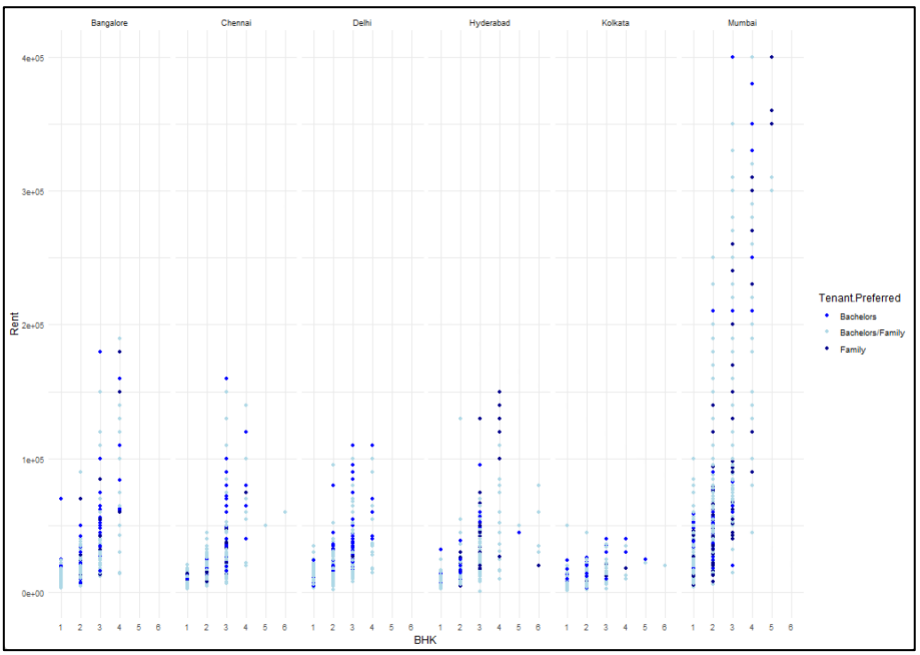
Violin Plot

```
#Question 8: How is the relationship between BHK and Rent in different Tenant Preferred across Cities
ggplot(HouseRent_Cleaned, aes(x = BHK, y = Rent, fill = Tenant.Preferred)) +
  geom_violin(scale = "width", trim = FALSE) +
  labs(x = "BHK", y = "Rent") +
  scale_fill_manual(values = c("Family" = "darkblue", "Bachelors" = "blue", "Bachelors/Family" = "lightblue")) +
  facet_grid(~ City) +
  theme_minimal()
```



Point Plot

```
ggplot(HouseRent_Cleaned, aes(x = factor(BHK), y = Rent, color = Tenant.Preferred)) +
  geom_point() +
  labs(x = "BHK", y = "Rent") +
  scale_color_manual(values = c("Family" = "darkblue", "Bachelors" = "blue", "Bachelors/Family" = "lightblue")) +
  facet_grid(~ City) +
  theme_minimal()
```



Correlation

```
HouseRent_Cleaned_Preprocessed %>%
  group_by(City) %>%
  summarize(correlation = cor(Tenant.Preferred, Rent))
```

```
> HouseRent_Cleaned_Preprocessed %>%
+   group_by(City) %>%
+   summarize(correlation = cor(Tenant.Preferred, Rent))
# A tibble: 6 × 2
  City      correlation
  <chr>           <dbl>
1 Bangalore    -0.174
2 Chennai      -0.119
3 Delhi        -0.191
4 Hyderabad    -0.277
5 Kolkata      -0.0653
6 Mumbai       -0.000820
```

ANOVA testing

```
HouseRent_Cleaned_Preprocessed %>%
  group_by(City) %>%
  do(anova(lm(Rent ~ Tenant.Preferred, data = .)))
```

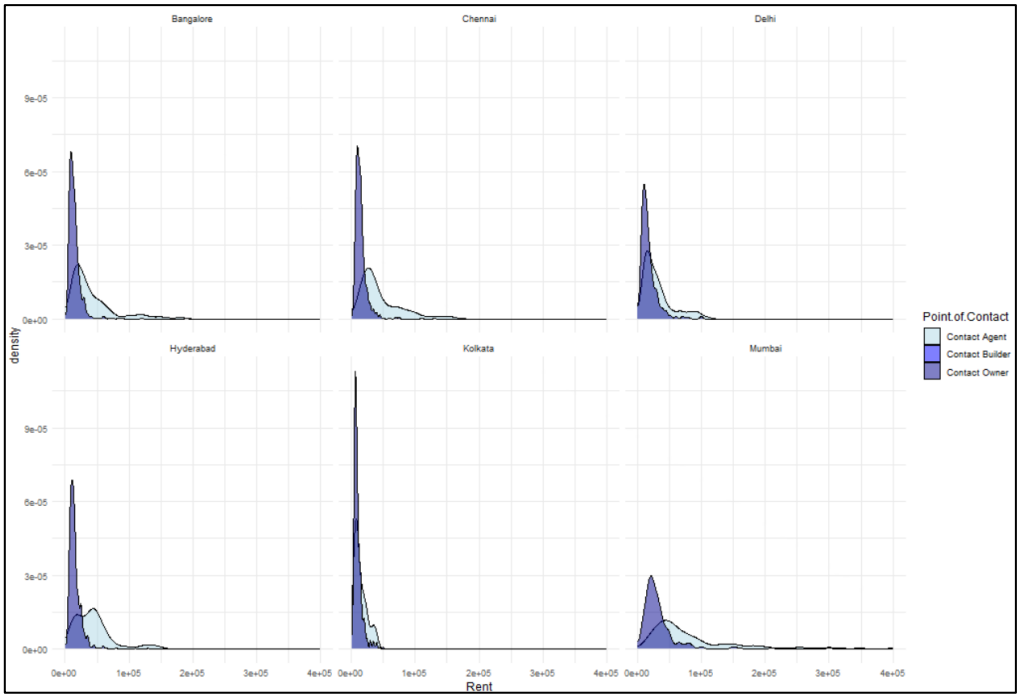
```
> HouseRent_Cleaned_Preprocessed %>%
+   group_by(City) %>%
+   do(anova(lm(Rent ~ Tenant.Preferred, data = .)))
# A tibble: 12 × 6
# Groups:   City [6]
   City         Df `Sum Sq`    `Mean Sq` `F value`  `Pr(>F)`
   <chr>     <int>    <dbl>        <dbl>     <dbl>     <dbl>
 1 Bangalore     1  1.27e10 12651688182. 27.5       1.95e- 7
 2 Bangalore   880  4.05e11   459804655. NA         NA
 3 Chennai       1  4.41e 9  4413830796. 12.7       3.89e- 4
 4 Chennai     881  3.07e11   348064656. NA         NA
 5 Delhi         1  8.07e 9  8068641713. 22.1       3.22e- 6
 6 Delhi       582  2.12e11   364964563. NA         NA
 7 Hyderabad     1  2.03e10 20265907358. 71.3       1.30e-16
 8 Hyderabad   861  2.45e11   284250987. NA         NA
 9 Kolkata       1  1.28e 8   128011544.  2.21      1.37e- 1
10 Kolkata     517  2.99e10    57803432. NA         NA
11 Mumbai        1  3.53e 6     3527028.  0.000643  9.80e- 1
12 Mumbai      957  5.25e12  5481871529. NA         NA
```

From the violin plot and point plot, we can analyse that the Rent price will not have a significant difference when the Tenant Preferred changes. In each city, the rent price are similar for each Tenant Preferred value. Also, the correlation of Rent and Tenant Preferred for each city is very close to zero. Although the p-value from ANOVA testing is lower than 0.05, but it is still higher than the p-value from BHK and rent. Hence, as result, Tenant Preferred to have less impact compared to BHK towards Rent.

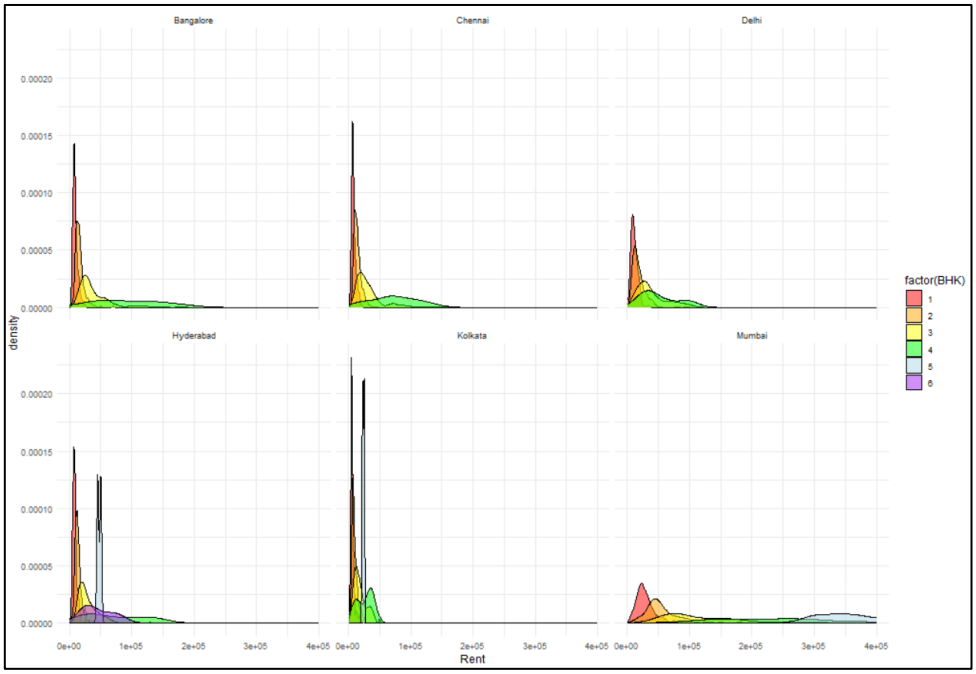
8.1.10 Question 9: How is the relationship changes between BHK and Rent with different Point of Contact in each city.

Density Plot

```
#Question 9: How is the relationship changes between BHK and Rent with different Point of Contact in each cities
ggplot(HouseRent_Cleaned, aes(x = Rent, fill = Point.of.Contact)) +
  geom_density(alpha = 0.5) +
  labs(x = "Rent") +
  scale_fill_manual(values = c("Contact Owner" = "darkblue", "Contact Builder" = "blue", "Contact Agent" = "lightblue")) +
  facet_wrap(~ City) +
  theme_minimal()
```




```
ggplot(HouseRent_Cleaned, aes(x = Rent, fill = factor(BHK))) +
  geom_density(alpha = 0.5) +
  labs(x = "Rent") +
  scale_fill_manual(values = c('1' = 'red', '2' = 'orange', '3' = 'yellow', '4' = 'green', '5' = 'lightblue', '6' = 'purple')) +
  facet_wrap(~ City) +
  theme_minimal()
```



Correlation

```
HouseRent_Cleaned_Preprocessed %>%
  group_by(City) %>%
  summarize(correlation = cor(Point.of.Contact, Rent))
```

```
> HouseRent_Cleaned_Preprocessed %>%
+   group_by(City) %>%
+   summarize(correlation = cor(Point.of.Contact, Rent))
# A tibble: 6 × 2
  City      correlation
  <chr>           <dbl>
1 Bangalore       0.482
2 Chennai         0.565
3 Delhi           0.353
4 Hyderabad       0.521
5 Kolkata         0.210
6 Mumbai          0.297
```

ANOVA Testing

```
HouseRent_Cleaned_Preprocessed %>%
  group_by(City) %>%
  do(anova(lm(Rent ~ Point.of.Contact, data = .)))
```

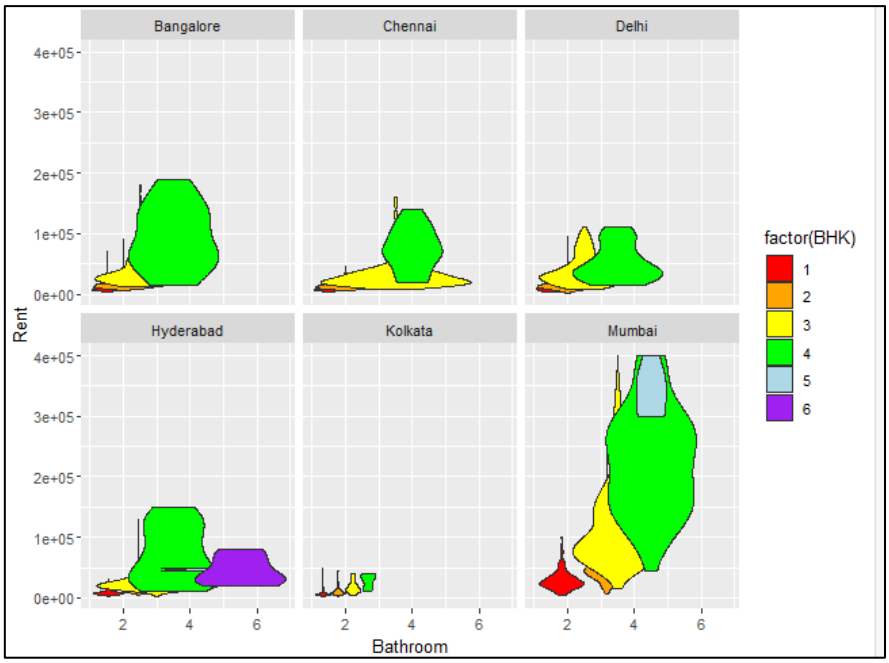
```
> HouseRent_Cleaned_Preprocessed %>%
+   group_by(City) %>%
+   do(anova(lm(Rent ~ Point.of.Contact, data = .)))
# A tibble: 12 × 6
# Groups:   City [6]
   City         Df `Sum Sq`     `Mean Sq` `F value`  `Pr(>F)`
   <chr>     <int>    <dbl>         <dbl>     <dbl>     <dbl>
 1 Bangalore     1  9.69e10  96936844927.      266.   1.69e-52
 2 Bangalore   880  3.20e11    364026068.       NA   NA
 3 Chennai       1  9.93e10  99285376384.      413.   1.34e-75
 4 Chennai     881  2.12e11    240378452.       NA   NA
 5 Delhi         1  2.76e10  27550088276.       83.1  1.25e-18
 6 Delhi       582  1.93e11    331491287.       NA   NA
 7 Hyderabad     1  7.18e10  71804321318.      320.   4.29e-61
 8 Hyderabad   861  1.93e11    224392202.       NA   NA
 9 Kolkata       1  1.32e 9   1324587895.       23.9  1.38e- 6
10 Kolkata     517  2.87e10     55488971.       NA   NA
11 Mumbai        1  4.61e11 461265854896.       92.3  6.45e-21
12 Mumbai      957  4.78e12   4999883726.       NA   NA
```

In this section, I compared the relationship for both BHK and Point of Contact towards Rent price. As a result, some of the cities do have a high relationship between Point of Contact and Rent price, such as Bangalore, Chennai, and Hyderabad. Overall, in correlation and ANOVA testing results, the relation for Point of contact is slightly lower than BHK.

8.1.11 Question 10: How is the relationship between Rent and BHK with a change of Bathroom for each city.

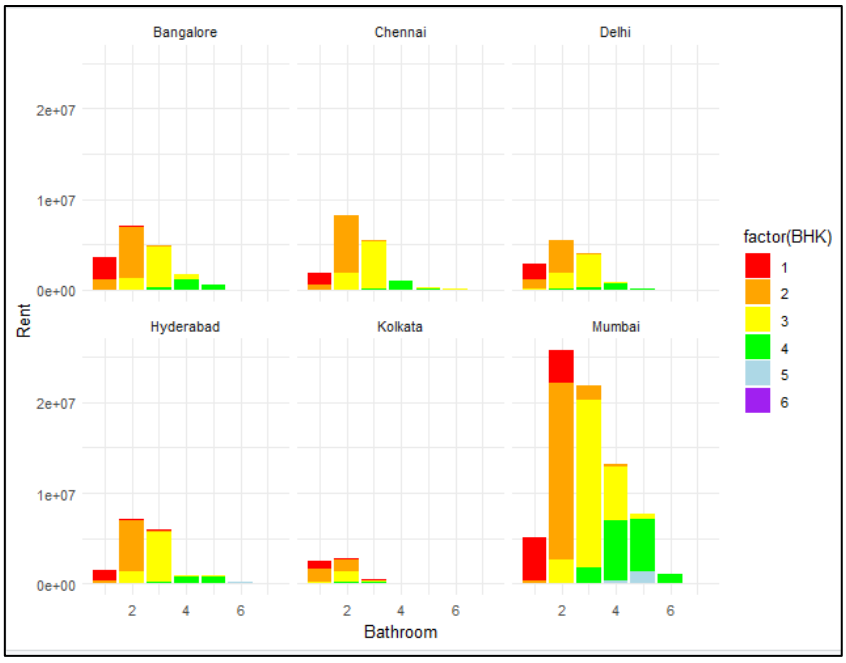
Violin Plot

```
ggplot(HouseRent_Cleaned, aes(x = Bathroom, y = Rent, fill = factor(BHK))) +
  geom_violin(scale = "width", trim = TRUE, position = 'dodge') +
  labs(x = "Bathroom", y = "Rent") +
  scale_fill_manual(values = c('1' = 'red', '2' = 'orange', '3' = 'yellow', '4' = 'green', '5' = 'lightblue', '6' = 'purple'))+
  facet_wrap(~City)
```



Bar Chart

```
ggplot(HouseRent_Cleaned, aes(x = Bathroom, y = Rent, fill = factor(BHK))) +
  geom_bar(stat = "identity") +
  labs(x = "Bathroom", y = "Rent") +
  scale_fill_manual(values = c('1' = 'red', '2' = 'orange', '3' = 'yellow', '4' = 'green', '5' = 'lightblue', '6' = 'purple')) +
  facet_wrap(~ City) +
  theme_minimal()
```



Correlation

```
HouseRent_Cleaned_Preprocessed %>%
  group_by(City) %>%
  summarize(correlation = cor(Bathroom, Rent))
```

```
> HouseRent_Cleaned_Preprocessed %>%
+   group_by(City) %>%
+   summarize(correlation = cor(Bathroom, Rent))
# A tibble: 6 × 2
  City      correlation
  <chr>           <dbl>
1 Bangalore       0.674
2 Chennai         0.587
3 Delhi           0.718
4 Hyderabad       0.626
5 Kolkata         0.562
6 Mumbai          0.740
```

ANOVA testing

```
HouseRent_Cleaned_Preprocessed %>%
  group_by(City) %>%
  do(anova(lm(Rent ~ Bathroom, data = .)))
```

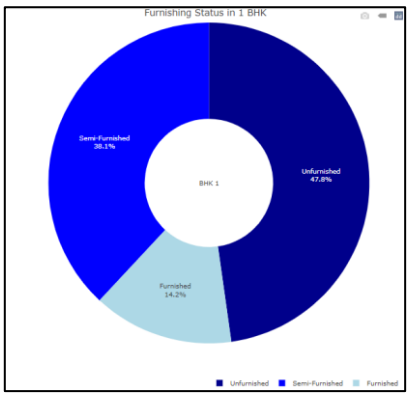
```
> HouseRent_Cleaned_Preprocessed %>%
+   group_by(City) %>%
+   do(anova(lm(Rent ~ Bathroom, data = .)))
# A tibble: 12 × 6
# Groups:   City [6]
   City         Df `Sum Sq` `Mean Sq` `F value`   `Pr(>F)`
   <chr>     <int>    <dbl>     <dbl>     <dbl>      <dbl>
 1 Bangalore     1  1.90e11   1.90e11      733.  6.54e-118
 2 Bangalore   880  2.28e11   2.59e 8       NA  NA
 3 Chennai       1  1.07e11   1.07e11      464.  5.59e- 83
 4 Chennai     881  2.04e11   2.31e 8       NA  NA
 5 Delhi         1  1.14e11   1.14e11      620.  1.02e- 93
 6 Delhi       582  1.07e11   1.83e 8       NA  NA
 7 Hyderabad     1  1.04e11   1.04e11      554.  6.07e- 95
 8 Hyderabad   861  1.61e11   1.87e 8       NA  NA
 9 Kolkata       1  9.47e 9   9.47e 9      238.  1.71e- 44
10 Kolkata     517  2.05e10   3.97e 7       NA  NA
11 Mumbai        1  2.87e12   2.87e12     1155.  1.08e-166
12 Mumbai      957  2.38e12   2.48e 9       NA  NA
```

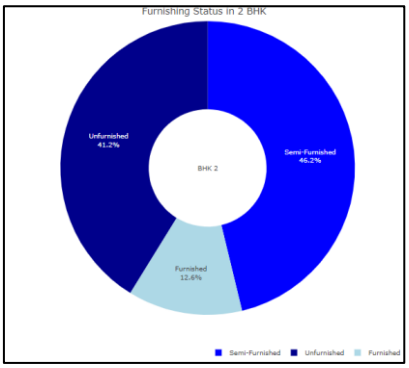
Above analysis shows the relation of BHK and Rent with the changes of Bathroom. From the plots and charts, Rent price do increases when number of bathroom increases. The most obvious changes falls at Mumbai city and the least falls at Kolkata city. This also explain why the highest correlation between Bathroom and Rent is in Mumbai city and the least is in Kolkata city. In summary, we found out that Bathroom have a higher relation compared to BHK towards Rent price.

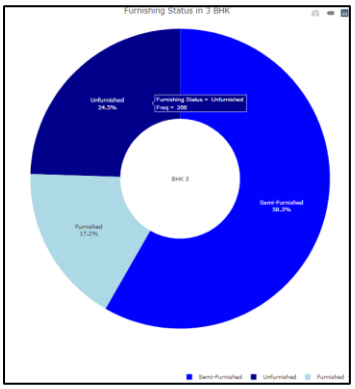
8.1.12 Question 11: How is the relationship between BHK and Rent with different furnishing status in each city.

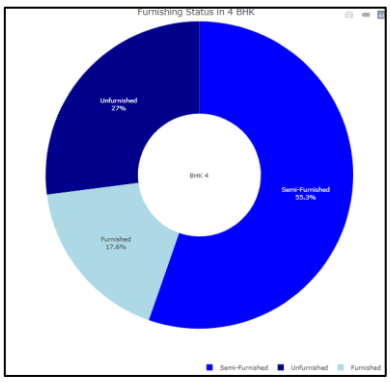
Donut Plot (To show the distribution of Furnishing status in different number of BHK)

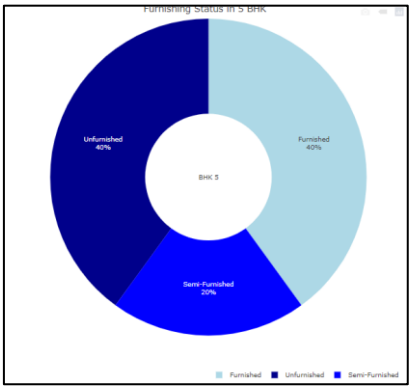
```
unique_BHK = unique(HouseRent_Cleaned$BHK)

BHK_subsets <- setNames(replicate(length(unique_BHK), NULL), unique_BHK)

for(i in unique_BHK){
  BHK_subset = HouseRent_Cleaned %>% filter(BHK == i)
  BHK_subsets[[i]] = BHK_subset
}

#BHK 1
{
BHK_1_fur = as.data.frame(table(BHK_subsets$'1'$Furnishing.Status))
names(BHK_1_fur)[1] = "Furnishing.Status"

BHK1 = plot_ly(
  data = BHK_1_fur,
  labels = ~Furnishing.Status,
  values = ~Freq,
  type = 'pie',
  hole = 0.4,
  textinfo = "label+percent",
  hoverinfo = 'text',
  text = ~paste('Furnishing Status = ',Furnishing.Status, '\nFreq = ',Freq),
  marker = list(colors = c("lightblue", "blue", "darkblue"))
)%>%
  layout(
    showlegend = TRUE,
    title = "Furnishing Status in 1 BHK", # Set the title
    legend = list(orientation = "h", x = 0.5, y = -0.1), # Customize the legend position
    annotations = list(text = "BHK 1", x = 0.5, y = 0.5, showarrow = FALSE) # Annotation
  )}
```

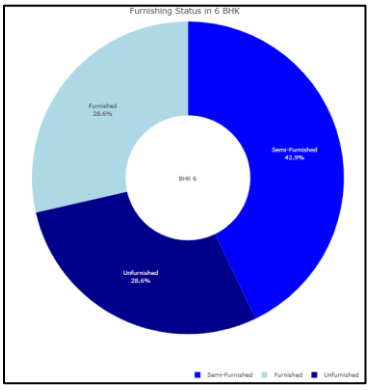






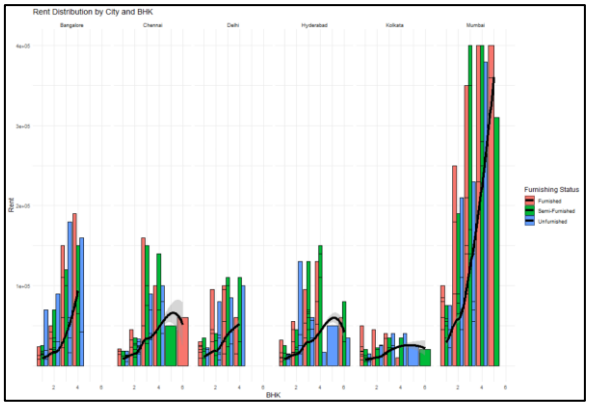






Bar Chart

```
ggplot(HouseRent_Cleaned, aes(x = BHK, y = Rent, fill = factor(Furnishing.Status))) +
  geom_bar(stat = "identity", position = "dodge", color="Black") +
  ggtitle("Rent Distribution by City and BHK") +
  labs(x = "BHK", y = "Rent") +
  scale_fill_discrete(name = "Furnishing Status") +
  scale_y_continuous(breaks = (seq(100000, max(HouseRent_Cleaned$Rent), by = 100000))) +
  geom_smooth(aes(group = City), method = loess, color = 'black', linewidth = 1.5)+
  facet_grid(~City)+
  theme_minimal()
```



Correlation

```
HouseRent_Cleaned_Preprocessed %>%
  group_by(City) %>%
  summarize(correlation = cor(Furnishing.Status, Rent))
```

```
# A tibble: 6 × 2
  City       correlation
  <chr>            <dbl>
1 Bangalore        0.151
2 Chennai          0.251
3 Delhi            0.147
4 Hyderabad        0.267
5 Kolkata          0.119
6 Mumbai           0.260
```

ANOVA testing

```
HouseRent_Cleaned_Preprocessed %>%
  group_by(City) %>%
  do(anova(lm(Rent ~ Furnishing.Status, data = .)))
```

```
# Groups:   City [6]
   City          Df `Sum Sq`     `Mean Sq` `F value`  `Pr(>F)`
   <chr>      <int>    <dbl>         <dbl>     <dbl>     <dbl>
 1 Bangalore      1  9.55e 9   9547621525.     20.6   6.42e- 6
 2 Bangalore    880  4.08e11    463332004.     NA     NA
 3 Chennai        1  1.96e10  19583290649.     59.2   3.83e-14
 4 Chennai      881  2.91e11    330846200.     NA     NA
 5 Delhi          1  4.79e 9   4792800462.     12.9   3.50e- 4
 6 Delhi        582  2.16e11    370593156.     NA     NA
 7 Hyderabad      1  1.90e10  18954400496.     66.3   1.34e-15
 8 Hyderabad    861  2.46e11    285774224.     NA     NA
 9 Kolkata        1  4.29e 8    428530231.      7.49  6.42e- 3
10 Kolkata      517  2.96e10     57222158.     NA     NA
11 Mumbai         1  3.55e11 354784524068.     69.4   2.75e-16
12 Mumbai       957  4.89e12   5111149484.     NA     NA
```

Above images shows the last attribute, which is Furnishing Status. First pie chart shows the furnishing status of each city, and there is no significance pattern of which furnishing status houses have the highest percentage among all the cities. Second bar chart shows the rent price for each furnishing status and again, there is no pattern showing which furnishing status will have higher Rent. Hence, as a result, the correlation of Furnishing Status towards Rent is not as significance and BHK.

8.1.13 Additional Features

- geom_text() : Use to labelling graph

    -aes(label = ..count..) : Specify the mapping to display count of observation

    - stat = “count” : Specify the type of statistical transformation, which the count should be calculated

    - vjust = -0.5 : Adjust the vertical position of text label

    - color = “black” : change text color

    - size = 3 : adjust the size of text

- scale_x_continuous()/scale_y_continuous() : adjust each column in x/y-axis to follow the sequence

- breaks : Create the sequence for scale_x/y_continuous()

-seq(0, max(HouseRent_Cleaned$Rent), by = 50000): Create a number sequence that starts from 0, ends at max Rent price, and each column separate by 50000

-scale_fill_discrete(name = “BHK”) : Name the discrete legend bar as BHK

-facet_grid(~City) : Create a chart that separated by City, and define the position by row and column

-theme_minimal() : Clear all non-data display to have a clearer view of data

-scale_color_discrete(name = “BHK”) : Name the color legend bar as BHK

- stat_summary() : to complete the summary statistic and display the statistic using graph

    - fun = median: Specify the summary function to applied to data, which in this case we use median value of each group

    - geom = ‘text’ : set the ‘text’ to indicate that the summary statistic will be displayed as text label

    - aes(label = round(..y..,2)) : to display the median rounded to two decimal place

    - position = position_dodge(width = 0.75) : to separate the text label for each group , with a width of 0.75

-scale_fill_manual(value = c(“Super Area” = “lightblue”)) = To fill the color manually

-scale_size_continuous(range = c(5,15)): to adjust the size of point in bubble chart, in this case, smallest value will have size of 5, and largest will have size of 15

-summarize(correlation = cor(Area.Type, Rent)) = summarize use to complete summary statistic in data frame, whereas cor() will provide the correlation between the variable passed

-do(): Use as a conjunction with the pipe %>% to perform a function

    - anova() : to perform an analysis which will return the variance and p-value of the model

- lm(Rent ~ Area.Type): To perform a liner regression of model that Rent act as the dependent and Area.Type act as independent variable

- anova(lm(Rent ~ Area.Type)) : The anova function will perform an analysis based on the model of Rent and Area Type. It will also provide the p-value, which examine the impact level of Area Type towards Rent

-ggpairs() : To do multiple plots with a given data set

-columns = c("BHK","Rent") = column that required to do ggpairs from a dataset

-aes(color = City) : create the plots using different color which represent city

-geom_violin(): Create a violin plot

-scale = "width" : specify how the width of the violin should be scale

- trim = FALSE: control whether the tails of the violin plot should be trimmed. In this case it is FALSE, mean the violin will show the range of the data in each tail.

-geom_density(): Create a density plot

-alpha = 0.5: Control the transparency from 0 (total transparent) to 1 (total opaque). In this case, 0.5 of transparency allow overlapping areas to be seen more clearly.

-plot_ly() : Create a plotly plot

-label = ~Furnishing Status : Specify the pie chart to use Furnishing Status as label

-values = ~Freq ; Specify the pie chart to use frequency as value

- type = 'pie' : Specify the plotly plot to be pie chart

- hole = '0.4' : set the size of the hole in the center of a pie chart, to create a donut chart

- textinfo = 'label+percent' : To show the label and percent on each pie slice

- hoverinfo = 'text' : To control the format of information provided when cursor moved to the pie slice.

- text = ~paste(): create custom text for each slice

- marker = list(colors = c('lightblue','blue','darkblue') = To specify the color of each pie slice

- layout(): Use to customize the layout of the plot

- showlegend = TRUE: control the legend to be shown

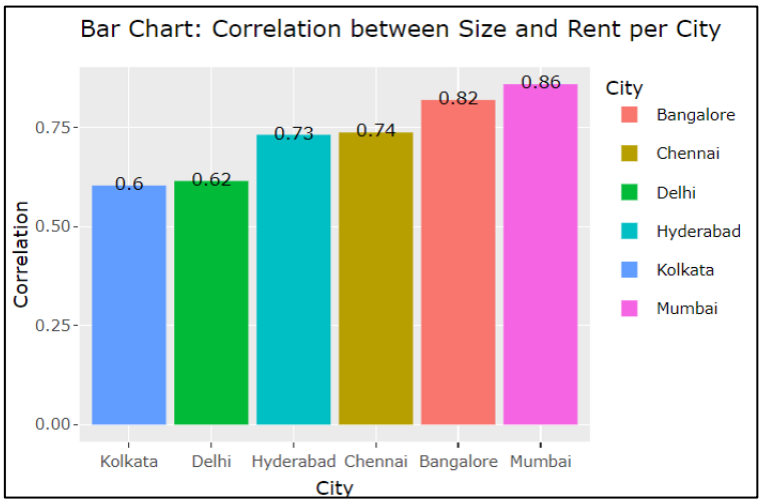
- legend = list(): Control the position of the legend

- annotations = list(): adds an annotation to the plot

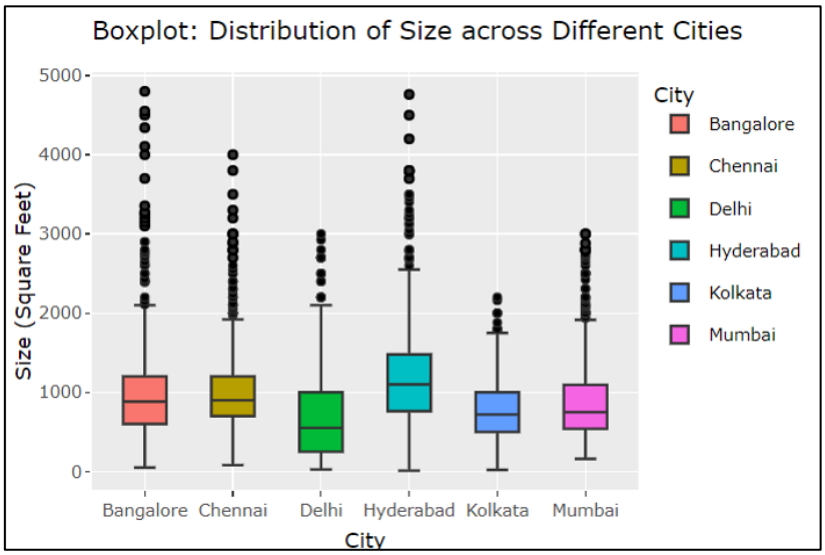
8.2 Objective 2: Investigate the relationship between size and rent across different cities while considering external factors. (William Odhiambo TP063033)

8.2.1 Question 1: How does the relationship between size and rent vary across different cities, and what is the average rent per city?

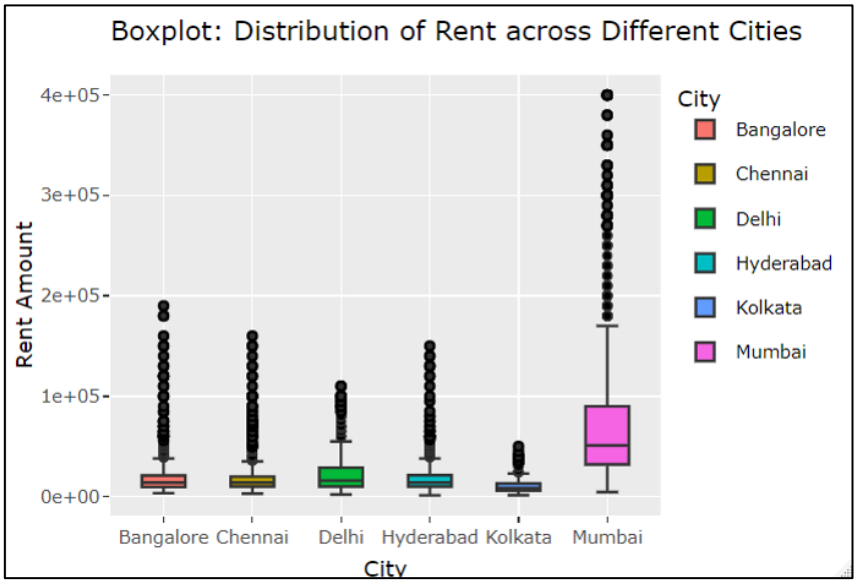
```
# Calculate the correlation between size and rent overall
correlation_size_rent <- cor(HouseRent_Cleaned$Size, HouseRent_Cleaned$Rent)

# Print the overall correlation
cat("Correlation between Size and Rent:", correlation_size_rent, "\n")

# Calculate the correlation between size and rent for each city
correlation_size_rent_city <- HouseRent_Cleaned %>%
  group_by(City) %>%
  summarise(Correlation = cor(Size, Rent))

# Print the correlation for each city
for (i in seq_len(nrow(correlation_size_rent_city))) {
  cat("Correlation between Size and Rent for", correlation_size_rent_city$City[i], ":", correlation_size_rent_city$Correlation[i], "\n")
}

# Create a bar chart to visualize the correlations per city
correlation_size_rent_city_bar <- ggplot(correlation_size_rent_city, aes(x = reorder(City, Correlation), y = Correlation, fill = City)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = round(Correlation, 2)), vjust = -0.5, size = 3.5) +
  labs(title = "Bar Chart: Correlation between Size and Rent per City", x = "City", y = "Correlation")

correlation_size_rent_city_bar <- ggplotly(correlation_size_rent_city_bar)
correlation_size_rent_city_bar
```



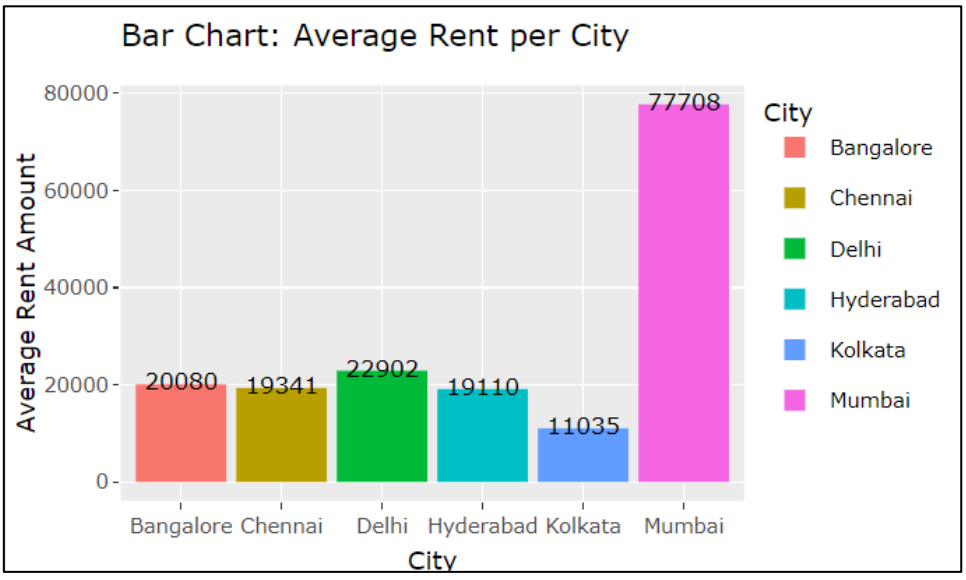

```
size_city_boxplot <- ggplot(HouseRent_Cleaned, aes(x = City, y = Size, fill = City)) +
  geom_boxplot() +
  labs(title = "Boxplot: Distribution of Size across Different Cities", x = "City", y = "Size (Square Feet)")

size_city_boxplot <- ggplotly(size_city_boxplot)
size_city_boxplot
```



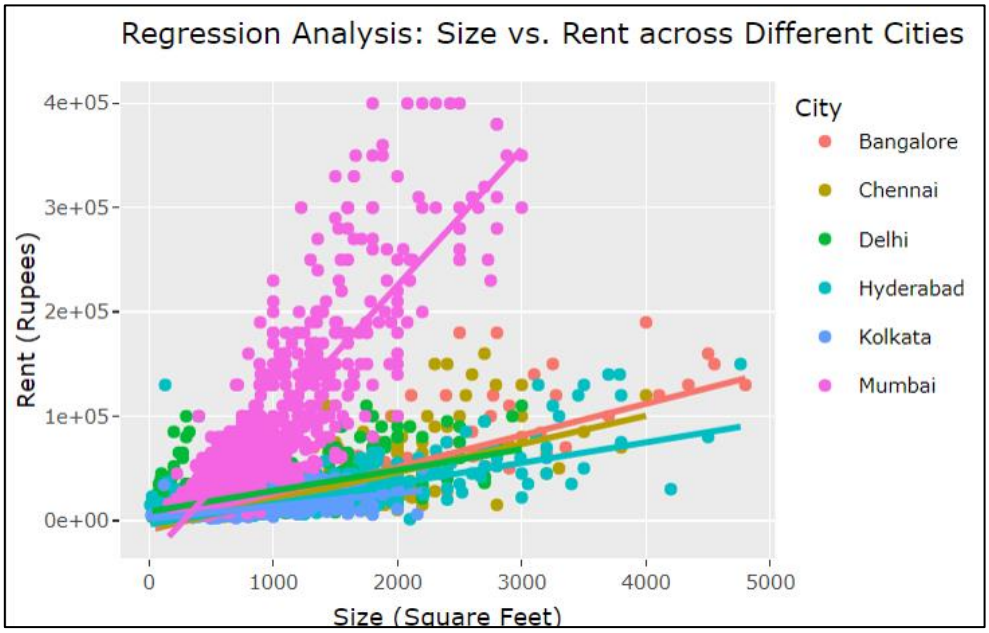
```
rent_city_boxplot <- ggplot(HouseRent_Cleaned, aes(x = City, y = Rent, fill = City)) +
  geom_boxplot() +
  labs(title = "Boxplot: Distribution of Rent across Different Cities", x = "City", y = "Rent Amount")

rent_city_boxplot <- ggplotly(rent_city_boxplot)
rent_city_boxplot
```




```
# Calculate the average rent for each city
avg_rent_city <- HouseRent_Cleaned %>%
  group_by(City) %>%
  summarise(Avg_Rent = mean(Rent))

# Bar chart to show average rent per city
avg_rent_city_bar <- ggplot(avg_rent_city, aes(x = City, y = Avg_Rent, fill = City)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = round(Avg_Rent)), vjust = -0.5, size = 3.5) +
  labs(title = "Bar Chart: Average Rent per City", x = "City", y = "Average Rent Amount")

avg_rent_city_bar <- ggplotly(avg_rent_city_bar)
avg_rent_city_bar
```



```
size_rent_scatter <- ggplot(HouseRent_Cleaned, aes(x = Size, y = Rent, color = City)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Regression Analysis: Size vs. Rent across Different Cities", x = "Size (Square Feet)", y = "Rent (Rupees)")

# Convert ggplot to plotly object
size_rent_scatter <- ggplotly(size_rent_scatter)
size_rent_scatter
```



General Observations

1. **General Trend**: In all cities, there's a positive correlation between size and rent. This means as properties get larger, their rent generally increases.
2. **Strong Correlation in Mumbai**: Mumbai stand out with the strongest positive correlation and steepest regression line, indicating that size plays a significant role in determining rent in this city. Larger properties in Mumbai are much more likely to command higher rents.
3. **Significant Correlations in Hyderabad and Delhi:** Both cities show strong positive correlations, suggesting that property size significantly influences rent amounts in these locations.
4. **Moderate Correlations:** Chennai and Kolkata display moderate correlations, implying a relationship between property size and rent, but other factors might also play a role.
5. Lowest Correlation in Bangalore: Despite being a major city, Bangalore show the weakest correlation among the cities listed. This suggests that while size is a factor, other elements (e.g., location, amenities, property condition) might have a more substantial impact on rent in Bangalore.
6. **Mumbai city** exhibits a steeper slope in their regression lines, suggesting a more significant increase in rent with increasing property size.
7. **Variability**: There's considerable variability in rents, especially for larger properties, as seen by the spread of points. Likely because of factors like location, amenities, age of the property, etc.

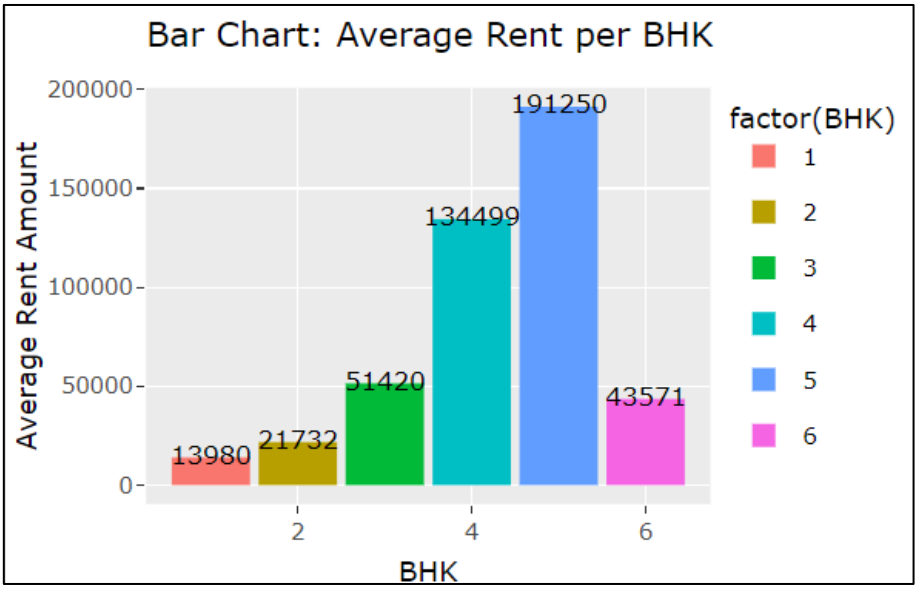
Observations of Average rent per city

1. **Mumbai**: Stands out with the highest average rent among all cities. This is consistent with the general perception of Mumbai being one of the most expensive cities for real estate in India.
2. **Delhi**: Follows Mumbai with the second-highest average rent. The capital city, with its central location and significance, has a premium on property rentals.
3. **Bangalore & Chennai**: These cities exhibit moderate average rents, lower than Mumbai and Delhi but higher than Hyderabad and Kolkata.
4. **Hyderabad & Kolkata**: These cities have the lowest average rents among the analysed cities. This could be attributed to various factors, including economic conditions, demand-supply dynamics, and the nature of residential areas

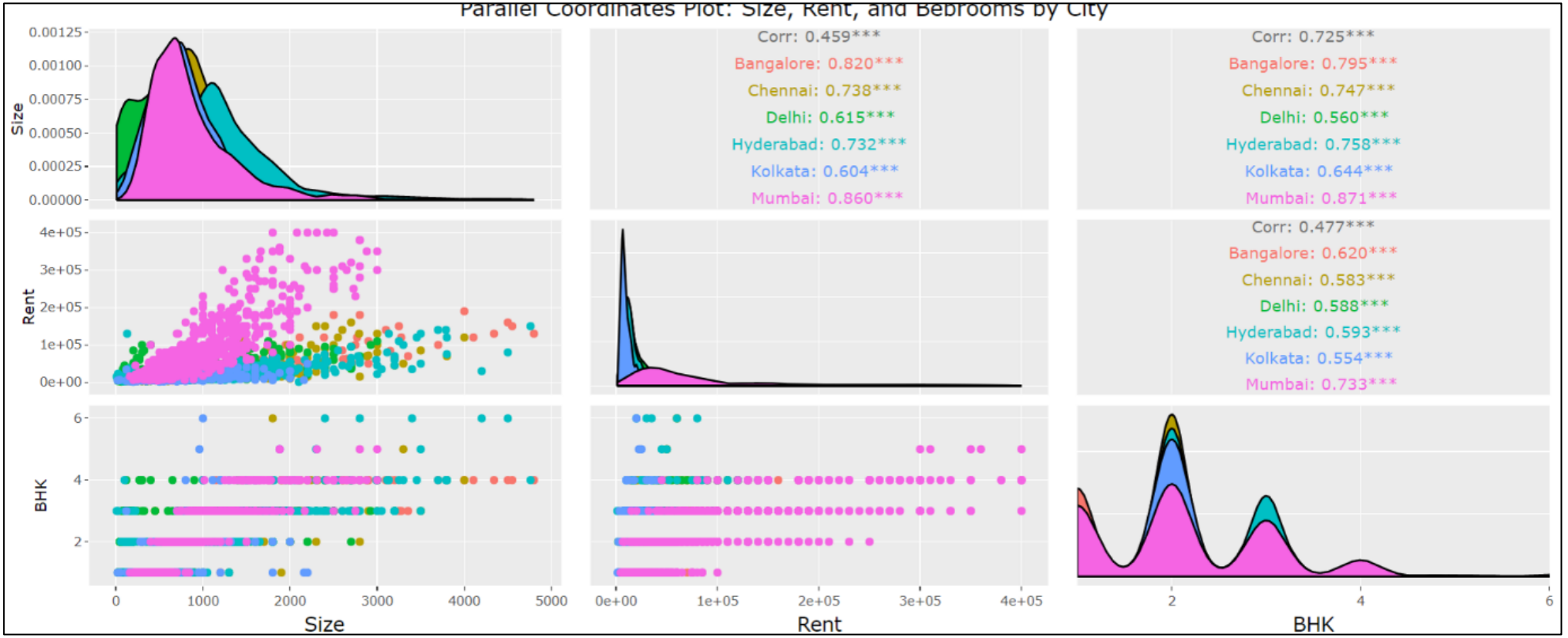
8.2.2 Question 2: How does the relationship between size and rent vary across different Bedrooms, and what is the average rent per Bedroom?

```
# Investigate how the relationship between size and rent changes with the number of bedrooms using violin plot
bedroom_size_rent_violin <- ggplot(HouseRent_Cleaned, aes(x = factor(BHK), y = Rent, fill = factor(BHK))) +
  geom_violin(alpha = 0.7) +
  labs(title = "Size & Rent by Number of Bedrooms",
       x = "Number of Bedrooms", y = "Rent (Rupees)") +
  theme_minimal()

ggplotly(bedroom_size_rent_violin)
```


```
# Calculate the average rent for each BHK
avg_rent_BHK <- HouseRent_Cleaned %>%
  group_by(BHK) %>%
  summarise(Avg_Rent = mean(Rent))

# Bar chart to show average rent per BHK
avg_rent_BHK_bar <- ggplot(avg_rent_BHK, aes(x = BHK, y = Avg_Rent, fill = factor(BHK))) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = round(Avg_Rent)), vjust = -0.5, size = 3.5) +
  labs(title = "Bar Chart: Average Rent per BHK", x = "BHK", y = "Average Rent Amount")

avg_rent_BHK_bar <- ggplotly(avg_rent_BHK_bar)
avg_rent_BHK_bar
```



```
parallel_coordinates_plot_BHK <- ggpairs(HouseRent_Cleaned,
                                         columns = c("Size", "Rent", "BHK"),
                                         mapping = aes(color = City),
                                         title = "Parallel Coordinates Plot: Size, Rent, and Bebrooms by City")

# Convert to plotly object
parallel_coordinates_plot_BHK <- ggplotly(parallel_coordinates_plot_BHK)
parallel_coordinates_plot_BHK
```



General observations

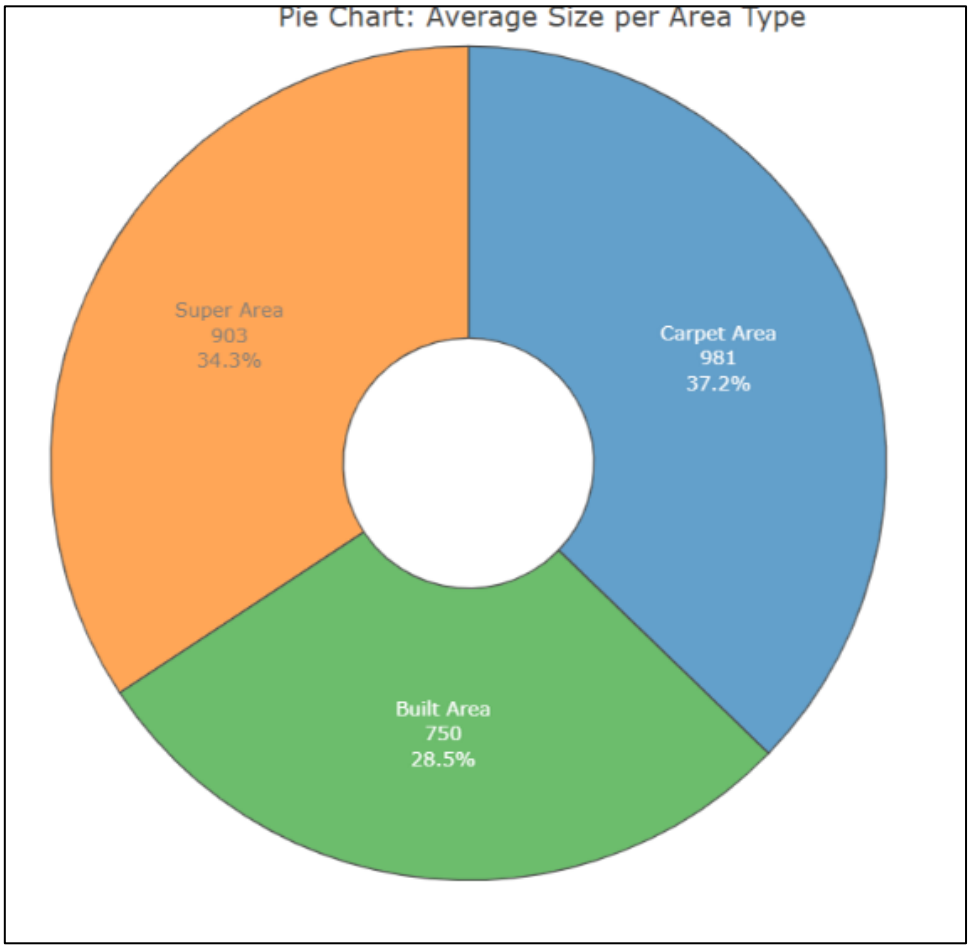
1. **Positive Correlation**: Across all bedrooms, there seems to be a positive correlation between size and rent. As the size of the property increases, the rent generally also increases.
2. **BHK Categories**: Different bedroom categories occupy distinct regions on the parallel coordinates plot. For instance, 1-bedroom properties tend to be smaller and less expensive, while 3- or 4-bedroom properties tend to be larger and command higher rents.
3. **Overlap in Rent Ranges**: While there's a general trend where higher bedroom properties command higher rents, there's overlap, especially between 2, 3, and 4 bedroom properties. For instance, some large 2-bedroom properties have rents like smaller 3-bedroom properties.
4. **Distribution Width**: The width of the violin plots indicates the density of properties at various rent values. A wider section of the plot suggests a higher concentration of properties at that rent range.
5. **Mumbai & Delhi:** These cities show a wide distribution for 1 and 2 BHK properties, suggesting a diverse rental market with properties available across a broad range of prices.
6. **Bangalore**: This city exhibits narrower distributions for higher BHKs, indicating a more consistent rental range for larger properties.
7. **Spread**: The range of the violin plots indicate the spread of the rents. Cities like Mumbai and Delhi have a broader spread, especially for 1 and 2 BHKs, showing properties with both low and high rents.

Observations on average rent per bedroom

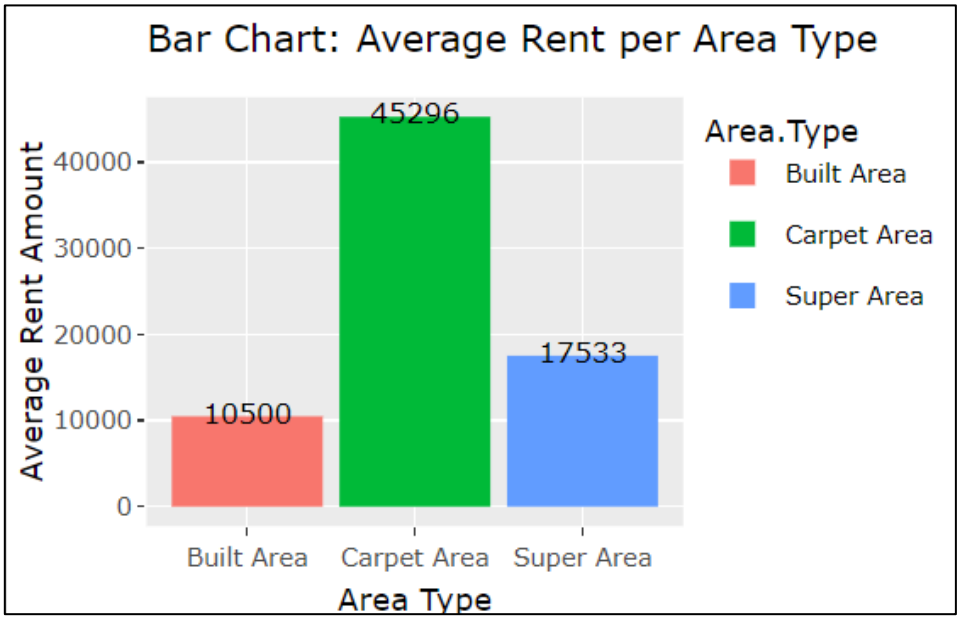
- 5 BHK properties have the highest average rent, followed closely by 4 BHKs.
- 6 BHKs have a slightly lower average rent, which might be due to fewer samples in the dataset or specific locations where these properties are found.
- As expected, the average rent decreases as the number of BHKs decreases, with 1 BHK properties having the lowest average rent

8.2.3 Question 3: How does the relationship between size and rent change with different area types: Super, Carpet, or Build)?

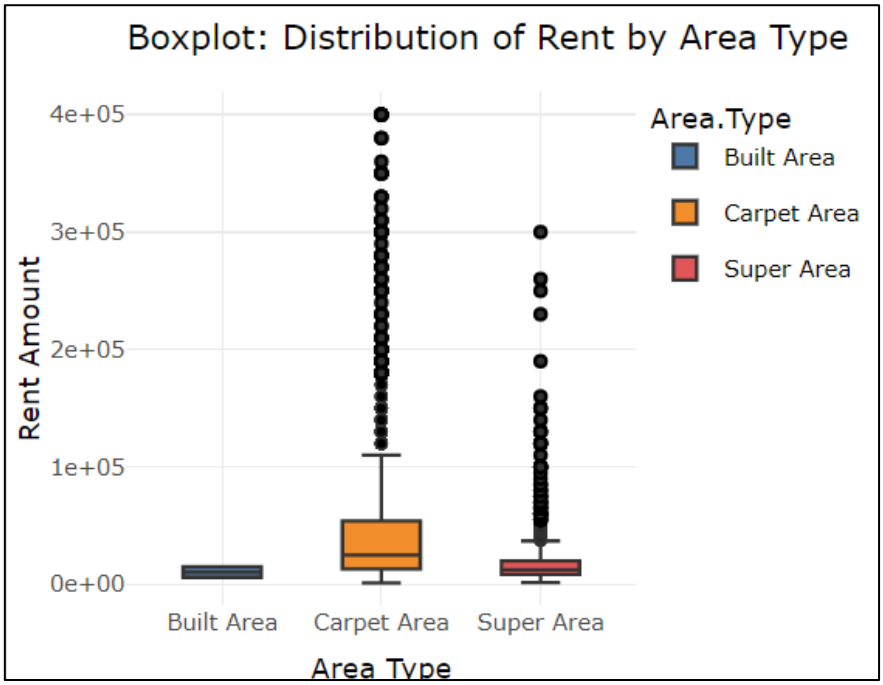
```
avg_size_area_type_pie <- plot_ly(
  labels = avg_size_area_type$Area.Type,
  values = round(avg_size_area_type$Avg_Size),
  type = "pie",
  opacity = 0.7,
  marker = list(line = list(color = "#000000", width = 1)),
  textinfo = "label+percent+value",
  textposition = "inside",
  hole = 0.3
)

# Add layout options
avg_size_area_type_pie <- avg_size_area_type_pie %>%
  layout(
    title = "Pie Chart: Average Size per Area Type",
    scene = list(
      aspectmode = "data",
      camera = list(
        eye = list(x = 1.5, y = 1.5, z = 1)
      )
    )
  )

# Show the pie chart
avg_size_area_type_pie
```



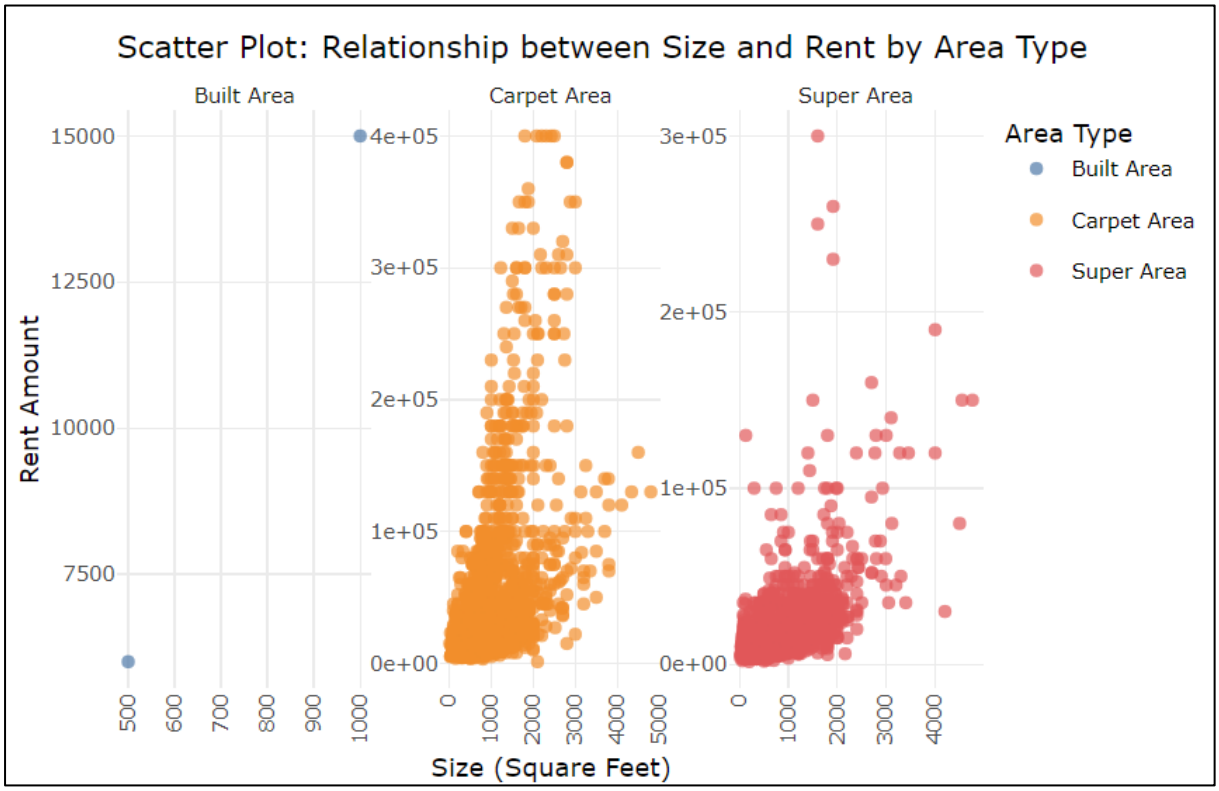
```
avg_rent_area_type_bar <- ggplot(avg_rent_area_type, aes(x = Area.Type, y = Avg_Rent, fill = Area.Type)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = round(Avg_Rent)), vjust = -0.5, size = 3.5) +
  labs(title = "Bar Chart: Average Rent per Area Type", x = "Area Type", y = "Average Rent Amount")

avg_rent_area_type_bar <- ggplotly(avg_rent_area_type_bar)
avg_rent_area_type_bar
```




```
area_type_boxplot <- ggplot(HouseRent_Cleaned, aes(x = Area.Type, y = Rent, fill = Area.Type)) +
  geom_boxplot() +
  labs(title = "Boxplot: Distribution of Rent by Area Type", x = "Area Type", y = "Rent Amount") +
  theme_minimal() +
  theme(legend.position = "bottom") +
  scale_fill_tableau()

ggplotly(area_type_boxplot)
```



```
library(ggthemes)
area_type_scatter <- ggplot(HouseRent_Cleaned, aes(x = Size, y = Rent, color = Area.Type)) +
  geom_point(alpha = 0.7) +
  labs(title = "Scatter Plot: Relationship between Size and Rent by Area Type",
       x = "Size (Square Feet)", y = "Rent Amount", color = "Area Type") +
  facet_wrap(~ Area.Type, scales = "free") +
  theme_minimal() +
  theme(legend.position = "bottom", axis.text.x = element_text(angle = 90, hjust = 1)) +  # Rotate x-axis ticks
  scale_color_tableau()

plotly_area_type_scatter <- ggplotly(area_type_scatter)

plotly_area_type_scatter
```



Built Area

- Built Area houses have fewer data points in our dataset, making the observed perfect correlation possibly less reliable.
- The fewer data points for Built Area could suggest that it is less commonly used as a measurement type in listings, or it might be more prevalent in certain regions or types of properties not well-represented in our dataset.

Super Area

- Super Area houses generally show a moderate positive relationship between size and rent, even though there are some outliers.
- Super Area includes the entire area along with shared spaces, which could make these properties more attractive to certain tenants, thus justifying higher rents.

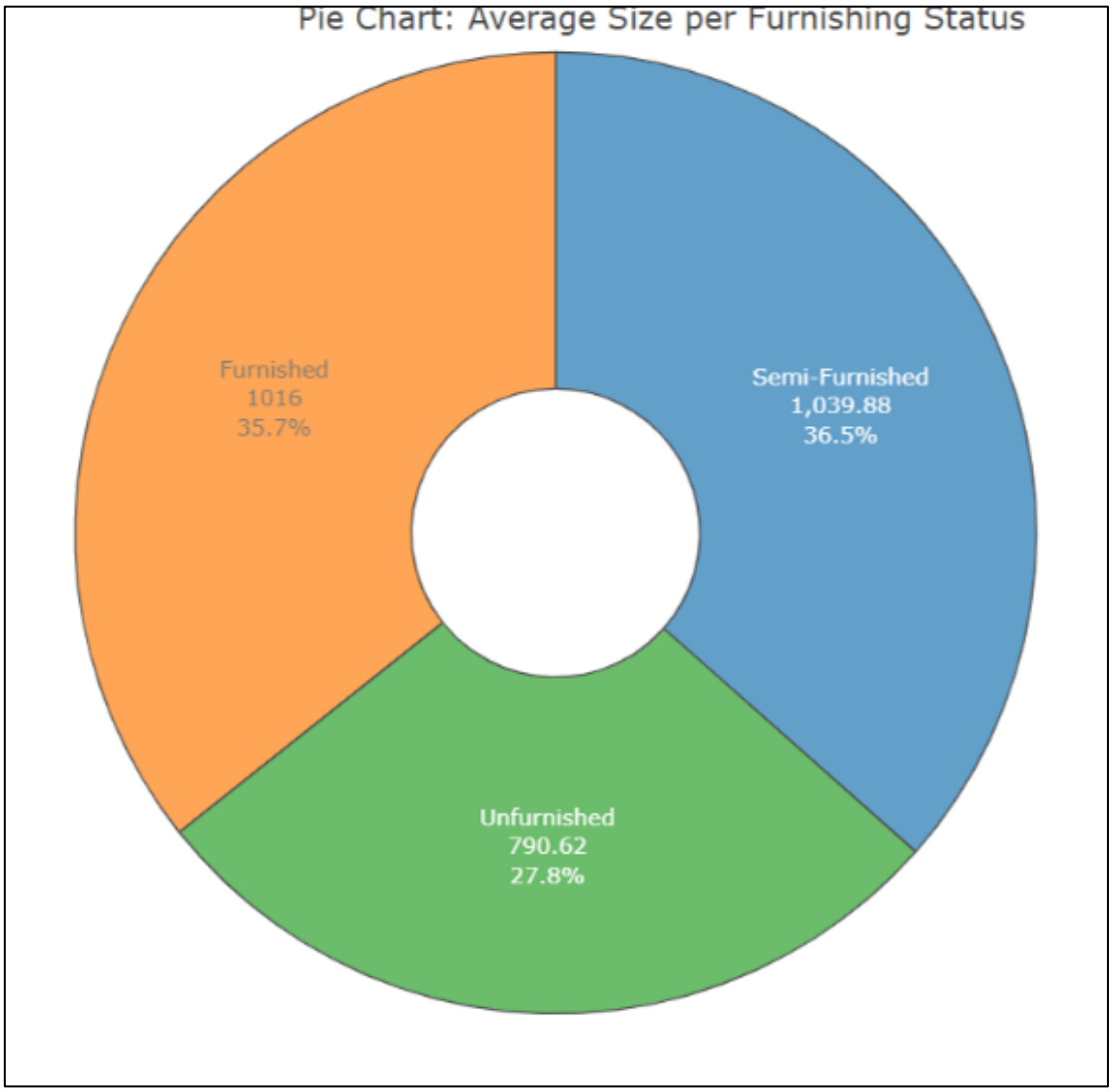
- The selection of properties listed with Super Area measurements might inherently be more premium, either due to location or other factors not captured in the dataset.

Carpet Area

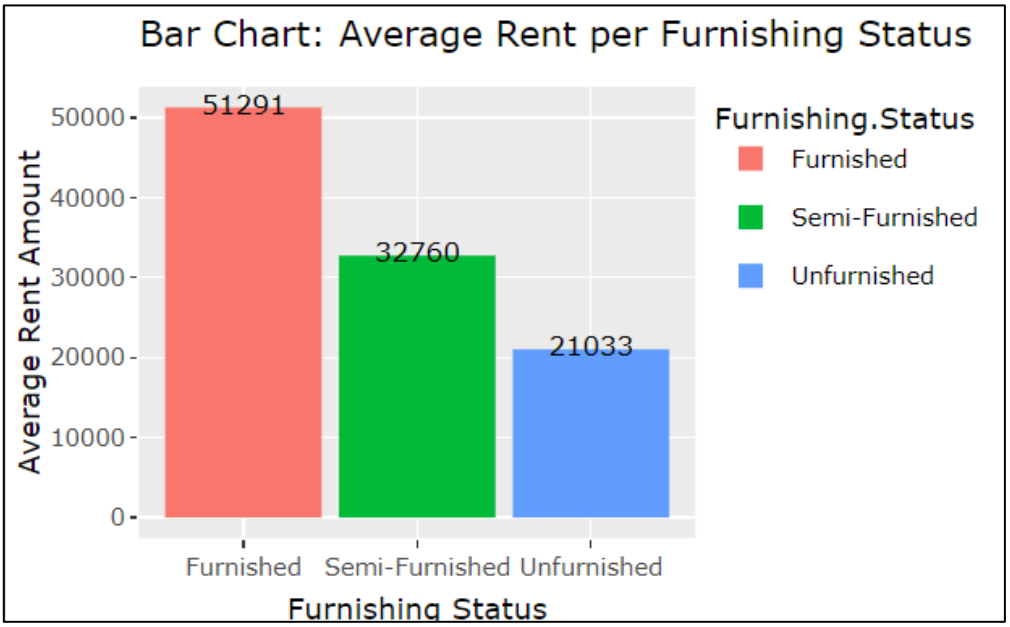
- Properties measured using the Carpet Area have a slightly weaker but still positive relationship between size and rent compared to Super Area.
- Carpet Area represents the usable area within the property, excluding balconies and walls. This might be more appealing to tenants who prioritise usable space, and therefore, properties with larger carpet areas might command higher rents.
- Like Super Area, properties listed with Carpet Area measurements might be in premium locations or offer certain amenities or features not captured in the dataset.

8.2.4 Question 4: How does the relationship between size and rent change with different furnishing types (Furnished, Semi-Furnished, or Unfurnished) for each city?

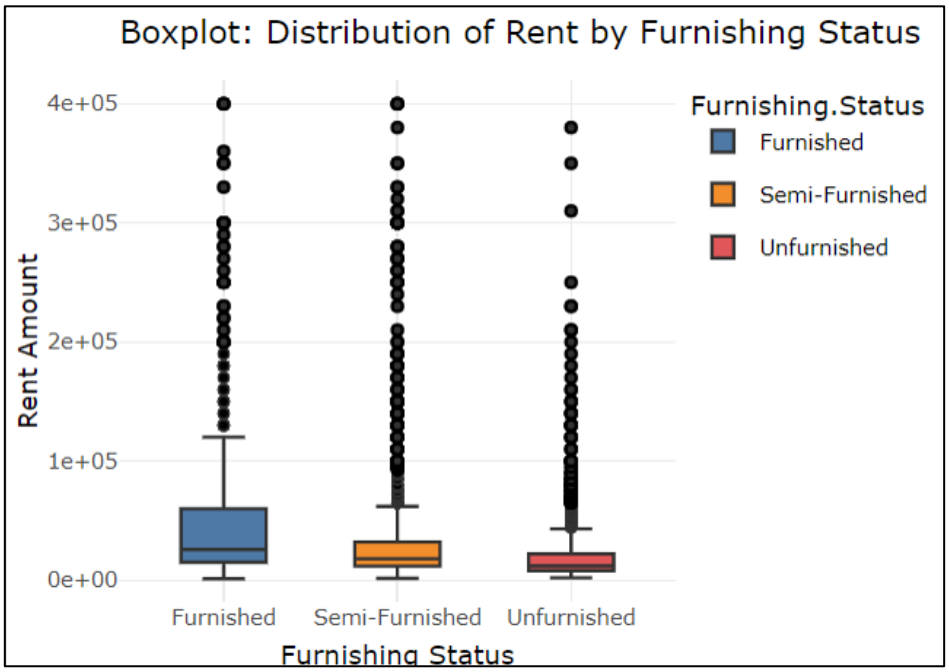
```
avg_size_furnishing_type_pie <- plot_ly(
  labels = avg_size_furnishing_type$Furnishing.Status,
  values = round(avg_size_furnishing_type$Avg_Size,2),
  type = "pie",
  opacity = 0.7,
  marker = list(line = list(color = "#000000", width = 1)),
  textinfo = "label+percent+value",
  textposition = "inside",
  hole = 0.3
)

# Add layout options
avg_size_furnishing_type_pie <- avg_size_furnishing_type_pie %>%
  layout(
    title = "Pie Chart: Average Size per Furnishing Status",
    scene = list(
      aspectmode = "data",
      camera = list(
        eye = list(x = 1.5, y = 1.5, z = 1)
      )
    )
  )

# Show the 3D-like pie chart
avg_size_furnishing_type_pie
```



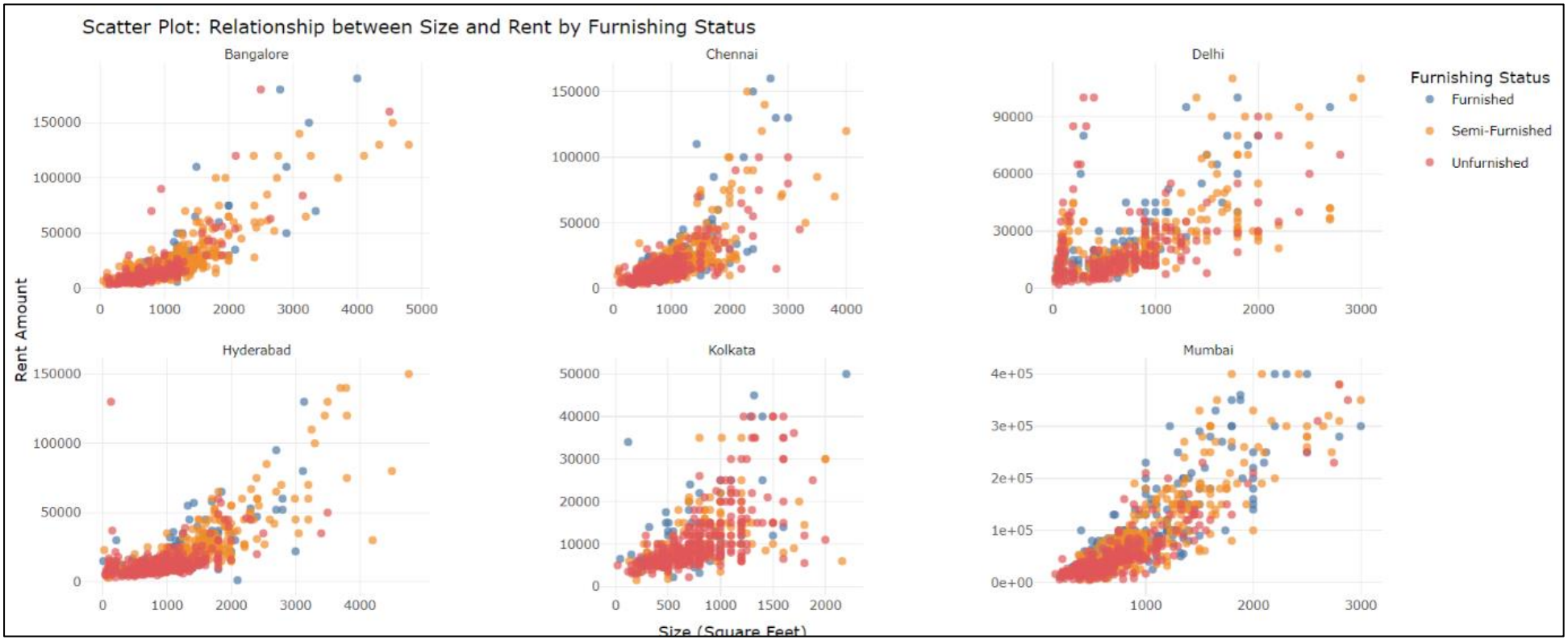
```r
avg_rent_furnishing_type <- HouseRent_Cleaned %>%
  group_by(Furnishing.Status) %>%
  summarise(Avg_Rent = mean(Rent))

# Bar chart to show average rent per furnishing type
avg_rent_furnishing_type_bar <- ggplot(avg_rent_furnishing_type, aes(x = Furnishing.Status, y = Avg_Rent, fill = Furnishing.Status)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = round(Avg_Rent)), vjust = -0.5, size = 3.5) +
  labs(title = "Bar Chart: Average Rent per Furnishing Status", x = "Furnishing Status", y = "Average Rent Amount")

avg_rent_furnishing_type_bar <- ggplotly(avg_rent_furnishing_type_bar)
avg_rent_furnishing_type_bar
```




```r
furnishing_type_boxplot <- ggplot(HouseRent_Cleaned, aes(x = Furnishing.Status, y = Rent, fill = Furnishing.Status)) +
  geom_boxplot() +
  labs(title = "Boxplot: Distribution of Rent by Furnishing Status", x = "Furnishing Status", y = "Rent Amount") +
  theme_minimal() +
  theme(legend.position = "bottom") +
  scale_fill_tableau()

ggplotly(furnishing_type_boxplot)
```



```
furnishing_type_scatter <- ggplot(HouseRent_Cleaned, aes(x = Size, y = Rent, color = Furnishing.Status)) +
  geom_point(alpha = 0.7) +
  labs(title = "Scatter Plot: Relationship between Size and Rent by Furnishing Status",
       x = "Size (Square Feet)", y = "Rent Amount", color = "Furnishing Status") +
  facet_wrap(~ City, scales = "free") +
  theme_minimal() +
  theme(legend.position = "bottom") +
  scale_color_tableau()

ggplotly(furnishing_type_scatter)
```

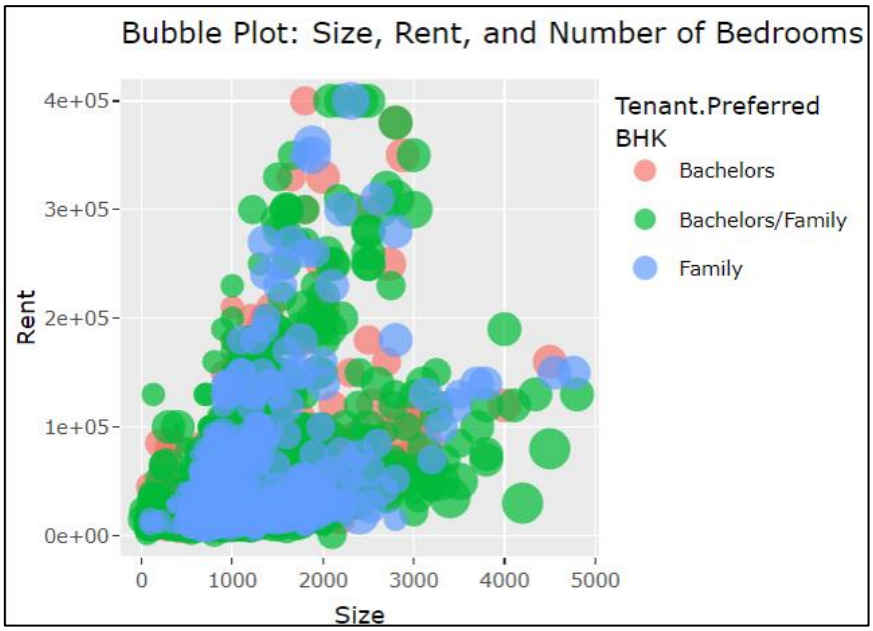


General observations

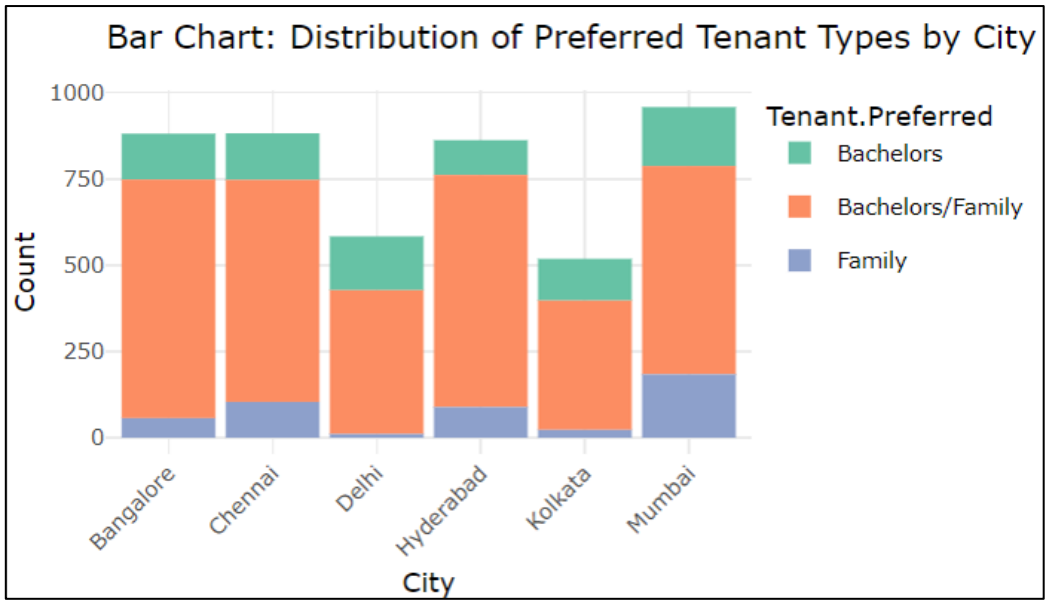
1. **Furnished Properties**: As anticipated, furnished properties (in most cities) tend to command higher rents compared to their semi-furnished or unfurnished counterparts of similar size. The convenience and amenities provided by furnished properties are likely driving this premium.
2. **City Variation**: The influence of furnishing on rent appears to vary by city:
    - In cities like Mumbai, the distinction in rent between furnished and unfurnished properties is pronounced, especially for larger properties.
    - In cities like Kolkata, while there's still a rent premium for furnished properties, the difference seems less stark.
3. **Unfurnished & Semi-Furnished Overlap**: For many cities, there's a significant overlap in the rent ranges of unfurnished and semi-furnished properties. This suggests that, in some markets, the perceived value of semi-furnishing might not be significantly higher than that of an unfurnished property.
4. **City Dynamics**: Cities like Mumbai have a broader spectrum of property sizes and rents compared to other cities. The city-specific markers (different shapes in the scatter plot) help illustrate how city dynamics play a role in the relationship between size, rent, and furnishing.
5. **Density**: The density of data points in specific regions of the plot indicates that most properties (across cities) lie in a particular size and rent range, with fewer properties in the higher size and rent categories.

8.2.5 Question 5: How does the relationship between size, rent, and the preferred tenant type (Tenant Preferred) vary across different cities?

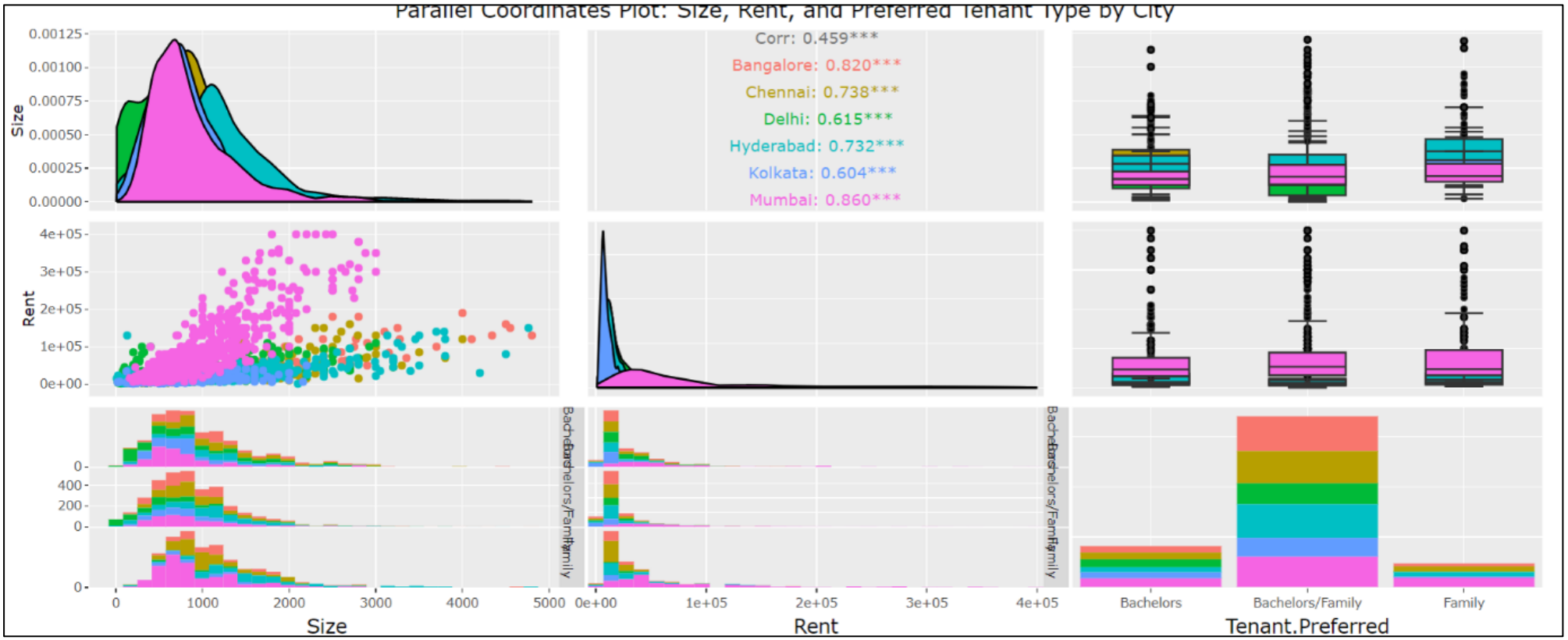
```
bubble_plot_tennant <- ggplot(HouseRent_Cleaned, aes(x = Size, y = Rent, size = BHK, color = Tenant.Preferred)) +
  geom_point(alpha = 0.7) +
  labs(title = "Bubble Plot: Size, Rent, and Number of Bedrooms by Tenant Preferred")

bubble_plot_tennant <- ggplotly(bubble_plot_tennant)
bubble_plot_tennant
```



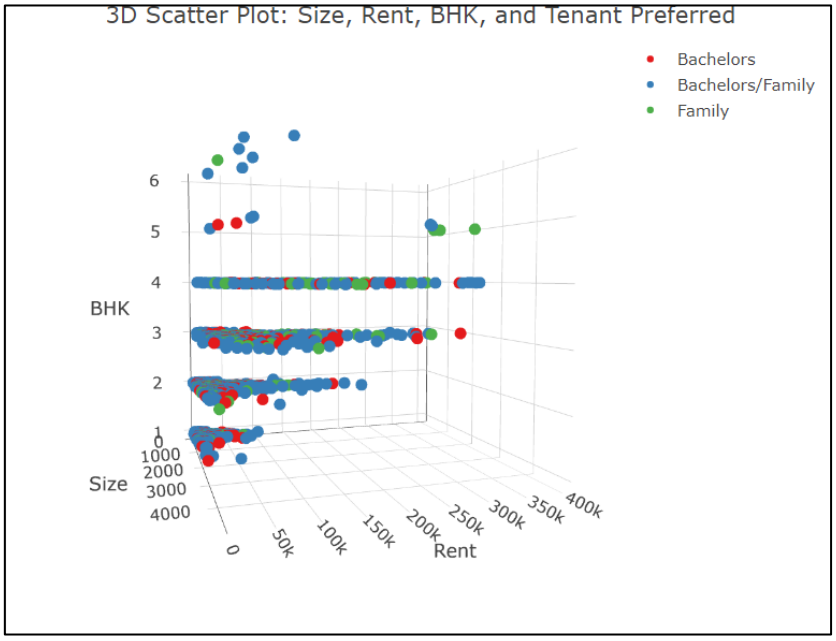

```
tenant_city_bar <- ggplot(HouseRent_Cleaned, aes(x = City, fill = Tenant.Preferred)) +
  geom_bar() +
  labs(title = "Bar Chart: Distribution of Preferred Tenant Types by City",
       x = "City", y = "Count") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_brewer(palette = "Set2")

tenant_city_bar <- ggplotly(tenant_city_bar)
tenant_city_bar
```



```
parallel_coordinates_tenant <- ggpairs(HouseRent_Cleaned,
                                       columns = c("Size", "Rent", "Tenant.Preferred"),
                                       mapping = aes(color = City),
                                       title = "Parallel Coordinates Plot: Size, Rent, and Preferred Tenant Type by City")

# Convert to plotly object
parallel_coordinates_tenant <- ggplotly(parallel_coordinates_tenant)
parallel_coordinates_tenant
```



```
scatter_3d <- plot_ly(data = HouseRent_Cleaned, x = ~Size, y = ~Rent, z = ~BHK,
                      color = ~Tenant.Preferred, colors = "Set1", marker = list(size = 5),
                      text = ~paste("Tenant:", Tenant.Preferred,
                                    "<br>BHK:", BHK,
                                    "<br>Size:", Size,
                                    "<br>Rent:", Rent),
                      type = "scatter3d", mode = "markers")

# Customize layout
scatter_3d <- scatter_3d %>% layout(
  title = "3D Scatter Plot: Size, Rent, BHK, and Tenant Preferred",
  scene = list(
    xaxis = list(title = "Size"),
    yaxis = list(title = "Rent"),
    zaxis = list(title = "BHK")
  )
)

# Display the 3D scatter plot
scatter_3d
```



1. **Tenant Preferences**:
    - Properties preferred for **Bachelors** are generally on the lower end in terms of size and rent, which makes sense as bachelors often seek smaller, more affordable accommodations.
    - Properties open to both **Bachelors and Family** span a broader range in terms of size and rent, indicating a diverse set of properties in this category.
    - Properties preferred for **Family** tend to occupy the mid to higher range, both in terms of size and rent.
2. **City Variation**:
    - The influence of tenant preference on rent appears to vary by city. For instance, in cities like Mumbai, there's a pronounced distinction in rent between properties preferred for families and those for bachelors. However, in cities like Kolkata, the distinction is less stark.
    - Some cities, like Mumbai, have properties spanning a wide spectrum of sizes and rents, while others, like Kolkata, are more clustered in specific regions.

3. **Overlap**: There's a considerable overlap in rent ranges across different tenant preferences, especially in the mid-range sizes. This suggests that while tenant preference plays a role, other factors (like location, condition, amenities) are also significant determinants of rent.
4. **City Dynamics**: The city-specific markers (different shapes in the scatter plot) help illustrate how city dynamics play a role in the relationship between size, rent, and tenant preference. For example, in some cities, properties preferred for families might command a higher rent, while in others, the distinction might be less pronounced.

1. **Dominant Tenant Types**:
    - In cities like **Bangalore, Chennai, Hyderabad, and Kolkata**, many properties are preferred for both "Bachelors and Family", indicating a more versatile rental market in these cities, catering to a wide range of tenants.
    - In **Delhi**, while the preference for "Bachelors and Family" is still dominant, there's a significant portion of properties that are preferred solely for "Family."
    - **Mumbai** stands out with a more even distribution between properties preferred for "Bachelors," "Family," and "Bachelors and Family."
2. **Exclusive Preferences**:
    - The properties preferred exclusively for "Bachelors" are relatively fewer in all cities except Mumbai. This could be due to landlords' preferences or the nature of residential areas in these cities.
    - **Chennai and Kolkata** have minimal properties preferred exclusively for "Bachelors," suggesting a rental market more inclined towards families or flexible tenant arrangements.
3. **Versatility**: Cities with a higher proportion of properties preferred for both "Bachelors and Family" can be seen as having a more flexible rental market, accommodating diverse tenant requirements.

1. **Rent Dynamics**:
    - In general, properties preferred for "Bachelors" tend to have lower median rents compared to those preferred for "Family" or "Bachelors and Family." This is expected, as bachelor accommodations are often smaller and may have fewer amenities.
2. **City Highlights**:
    - **Mumbai**: Demonstrates the highest median rents across all preferred tenant types. Interestingly, properties preferred for "Bachelors" have median rents

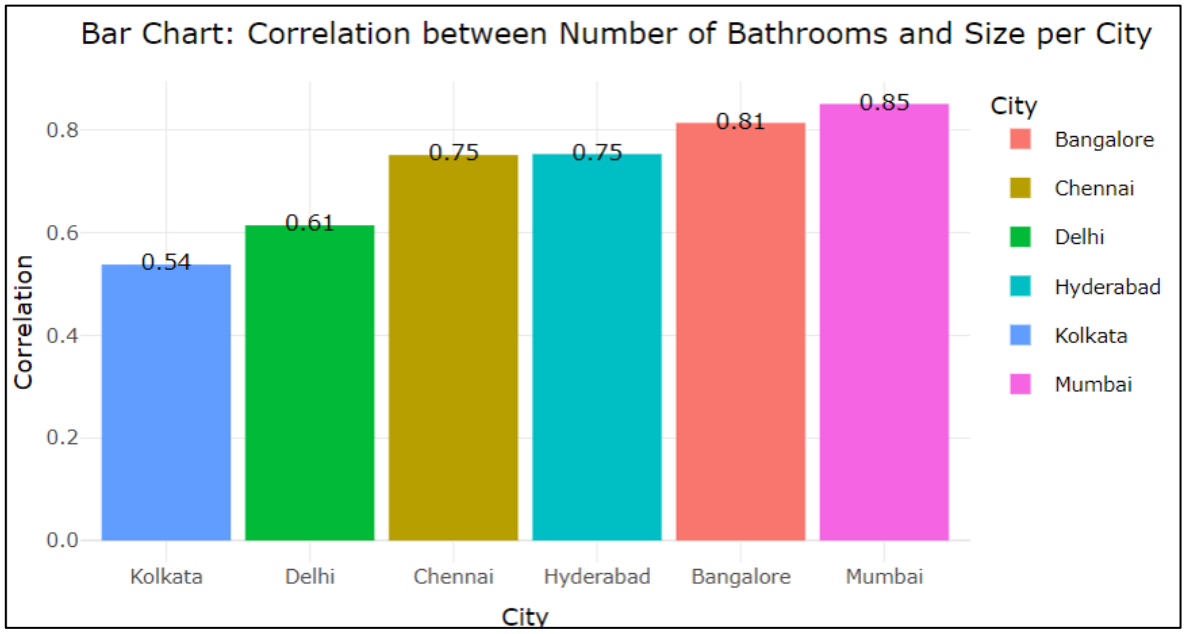
quite close to those preferred for families, indicating the premium nature of the Mumbai rental market.

- **Delhi**: Follows Mumbai with high median rents, especially for properties preferred for "Family."
- **Bangalore, Hyderabad, Chennai, and Kolkata**: Have more moderate median rents, with properties preferred for "Family" generally commanding higher rents than those for "Bachelors."

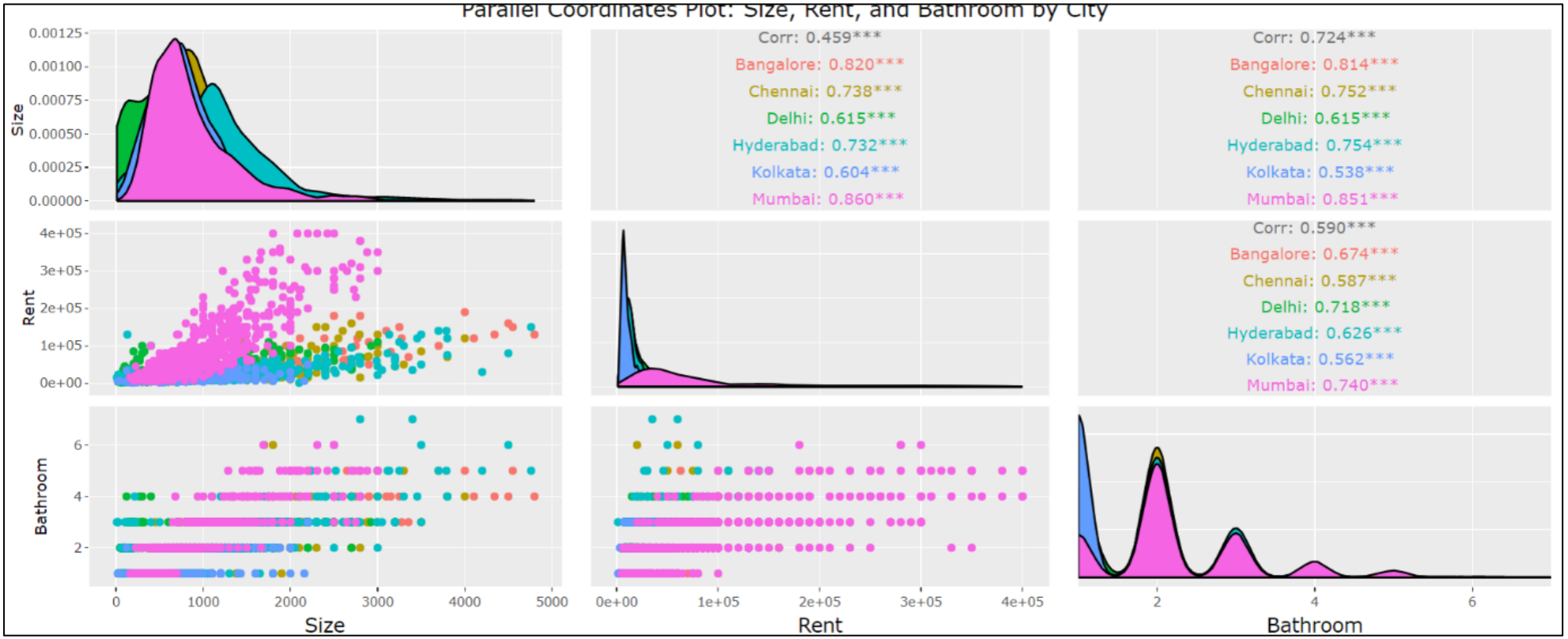
3. **Versatility in Rent**: Cities like Bangalore and Hyderabad show minimal differences in median rents between properties preferred for "Bachelors" and those for "Bachelors and Family." This could reflect the versatility of such properties, catering to a wider range of tenants without significant rent differences.

8.2.6 Question 6: How do the relationships between size, rent, and number of bathrooms vary across different cities?

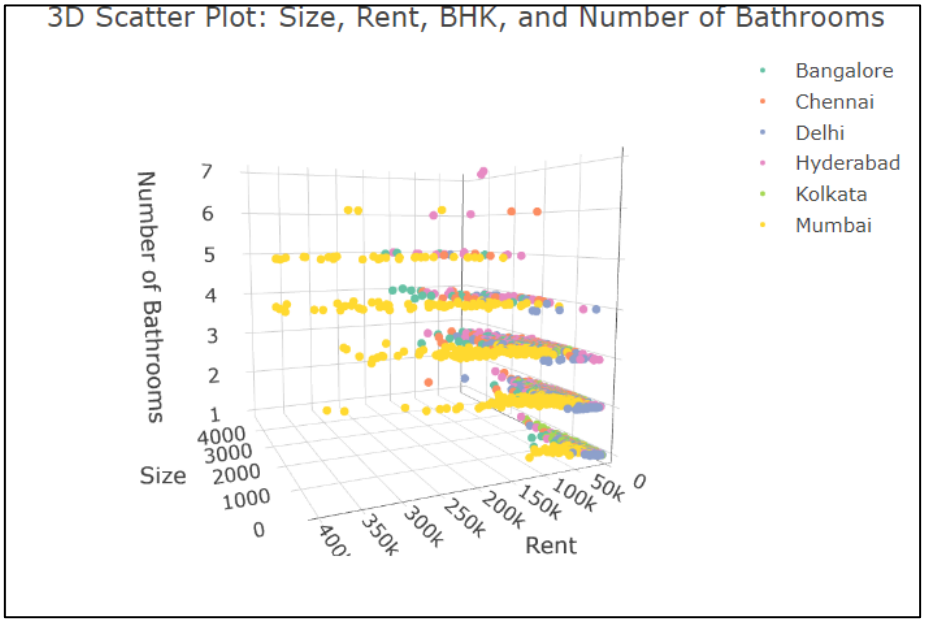
```
correlation_bath_size <- cor(HouseRent_Cleaned$Bathroom, HouseRent_Cleaned$Size)

# Display the correlation
cat("Correlation between Number of Bathrooms and Size:", correlation_bath_size)

# Calculate the correlation between number of bathrooms and size for each city using dplyr
correlation_bath_size_city <- HouseRent_Cleaned %>%
  group_by(City) %>%
  summarise(Correlation = cor(Bathroom, Size))

# Create a bar chart to visualize the correlation between number of bathrooms and size per city
correlation_bath_size_city_bar <- ggplot(correlation_bath_size_city, aes(x = reorder(City, Correlation), y = Correlation, fill = City)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = round(Correlation, 2)), vjust = -0.5, size = 3.5) +
  labs(title = "Bar Chart: Correlation between Number of Bathrooms and Size per City",
       x = "City", y = "Correlation") +
  theme_minimal()

correlation_bath_size_city_bar <- ggplotly(correlation_bath_size_city_bar)
correlation_bath_size_city_bar
```



```
parallel_coordinates_bathroom <- ggpairs(HouseRent_Cleaned,
                                         columns = c("Size", "Rent", "Bathroom"),
                                         mapping = aes(color = City),
                                         title = "Parallel Coordinates Plot: Size, Rent, and Bathroom by City")

# Convert to plotly object
parallel_coordinates_bathroom <- ggplotly(parallel_coordinates_bathroom )
parallel_coordinates_bathroom
```



```
scatter_3d_bathrooms <- plot_ly(HouseRent_Cleaned, x = ~Size, y = ~Rent, z = ~Bathroom,
                                color = ~City, marker = list(size = 3)) %>%
  add_markers() %>%
  layout(scene = list(xaxis = list(title = "Size"),
                      yaxis = list(title = "Rent"),
                      zaxis = list(title = "Number of Bathrooms")),
         title = "3D Scatter Plot: Size, Rent, BHK, and Number of Bathrooms")

# Display the 3D scatter plot with area type
scatter_3d_bathrooms
```



1. **General Trend**: In most cities, as the property size increases, the rent tends to increase as well. This is expected as larger properties usually command higher rents.

2. **Strong Correlation in Mumbai and Bangalore**: Both Mumbai and Bangalore display strong positive correlations, indicating that in these cities, as properties increase in size, there's a high likelihood they'll also have more bathrooms.

3. **Moderate to Strong Correlations**: Chennai, Delhi, and Hyderabad show moderate to strong correlations, suggesting a notable relationship between property size and the number of bathrooms in these cities.

4. **Weakest Correlation in Kolkata**: Kolkata has the weakest correlation among the cities listed, though it's still positive. This suggests that while there's a relationship between property size and the number of bathrooms, other factors might play a more significant role in determining the number of bathrooms in properties in Kolkata.

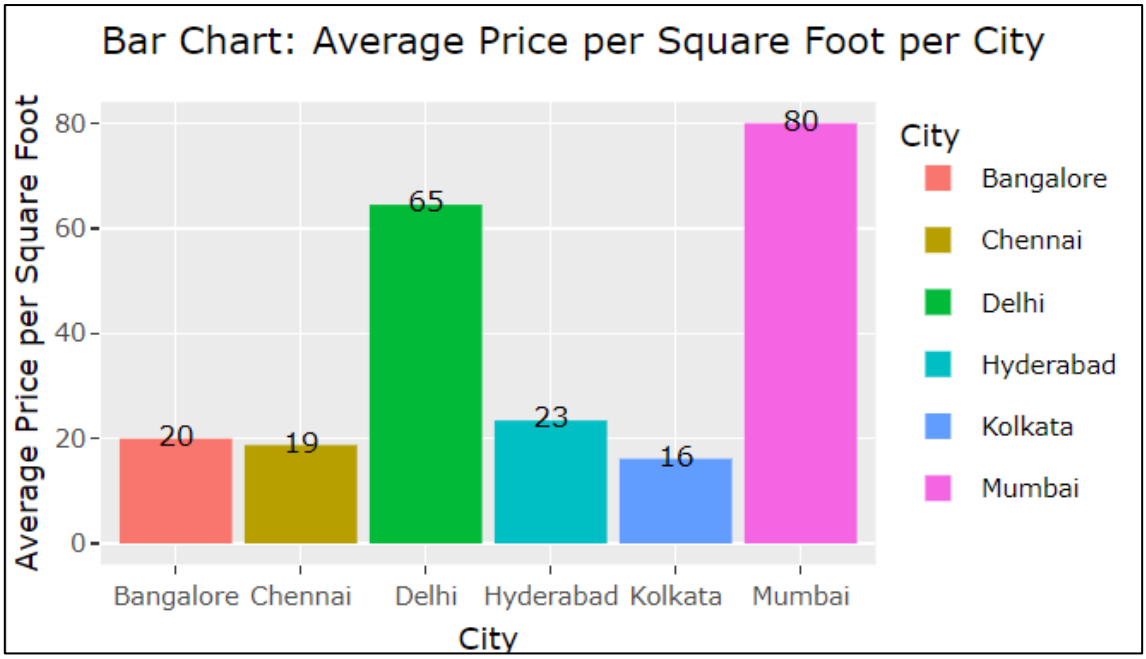
5. **Number of Bathrooms**: The number of bathrooms seems to play a significant role in rent, especially for properties of the same size. Properties with more bathrooms often

command a higher rent than those with fewer bathrooms, even if they are of similar size.

6. **City Differences**: The relationship varies across cities. Some cities like Kolkata and Mumbai show a dense concentration of points in particular regions, indicating higher competition and demand in those size and rent ranges.
7. **Outliers**: From the 3D scatter plot, there are a few outliers, especially in cities like Mumbai, where some large-sized properties command exceptionally high rents, possibly due to their prime locations or other amenities not captured in this datas

8.2.8 Question 8: What is the average price per square foot for each city?

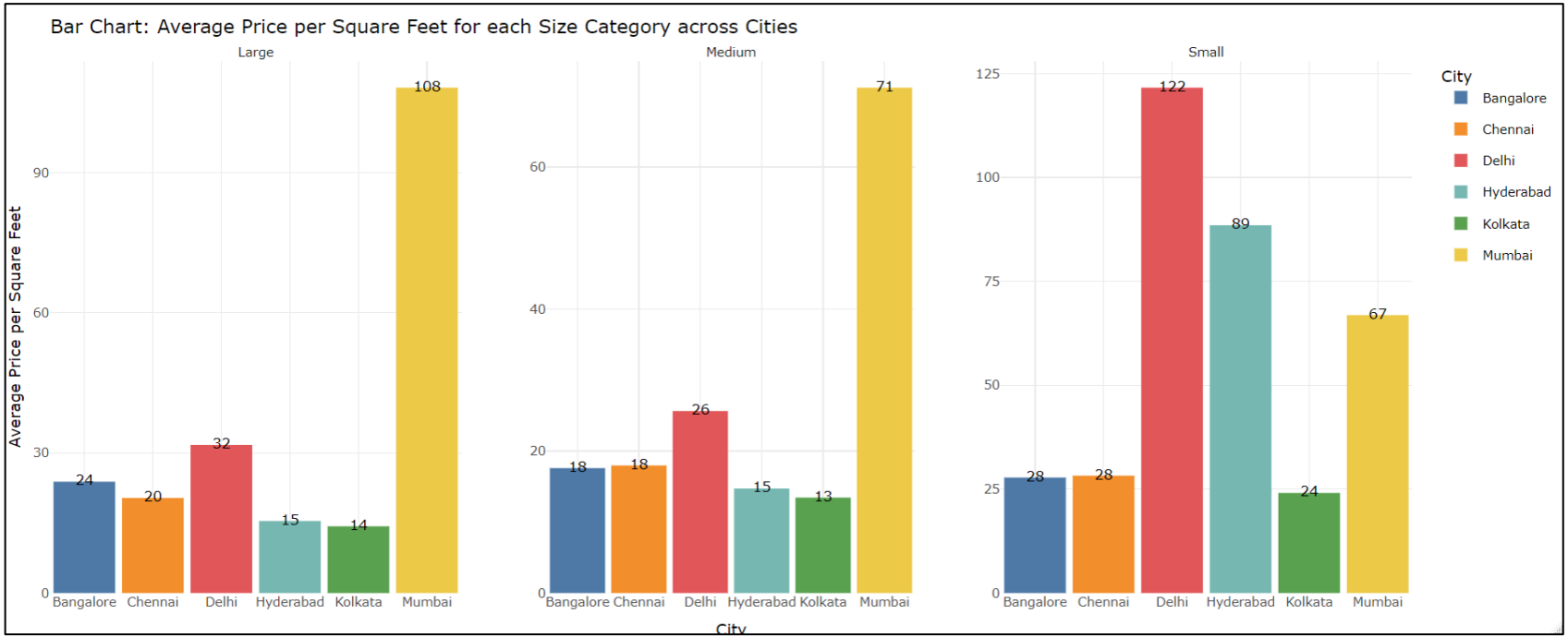
```
avg_price_per_square_feet_city <- HouseRent_Cleaned %>%
  group_by(City) %>%
  summarise(Avg_PricePerSquareFeet = mean(PricePerSquareFeet))

# Plot average price per square feet per city in a bar chart with labels on the bars
avg_price_per_square_feet_city_bar <- ggplot(avg_price_per_square_feet_city, aes(x = City, y = Avg_PricePerSquareFeet, fill = City)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = round(Avg_PricePerSquareFeet)), vjust = -0.5, size = 3.5) +
  labs(title = "Bar Chart: Average Price per Square Feet per City", x = "City", y = "Average Price per Square Feet")

avg_price_per_square_feet_city_bar <- ggplotly(avg_price_per_square_feet_city_bar)
avg_price_per_square_feet_city_bar
```



- **Mumbai** has the highest average price per square foot most likely due to the high demand for accommodation in the city.
- **Delhi** follows Mumbai with the second-highest average price, reflecting the city's status as the capital and its high demand for rental properties.
- **Hyderabad, Bangalore, and Chennai** have moderate rental prices per square foot, with Hyderabad slightly leading the pack. Ongoing in these cities urbanization might play roles in setting these prices.
- **Kolkata** offers the most affordable average rental price per square foot among the cities listed. This might be influenced by the city's unique market dynamics and cultural factors.

8.2.9 Question 9: How does the relationship between size categories (small, medium, large), rent, and price per square feet vary across different cities?

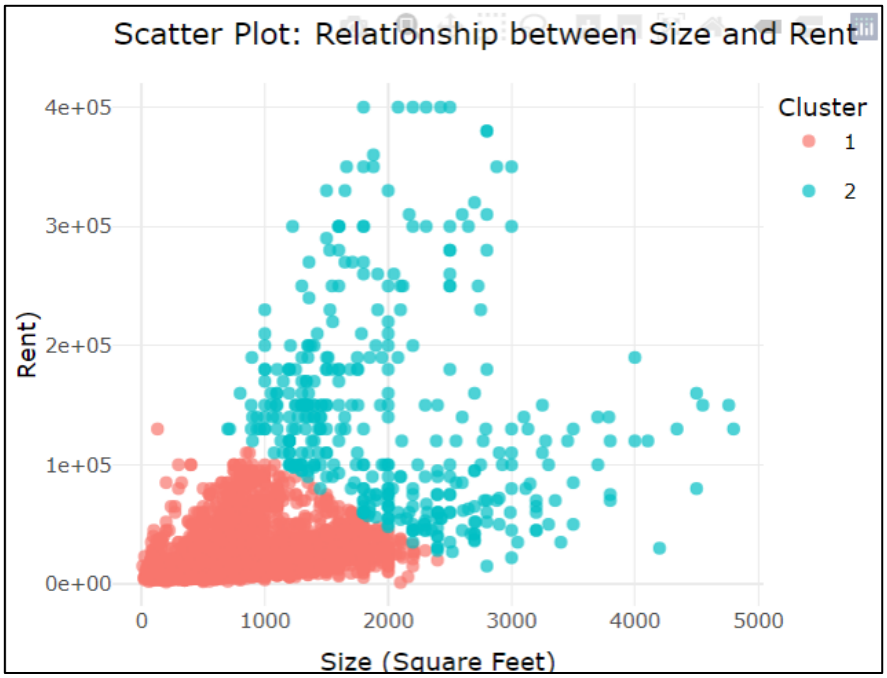
```
# Create size categories (small, medium, large) based on square footage
HouseRent_Cleaned$Size_Category <- ifelse(HouseRent_Cleaned$Size < 500, "Small",
                                          ifelse(HouseRent_Cleaned$Size >= 500 & HouseRent_Cleaned$Size < 1000, "Medium", "Large"))

avg_price_per_square_feet_size_category_city <- HouseRent_Cleaned %>%
  group_by(Size_Category, City) %>%
  summarise(Avg_PricePerSquareFeet = mean(PricePerSquareFeet))

# Plot the average price per square foot for each size category across cities in a bar chart with labels on the bars
avg_price_per_square_feet_size_category_city_bar <- ggplot(avg_price_per_square_feet_size_category_city, aes(x = City, y = Avg_PricePerSquareFeet, fill = City)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = round(Avg_PricePerSquareFeet)), vjust = -0.5, size = 3.5) +
  labs(title = "Bar Chart: Average Price per Square Feet for each Size Category across Cities", x = "City", y = "Average Price per Square Feet") +
  facet_wrap(~ Size_Category, scales = "free") +
  theme_minimal() +
  theme(legend.position = "bottom") +
  scale_fill_tableau()

avg_price_per_square_feet_size_category_city_bar <- ggplotly(avg_price_per_square_feet_size_category_city_bar)
avg_price_per_square_feet_size_category_city_bar
```



1. **Size and Price Dynamics**: From the bar chart, in several cities (e.g., Mumbai, Delhi, and Kolkata), smaller properties tend to have a higher price per square foot compared to larger properties. Therefore, even though larger properties command higher overall rents, smaller properties might be more expensive when considering the per square foot metric.

2. **Economic Insight**: The higher price per square foot for smaller properties can be influenced by factors like location premiums (smaller properties in prime locations), the inclusion of more amenities, or market demand for compact housing.

```
# Cluster the data based on size and rent
HouseRent_Cleaned_Cluster <- HouseRent_Cleaned %>%
  select(Size, Rent)

# Scale the data
HouseRent_Cleaned_Cluster_Scaled <- scale(HouseRent_Cleaned_Cluster)

# Cluster the data into 3 clusters
HouseRent_Cleaned_Cluster_KMeans <- kmeans(HouseRent_Cleaned_Cluster_Scaled, 2, nstart = 10, iter.max = 1000)

# Add the cluster labels to the data
HouseRent_Cleaned_Cluster <- HouseRent_Cleaned_Cluster %>%
  mutate(Cluster = as.factor(HouseRent_Cleaned_Cluster_KMeans$cluster))

# Scatter plot to visualize the relationship between size and rent
size_rent_scatter <- ggplot(HouseRent_Cleaned_Cluster, aes(x = Size, y = Rent, color = Cluster)) +
  geom_point(alpha = 0.7) +
  labs(title = "Scatter Plot: Relationship between Size and Rent",
       x = "Size (Square Feet)", y = "Rent)") +
  theme_minimal() +
  theme(legend.position = "bottom") +
  scale_color_discrete()

size_rent_scatter <- ggplotly(size_rent_scatter)
size_rent_scatter
```



The Dynamics of Clusters:

1. Cluster 1 (Orange): Properties in this cluster are smaller in size and have cheaper rentals. This could include studio apartments, one-bedroom apartments, or other small living spaces in less desirable places.

2. Cluster 2 (Sky Blue): Attracts properties that are medium to large in size yet have low rents. These could be properties in cities or suburbs with reduced rental rates.

There are properties in the second cluster that are larger and attract greater rentals. These could be luxury apartments, villas, or properties in prime city locations, and based on this clustering, they might be considered outliers. Further analysis could indicate whether they are actual outliers or a subset of the market.

Cluster Spread: The clusters depict the overall pattern in the data, which shows that rent rises with size. However, the variation within each cluster suggests that other factors influence rent as well.

8.2.10 How important is the size feature to determine rent in each city?

Anova test

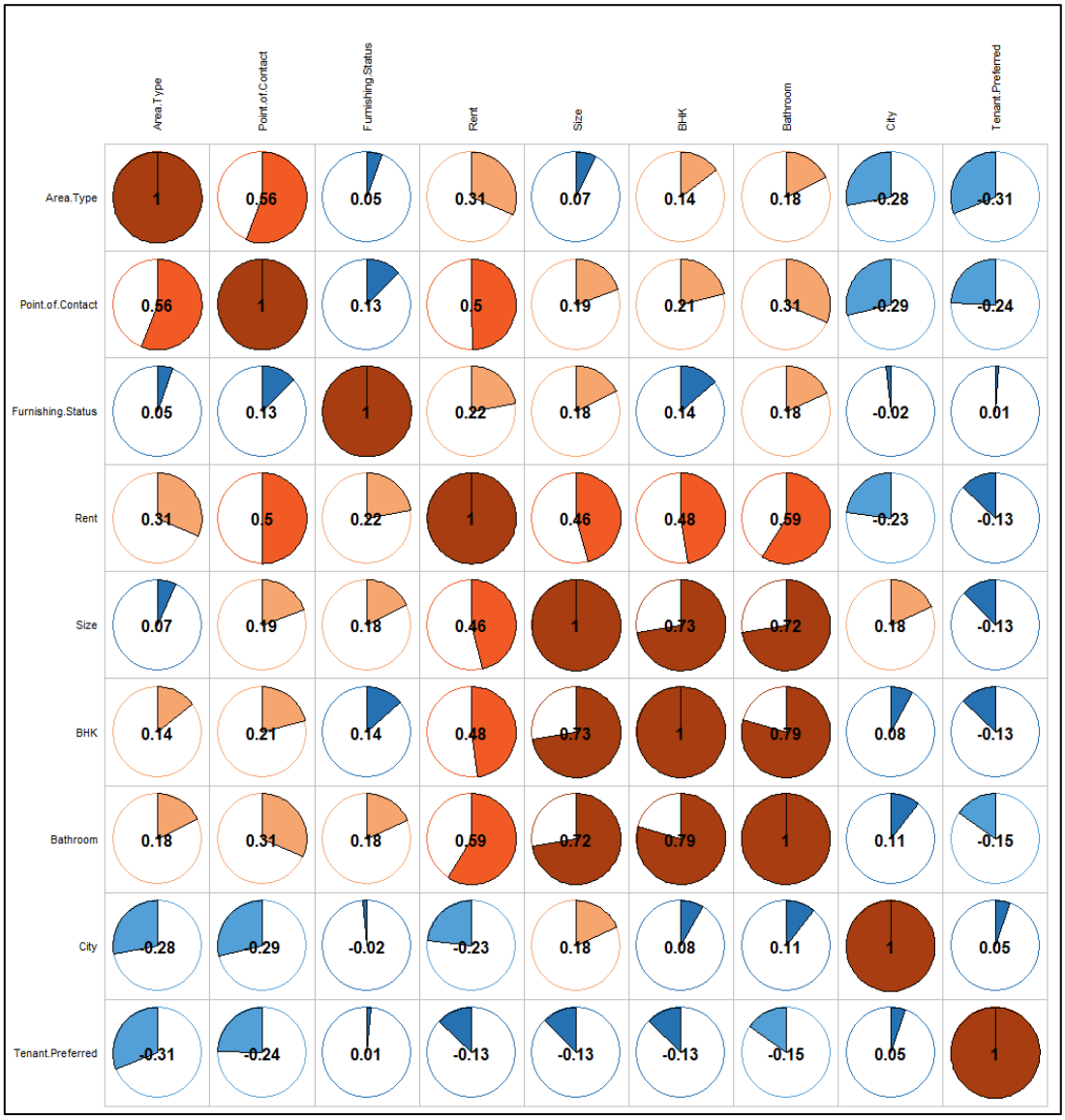
```
#Plotting anova results for each city
# Create a list of dataframes for each city
city_subsets <- split(HouseRent_Cleaned, HouseRent_Cleaned$City)

# Create a list of anova results for each city
city_anova <- lapply(city_subsets, function(x) {
  aov(Rent ~ Size, data = x)
})

# Create a list of anova summary for each city
city_anova_summary <- lapply(city_anova, summary)
city_anova_summary
```

```
$Bangalore
             Df    Sum Sq   Mean Sq F value Pr(>F)
Size          1 2.807e+11 2.807e+11    1809 <2e-16 ***
Residuals   880 1.366e+11 1.552e+08
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

$Chennai
             Df    Sum Sq   Mean Sq F value Pr(>F)
Size          1 1.694e+11 1.694e+11    1054 <2e-16 ***
Residuals   881 1.416e+11 1.608e+08
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

$Delhi
             Df    Sum Sq   Mean Sq F value Pr(>F)
Size          1 8.352e+10 8.352e+10   354.9 <2e-16 ***
Residuals   582 1.370e+11 2.353e+08
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

$Hyderabad
             Df    Sum Sq   Mean Sq F value Pr(>F)
Size          1 1.421e+11 1.421e+11   995.3 <2e-16 ***
Residuals   861 1.229e+11 1.428e+08
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

$Kolkata
             Df    Sum Sq   Mean Sq F value Pr(>F)
Size          1 1.095e+10 1.095e+10   296.9 <2e-16 ***
Residuals   517 1.906e+10 3.688e+07
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

$Mumbai
             Df    Sum Sq   Mean Sq F value Pr(>F)
Size          1 3.880e+12 3.880e+12    2719 <2e-16 ***
Residuals   957 1.366e+12 1.427e+09
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

From the values above, all cities have very high F values and very low p-values, which indicate that size has a very strong and significant effect on rent. Thus, as the size of the house increases, so does the rent, which is logical and expected.

Correlation Heatmap of selected features

```
# Update the "City" column using label encoding
HouseRent_Cleaned_Preprocessed <- HouseRent_Cleaned_Preprocessed %>%
  mutate(City = case_when(
    City == "Kolkata" ~ 0,
    City == "Mumbai" ~ 1,
    City == "Bangalore" ~ 2,
    City == "Delhi" ~ 3,
    City == "Chennai" ~ 4,
    City == "Hyderabad" ~ 5
    # Add more city mappings if needed
  ))
columns <- c("BHK", "Rent", "Size", "Area.Type", "City",
             "Furnishing.Status", "Tenant.Preferred", "Bathroom", "Point.of.Contact")

# Extract the selected columns from the dataset
selected_data <- HouseRent_Cleaned_Preprocessed[columns]

# Compute the correlation matrix
correlation_matrix <- cor(selected_data)

colour_palette <- c("white","#9FD0F9","#53A0D8","#2671B2","#F7A56F", "#F15A24", "#A73D10")

# Create a correlation plot using corrplot
corrplot(correlation_matrix, method = "pie", tl.col = "black", tl.cex = 0.7, col = colour_palette,
         addCoef.col = "black", # Color of correlation coefficient text
         order = "hclust",      # Order rows and columns using hierarchical clustering
         cl.pos = "n")          # Position of cluster labels (none)
```



In the correlation heatmap above, each cell contains a pie chart representing the correlation between the respective row and column variables.

Larger pie sections indicate stronger correlations. Therefore, **Bathroom**, **Point.of.Contact**, **Bedroom**, and **Size** are the strongest positive correlations, while **City** is the strongest negative correlation.

Conclusion of Models (see Building and Evaluating Predictive Models in appendix).

- In terms of pure predictive performance, the **Random Forest** outperforms the other models with an $R2$ score of 98.59%, indicating it captures most of the variance in the data. Its MAE and RMSE also provide insight into its predictive accuracy, with the MAE suggesting relatively small average errors, but the RMSE indicating some larger errors in specific instances.
- **Decision Trees** offer a balance between performance and interpretability, but their performance was notably less than the Random Forest in this case.
- **Linear Regression** offers high interpretability, but the residuals plot suggests that there might be non-linear relationships or interactions in the data that the linear model isn't capturing.

8.2.11 Conclusion of Objective

The research of house rent data shows some intriguing insights into the relationship between property size and rent in various Indian towns. In general, there is a positive link between property size and rent, which means that larger properties tend to have higher rentals. This relationship, however, varies by city, with Mumbai having the largest relation and Bangalore having the weakest. This implies that city-specific factors such as economic situations, preferences, and the availability of resources all have a part in deciding rent levels. Furthermore, the data demonstrates that tenant preferences and furnishing choices influence the relationship between size and rent, with properties preferred for families commanding higher rents. Various predictive models are also used in the analysis to estimate rent based on different features, with Random Forest showing the best performance. The results of this analysis can help property owners and renters make informed decisions about their housing options.

8.2.12 Additional Features

1. Scatter plots with regression lines for size vs. rent across different cities.
2. Box plots to visualise the distribution of rent per city.

3. Calculation and visualisation of average rent per city using bar charts.
4. Scatter plots with regression lines for size vs. rent across different BHKs.
5. Calculation and visualisation of average rent per BHK using bar charts.
6. Scatter plots to show the relationship between size and rent by different area types.
7. Rotating the x-axis ticks using **axis.text.x** from ggthemes library
8. Box plots to visualise the distribution of rent by different area types.
9. Calculation and visualisation of average rent per area type using bar charts.
10. Creation of a 3D pie chart to show the average size per area type.
11. Scatter plots with facet wrap to show the relationship between size and rent by different furnishing statuses.
12. Calculation and visualisation of average rent per furnishing status using bar charts.
13. Creation of a 3D pie chart to show the average size per furnishing status.
14. Scatter plots to show the relationship between size and rent by the number of bedrooms.
15. Creation of parallel coordinates plots to visualise relationships between size, rent, and other variables.
16. 3D scatter plot to show relationships between size, rent, and BHK with different colors based on tenant preference.
17. Bubble plots for visualising relationships between size, rent, and other variables for each city.
18. Violin plot to show the relationship between size and rent based on the number of bedrooms.
19. Correlation analysis between the number of bathrooms and the size of the house.
20. Bar plots to visualise the distribution of preferred tenant types across different cities.
21. Correlation analysis between the size of the house and rent, both overall and per city.
22. Visualisation of average price per square foot for each city using bar charts.

23. Analysis of the relationship between size categories (small, medium, large), rent, and price per square foot across different cities.
24. Cluster analysis based on size and rent, with a scatter plot for visualisation.

A. **Linear Regression Model**

- Trained the model and generated a summary.
- Plotted a residual plot and evaluated the model using metrics such as RMSE, and R-squared).

B. **Random Forest Model**:

- Built a random forest model using the **randomForest** library.
- Made predictions and calculated residuals.
- Plotted a residual plot and evaluated the model using metrics (MAE, MSE, RMSE, R-squared).
- Visualised feature importance using Plotly, as well as actual vs predicted values.

C. **Decision Tree Model**:

- Created a decision tree model using the **rpart** and **rpart.plot** libraries.
- Plotted the decision tree and made predictions.
- Evaluated the model using metrics (MAE, MSE, RMSE, R-squared).
- Visualised actual vs. predicted values.

D. **ANOVA Testing and Residual Analysis**:

- Performed ANOVA testing using the **aov** function, including for each city.
- Summarized ANOVA results and extracted F-statistics and p-values.
- Created a histogram and QQ-plot to analyse residuals.

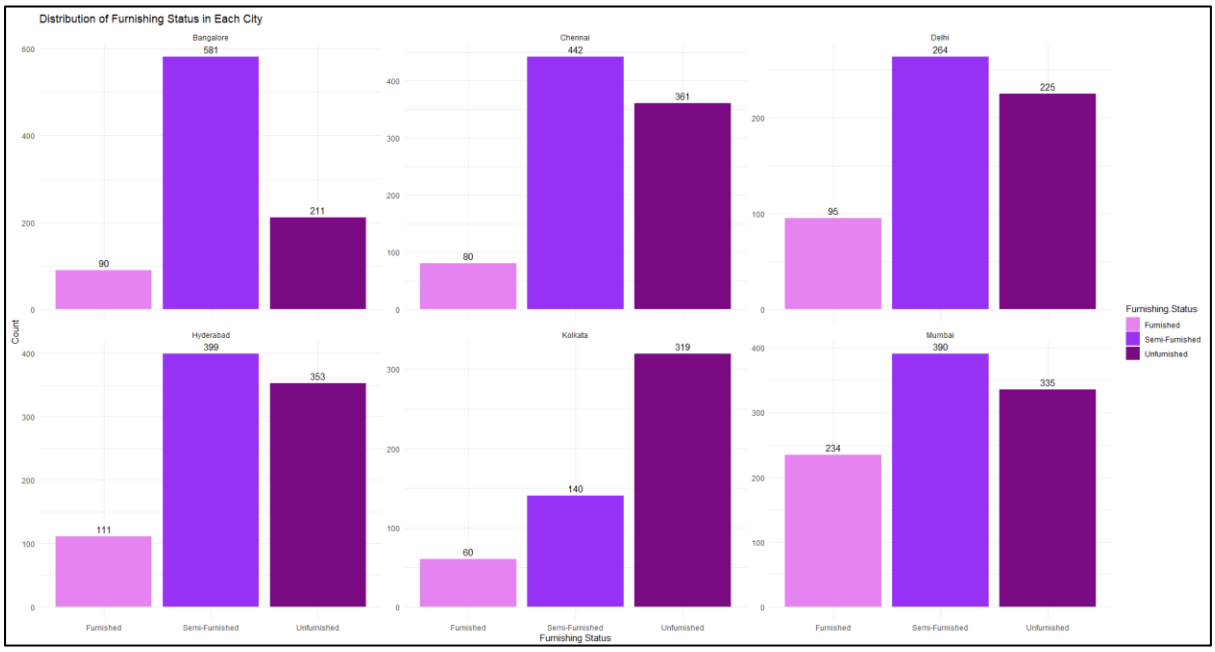
E. **Label Encoding and Correlation Analysis**:

- Encoded categorical "City" values using label encoding.
- Computed and visualised a correlation matrix using **corrplot**.

8.3 Objective 3: Analyse how the furnishing status affects the rent amount across different cities. (WONG SHI WEI TP063736)

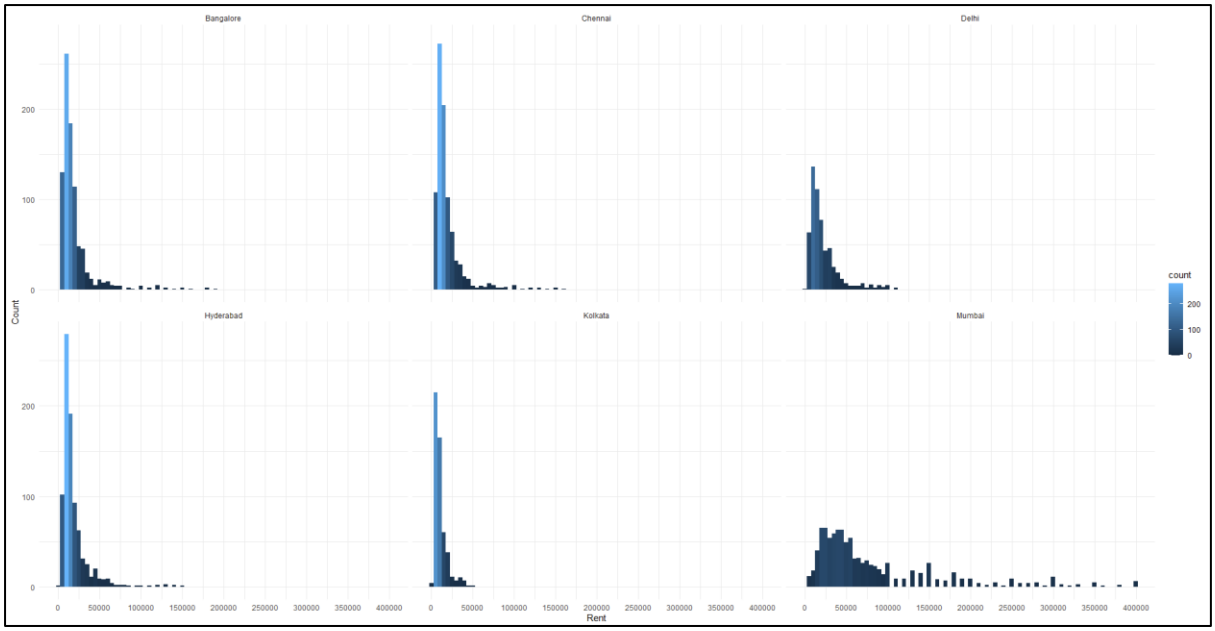
8.3.1 Question 1: How the distribution of Furnishing Status and Rent in each city?

```
ggplot(HouseRent_Cleaned, aes(x = Furnishing.Status)) +
  geom_bar(aes(fill = Furnishing.Status)) +
  scale_fill_manual(values = c("violet", "purple", "#800080")) +
  geom_text(stat='count', aes(label=..count..), vjust=-0.5) +
  facet_wrap(~City) +
  labs(title = "Distribution of Furnishing Status in Each City",
       x = "Furnishing Status",
       y = "Count") +
  theme_minimal()
```



According to the distribution of 'Furnishing Status' in each city, we can clearly see that except the city 'Kolkata', people would rather to rent the house with unfurnished status, while other cities most people will choose to rent the house with semi-furnished status following with unfurnished status and the people who choose the house with furnished status is the least.

```
ggplot(HouseRent_Cleaned, aes(x = Rent)) +
  geom_histogram(binwidth = 5000, aes(fill = ..count..)) +
  facet_wrap(~City) +
  labs(title = "Distribution of Rent in Each City",
       x = "Rent",
       y = "Count") +
  scale_x_continuous(breaks = seq(0, max(HouseRent_Cleaned$Rent), by = 50000)) +
  theme_minimal()
```



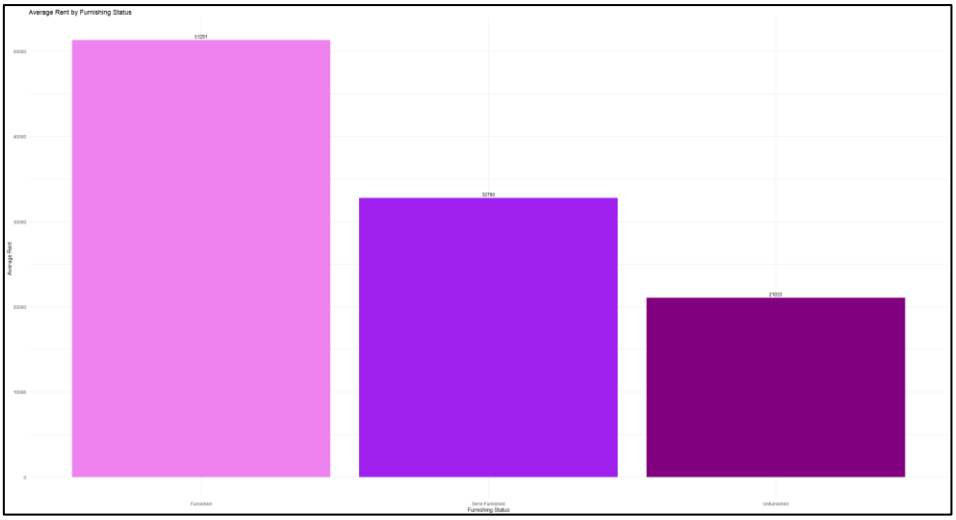
Except for Mumbai City, the histogram of rent in each city shows that rent in most cities is less than 50000. Only in Mumbai City is the rent price distribution more symmetrical, resulting in higher rent price frequencies.

8.3.2 Question 2: What is the average rent of the furnishing status across all the cities.

```
# Average rent for each furnishing status across all cities
HouseRent_Cleaned %>%
  group_by(Furnishing.Status) %>%
  summarise(Average_Rent = mean(Rent, na.rm = TRUE))
```

	Furnishing.Status	Average_Rent
	<chr>	<dbl>
1	Furnished	51291.
2	Semi-Furnished	32760.
3	Unfurnished	21033.

```
# Bar chart
ggplot(avg_rent_by_status, aes(x = Furnishing.Status, y = Average_Rent, fill = Furnishing.Status)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_fill_manual(values = c("violet", "purple", "#800080")) +
  geom_text(aes(label = sprintf("%.0f", Average_Rent)), vjust = -0.5, size = 3.5) +
  labs(title = "Average Rent by Furnishing Status",
       x = "Furnishing Status",
       y = "Average Rent") +
  theme_minimal() +
  theme(legend.position = "none")
```



According to the picture above, it shows that houses with ‘Furnished’ status have the highest average rent, which is 51291, while houses with ‘Semi-Furnished’ status have the average of 32760 and ‘Unfurnished’ status houses has the lowest average of rent which is 21033.

8.3.3 Question 3: Which city has the highest average rent for each furnished property?

```
HouseRent_Cleaned %>%
  filter(Furnishing.Status == "Furnished") %>%
  group_by(City) %>%
  summarise(Average_Rent = mean(Rent, na.rm = TRUE)) %>%
  arrange(desc(Average_Rent)) %>%
  head(1)
```

City	Average_Rent
<chr>	<dbl>
1 Mumbai	98995.

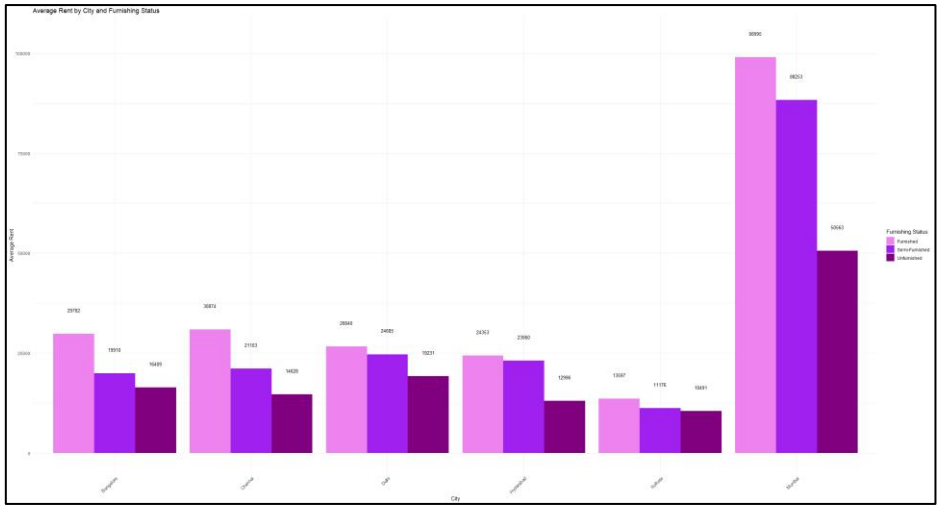
```
HouseRent_Cleaned %>%
  filter(Furnishing.Status == "Semi-Furnished") %>%
  group_by(City) %>%
  summarise(Average_Rent = mean(Rent, na.rm = TRUE)) %>%
  arrange(desc(Average_Rent)) %>%
  head(1)
```

City	Average_Rent
<chr>	<dbl>
1 Mumbai	88253.

```
HouseRent_Cleaned %>%
  filter(Furnishing.Status == "Unfurnished") %>%
  group_by(City) %>%
  summarise(Average_Rent = mean(Rent, na.rm = TRUE)) %>%
  arrange(desc(Average_Rent)) %>%
  head(1)
```

City	Average_Rent
<chr>	<dbl>
1 Mumbai	50563.

```
avg_rent_by_city_and_fstatus <- HouseRent_Cleaned %>%
  group_by(City, Furnishing.Status) %>%
  summarise(Average_Rent = mean(Rent, na.rm = TRUE))

# Bar chart
ggplot(avg_rent_by_city_and_fstatus, aes(x = City, y = Average_Rent, fill = Furnishing.Status)) +
  geom_bar(stat="identity", position="dodge") +
  scale_fill_manual(values = c("Violet", "purple", "#80008D")) +
  geom_text(aes(label=sprintf("%.0f", Average_Rent), y = Average_Rent + max(Average_Rent)*0.05),
            position = position_dodge(width=0.9), vjust=-0.5, size=3.5) +
  labs(title = "Average Rent by City and Furnishing Status",
       x = "City",
       y = "Average Rent") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



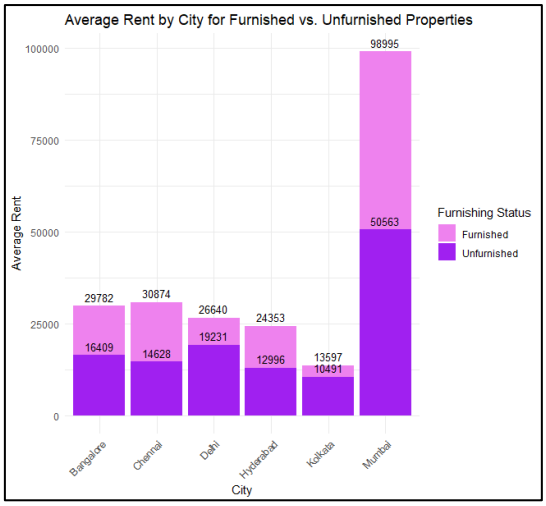
The city ‘Mumbai’ has the highest average rent of each furnished status across all the cities with the average rent of 98995 in ‘Furnished’ house, 88253 in ‘Semi-Furnished’ house and 50563 in ‘Unfurnished’ house.

8.3.4 Question 4: Is there a significant difference in the average rent between furnished and unfurnished properties in each city?

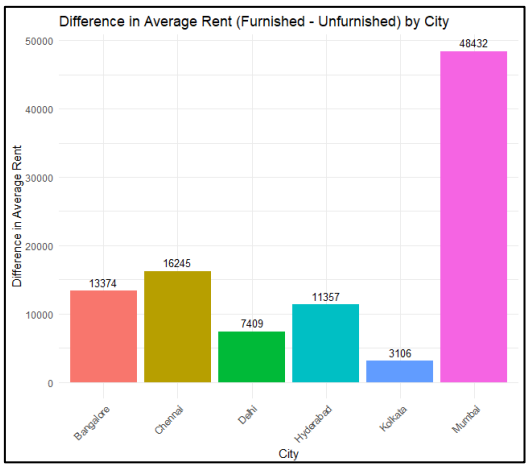
```
library(tidyr)
avg_rent_difference <- HouseRent_Cleaned %>%
  filter(Furnishing.Status %in% c("Furnished", "Unfurnished")) %>%
  group_by(City, Furnishing.Status) %>%
  summarise(Average_Rent = mean(Rent, na.rm = TRUE)) %>%
  spread(Furnishing.Status, Average_Rent) %>%
  mutate(Difference = Furnished - Unfurnished)
```

	City	Furnished	Unfurnished	Difference
	<chr>	<dbl>	<dbl>	<dbl>
1	Bangalore	29782.	16409.	13374.
2	Chennai	30874.	14628.	16245.
3	Delhi	26640.	19231.	7409.
4	Hyderabad	24353.	12996.	11357.
5	Kolkata	13597.	10491.	3106.
6	Mumbai	98995.	50563.	48432.

```
# Bar chart for average rents
ggplot(avg_rent_difference, aes(x = City)) +
  geom_bar(aes(y = Furnished, fill = "Furnished"), stat="identity", position="dodge") +
  geom_bar(aes(y = Unfurnished, fill = "Unfurnished"), stat="identity", position="dodge") +
  labs(title = "Average Rent by City for Furnished vs. Unfurnished Properties",
       x = "City",
       y = "Average Rent") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_manual(values = c("Furnished" = "violet", "Unfurnished" = "purple"), name="Furnishing Status")
```




```
# Bar chart for difference in rents
ggplot(avg_rent_difference, aes(x = City, y = Difference, fill = City)) +
  geom_bar(stat="identity") +
  geom_text(aes(label=sprintf("%.0f", Difference)), vjust=-0.5, size=3.5) +
  labs(title = "Difference in Average Rent (Furnished - Unfurnished) by City",
       x = "City",
       y = "Difference in Average Rent") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1), legend.position = "none")
```



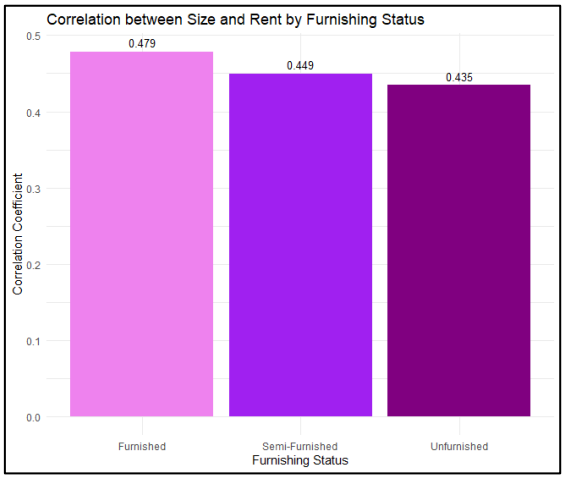
We can find out that 'Mumbai' city has the most difference in the average rent between 'Furnished' and 'Unfurnished' status in each city which is up to 48432, while the city 'Delhi' and 'Kolkata' has the lowest difference of average rent between 'Furnished' and 'Unfurnished' status among other cities which is 7409 and 3106.

8.3.5 Question 5: How does the size of the property correlate with rent differences among furnishing statuses?

```
correlation_data <- HouseRent_Cleaned %>%
  group_by(Furnishing.Status) %>%
  summarise(Correlation = cor(Size, Rent, use = "complete.obs"))
```

	Furnishing.Status	Correlation
	<chr>	<dbl>
1	Furnished	0.479
2	Semi-Furnished	0.449
3	Unfurnished	0.435

```
# Bar chart
ggplot(correlation_data, aes(x = Furnishing.Status, y = Correlation, fill = Furnishing.Status)) +
  geom_bar(stat="identity", position="dodge") +
  scale_fill_manual(values = c("violet", "purple", "#800080")) +
  geom_text(aes(label=sprintf("%.3f", Correlation)), vjust=-0.5, size=3.5) +
  labs(title = "Correlation between Size and Rent by Furnishing Status",
       x = "Furnishing Status",
       y = "Correlation Coefficient") +
  theme_minimal() +
  theme(legend.position = "none")
```



According to the picture above, the size of the property correlates with the rent and 'Furnished' status property with 0.479, 'Semi-Furnished' status property with 0.449, and 'Unfurnished' status property with 0.435.

8.3.6 Question 6: How skewed is the rent distribution for each furnishing status in each city?

```
install.packages("moments")
library("moments")
```

```
HouseRent_Cleaned %>%
  filter(City == "Bangalore") %>%
  group_by(Furnishing.Status) %>%
  summarise(Skewness = skewness(Rent, na.rm = TRUE))
```

	Furnishing.Status	Skewness
	<chr>	<dbl>
1	Furnished	3.13
2	Semi-Furnished	3.58
3	Unfurnished	4.88

```
HouseRent_Cleaned %>%
  filter(City == "Chennai") %>%
  group_by(Furnishing.Status) %>%
  summarise(Skewness = skewness(Rent, na.rm = TRUE))
```

	Furnishing.Status	Skewness
	<chr>	<dbl>
1	Furnished	2.53
2	Semi-Furnished	3.45
3	Unfurnished	3.67

```
HouseRent_Cleaned %>%
  filter(City == "Delhi") %>%
  group_by(Furnishing.Status) %>%
  summarise(Skewness = skewness(Rent, na.rm = TRUE))
```

	Furnishing.Status <chr>	Skewness <dbl>
1	Furnished	1.66
2	Semi-Furnished	2.20
3	Unfurnished	2.47

```
HouseRent_Cleaned %>%
  filter(City == "Hyderabad") %>%
  group_by(Furnishing.Status) %>%
  summarise(Skewness = skewness(Rent, na.rm = TRUE))
```

	Furnishing.Status <chr>	Skewness <dbl>
1	Furnished	2.55
2	Semi-Furnished	3.21
3	Unfurnished	5.67

```
HouseRent_Cleaned %>%
  filter(City == "Kolkata") %>%
  group_by(Furnishing.Status) %>%
  summarise(Skewness = skewness(Rent, na.rm = TRUE))
```

	Furnishing.Status <chr>	Skewness <dbl>
1	Furnished	1.72
2	Semi-Furnished	1.50
3	Unfurnished	2.23

```
HouseRent_Cleaned %>%
  filter(City == "Mumbai") %>%
  group_by(Furnishing.Status) %>%
  summarise(Skewness = skewness(Rent, na.rm = TRUE))
```

	Furnishing.Status <chr>	Skewness <dbl>
1	Furnished	1.67
2	Semi-Furnished	1.75
3	Unfurnished	3.26

First, to use the function skewness () we need to install the package “moments”. According to the results above, we can find out that the city ‘Bangalore’ has the highest skewness for the ‘Furnished’ properties which is 3.13, while ‘Delhi’ has the lowest skewness which is 1.66. For the ‘Semi-Furnished’ status, the city ‘Bangalore’ has the highest skewness which is up to 3.58 while the city ‘Kolkata’ has the lowest skewness which is 1.50. For the ‘Unfurnished’ status, the city ‘Hyderabad’ has the highest skewness, which is 5.67, while city ‘Kolkata’ has the lowest skewness which is 2.23.

8.3.7 Question 7: Can Furnishing Status predict the Rent price of each city?

```
# Can Furnishing Status predict Rent price in each city?
HouseRent_Cleaned_Preprocessed %>%
  group_by(City) %>%
  do(anova(lm(Rent ~ Furnishing.Status, data = .)))
```

	City	Df	`Sum Sq`	`Mean Sq`	`F value`	`Pr(>F)`
	<chr>	<int>	<dbl>	<dbl>	<dbl>	<dbl>
1	Bangalore	1	9.55e 9	9547621525.	20.6	6.42e- 6
2	Bangalore	880	4.08e11	463332004.	NA	NA
3	Chennai	1	2.00e10	19962126545.	59.2	3.88e-14
4	Chennai	882	2.98e11	337409491.	NA	NA
5	Delhi	1	4.79e 9	4792800462.	12.9	3.50e- 4
6	Delhi	582	2.16e11	370593156.	NA	NA
7	Hyderabad	1	1.90e10	18954400496.	66.3	1.34e-15
8	Hyderabad	861	2.46e11	285774224.	NA	NA
9	Kolkata	1	4.29e 8	428530231.	7.49	6.42e- 3
10	Kolkata	517	2.96e10	57222158.	NA	NA
11	Mumbai	1	3.50e11	349728073461.	67.2	7.95e-16
12	Mumbai	959	4.99e12	5207064124.	NA	NA

The figure above depicts the ANOVA testing for Furnishing Status and Rent in each city. According to the findings, the p-value of the main impact of Furnishing Status in each city is not near to 0 when compared to other variables, indicating that Furnishing Status and Rent do not have a highly significant association.

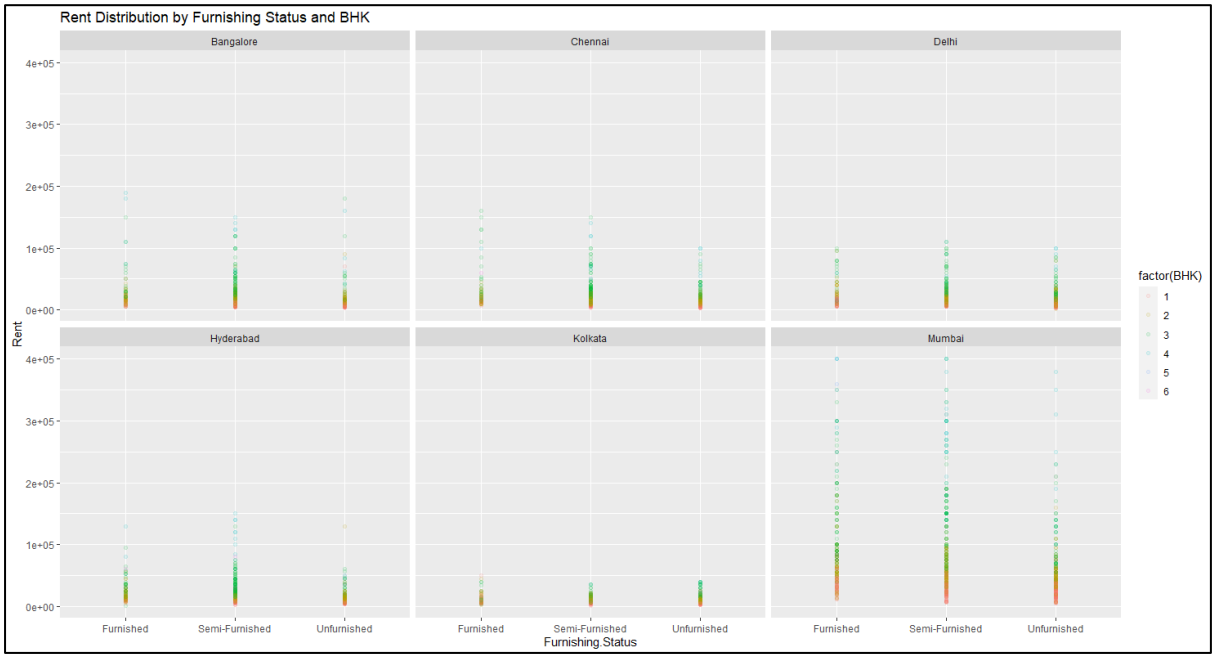
8.3.8 Question 8: How is the relationship between Furnishing Status and Rent with Size in different cities?

```
ggplot(HouseRent_Cleaned, aes(x = Size, y = Rent, z = factor(Furnishing.Status), color = factor(Furnishing.Status))) +
  geom_point(size = 3) +
  scale_color_manual(name = "Furnishing Status", values = c("violet", "purple", "#800080")) +
  labs(title = "Rent Distribution by Furnishing Status and City", x = "Size", y = "Rent") +
  facet_wrap(~City)
```

According to the graph above, we can see that people are more willingly to rent a house that size is more than 2500 in others city besides ‘Kolkata’ city. Moreover, people living in ‘Mumbai’ city has pay more rent than others city even having the same size.

8.3.9 Question 9: How is the relationship between Furnishing Status and Rent with BHK in different cities?

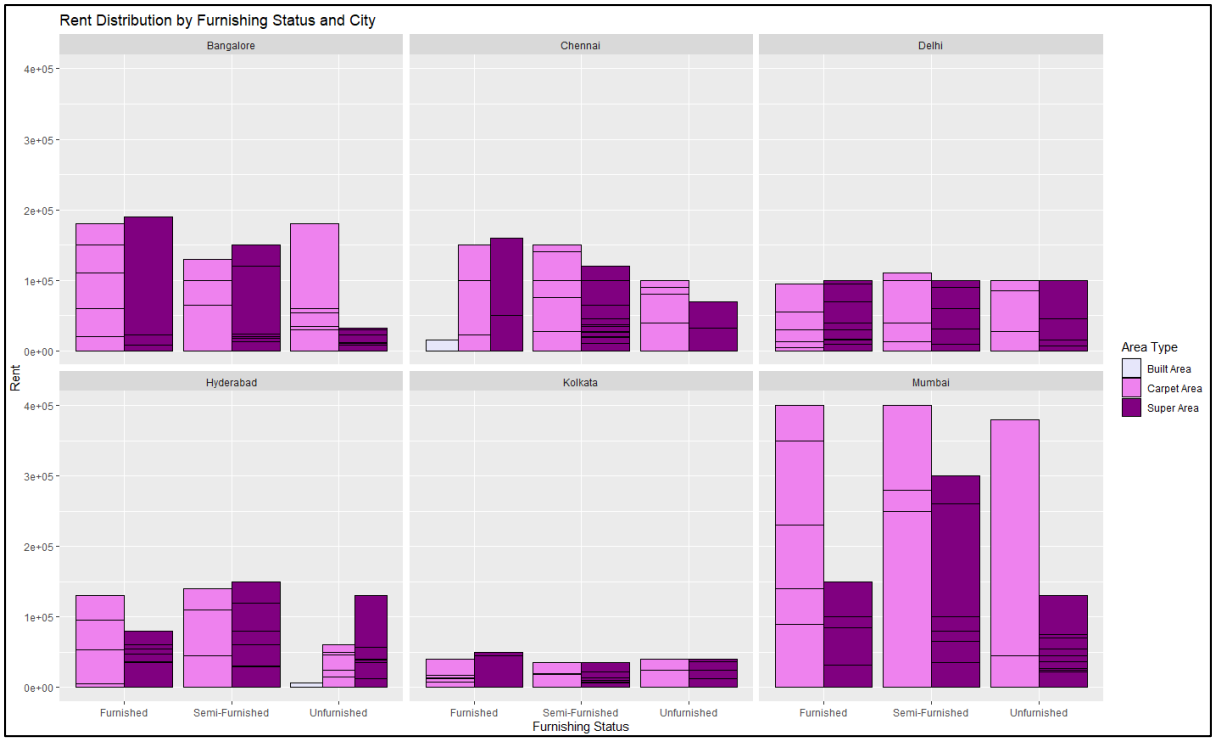
```
ggplot(HouseRent_Cleaned, aes(x = factor(Furnishing.Status), y = Rent, fill = factor(BHK), color = factor(BHK))) +
  geom_point(alpha = 0.1) +
  scale_size_continuous(range = c(1, 12)) +
  facet_wrap(~City) +
  labs(title = "Rent Distribution by Furnishing Status and BHK", x = "Furnishing.Status", y = "Rent")
```



The figure above shows prove that people will rent for a house up to 4 BHK in either ‘Furnished’ status, ‘Semi-Furnished’ status or ‘Unfurnished’ status.

8.3.10 Question 10: How is the relationship between Furnishing Status and Rent with Area Type in different cities?

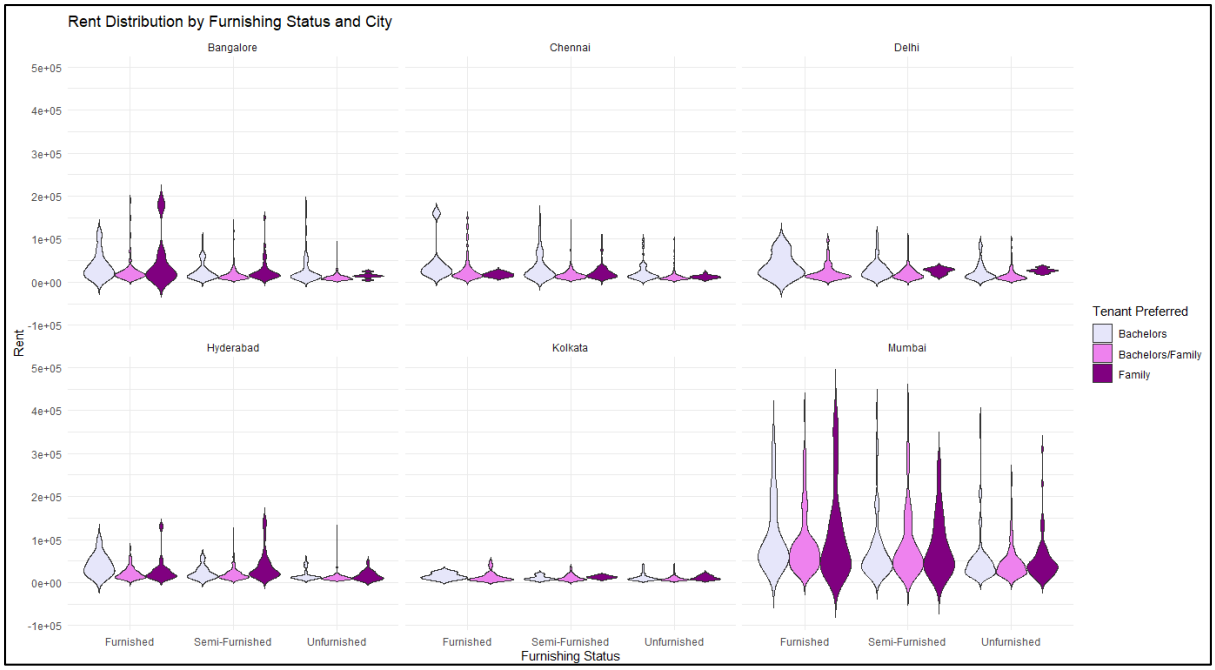
```
ggplot(HouseRent_Cleaned, aes(x = Furnishing.Status, y = Rent, fill = factor(Area.Type))) +
  geom_bar(stat = "identity", position = "dodge", color = "black") +
  facet_wrap(~City) +
  scale_fill_manual(name = "Area Type", values = c("#E6E6FA", "violet", "#800080")) +
  labs(title = "Rent Distribution by Furnishing Status and City", x = "Furnishing Status", y = "Rent")
```



The figure above prove that there are very least people would rent a house with ‘Build Area’ across all the cities. Besides, the overall of ‘Carpet Area’ is more than ‘Super Area’ across the cities.

8.3.11 Question 11: How is the relationship between Furnishing Status and Rent with Tenant Preferred in different cities?

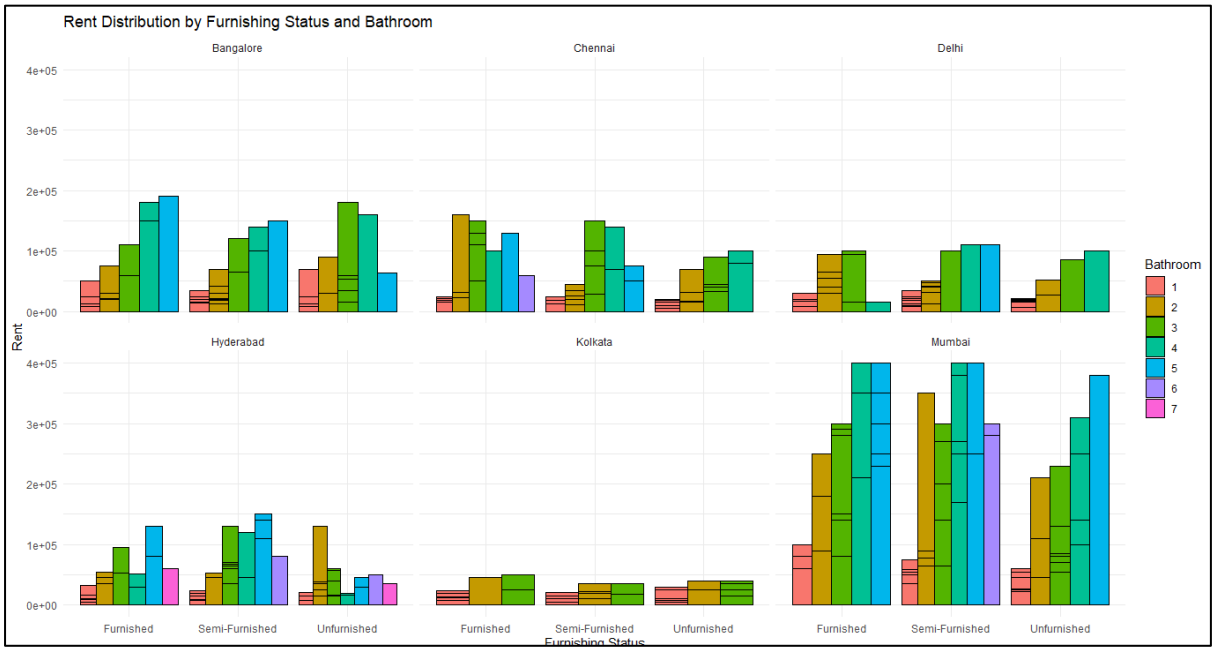
```
ggplot(HouseRent_Cleaned, aes(x = Furnishing.Status, y = Rent, fill = Tenant.Preferred)) +
  geom_violin(scale = "width", trim = FALSE) +
  facet_wrap(~City) +
  scale_fill_manual(name = "Tenant Preferred", values = c("#E6E6FA", "violet", "#800080")) +
  labs(title = "Rent Distribution by Furnishing Status and City", x = "Furnishing Status", y = "Rent") +
  theme_minimal()
```



The figure above proved that every tenant preferred in all city are average.

8.3.12 Question 12: How is the relationship between Furnishing Status and Rent with Bathroom in different cities?

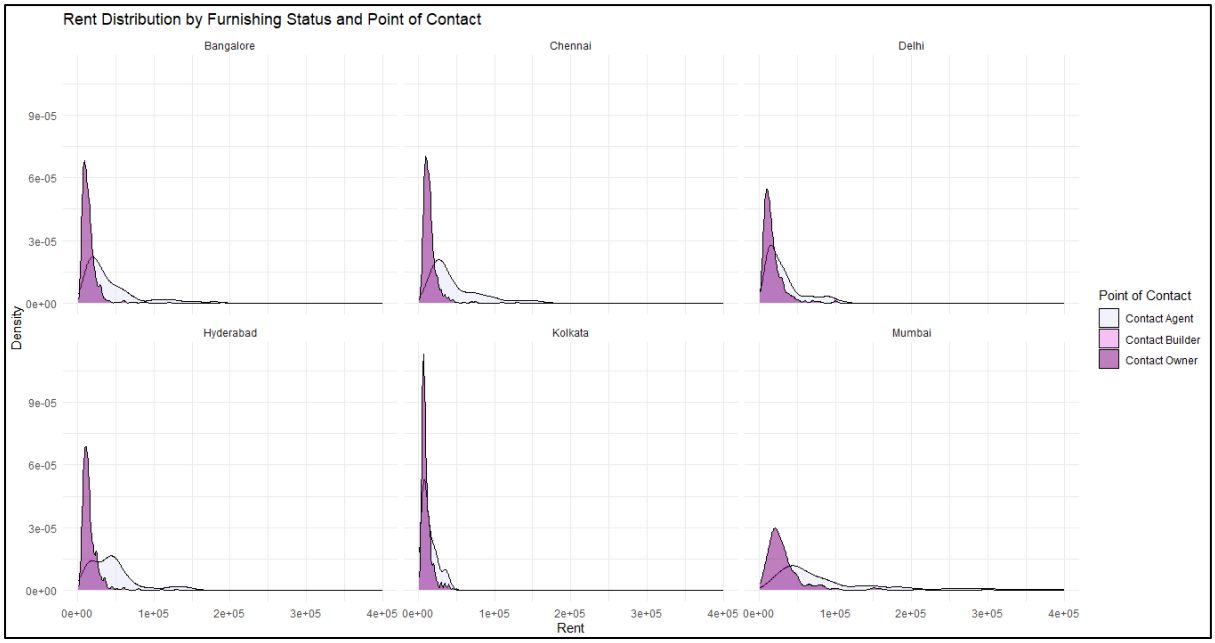
```
ggplot(HouseRent_Cleaned, aes(x = Furnishing.Status, y = Rent, fill = factor(Bathroom))) +
  geom_bar(stat = "identity", position = "dodge", color = "black") +
  facet_wrap(~City) +
  scale_fill_discrete(name = "Bathroom") +
  labs(title = "Rent Distribution by Furnishing Status and Bathroom", x = "Furnishing Status", y = "Rent") +
  theme_minimal()
```



The figure above proved that in city ‘Bangalore’, ‘Chennai’ and ‘Delhi’, people will rent the house with 5 bathrooms. In ‘Kolkata’, only up to 3 bathrooms. ‘Hyderabad’ and ‘Mumbai’ city is between 1 to 7 bathrooms.

8.3.13 Question 13: How is the relationship between Furnishing Status and Rent with Point of Contact in different cities?

```
ggplot(HouseRent_Cleaned, aes(x = Rent, fill = Point.of.Contact)) +
  geom_density(alpha = 0.5) +
  labs(title = "Rent Distribution by Furnishing Status and Point of Contact", x = "Rent", y = "Density") +
  scale_fill_manual(name = "Point of Contact", values = c("#E6E6FA", "violet", "#800080")) +
  facet_wrap(~City) +
  theme_minimal()
```



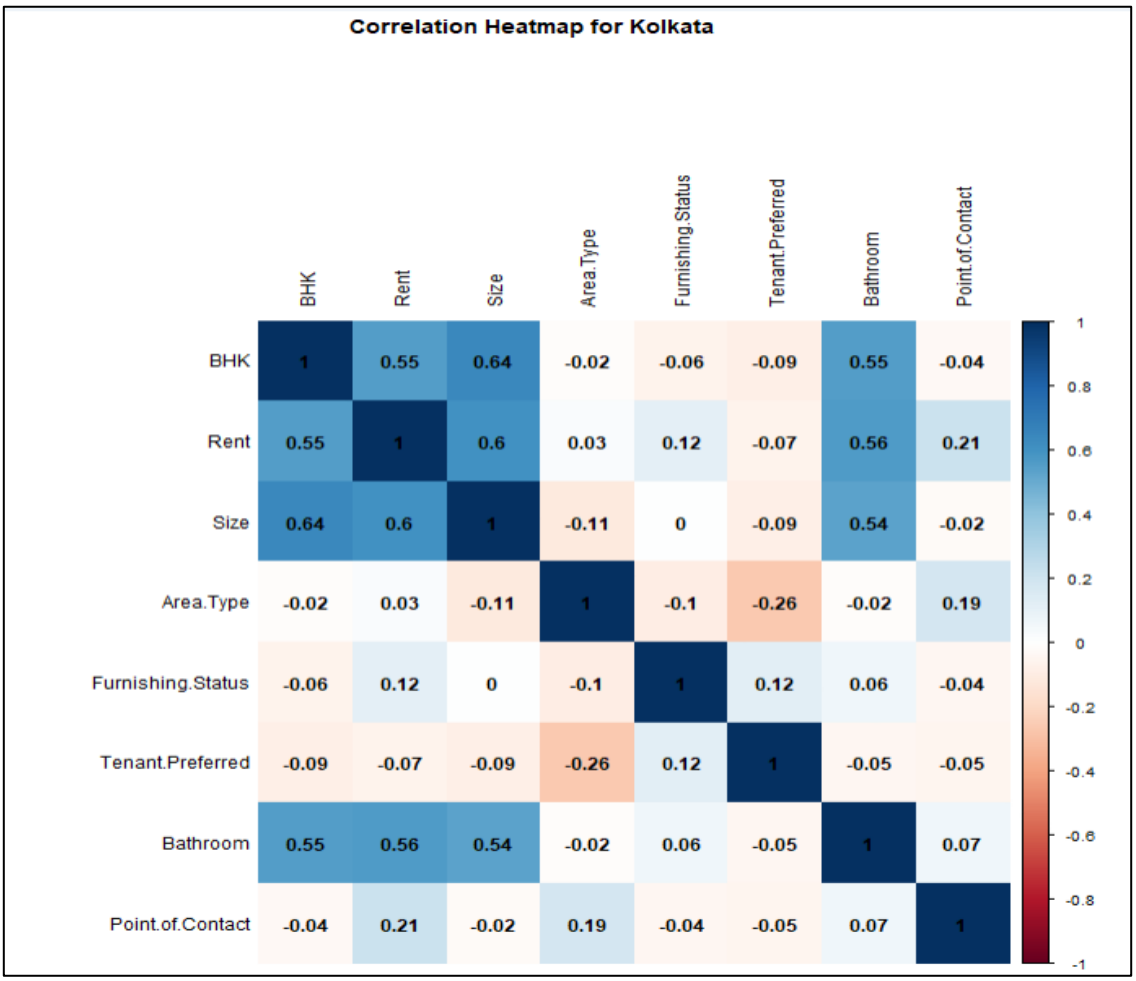
From the figure above, we can get a conclusion that the point of contact across all city are based on contact agent and contact owner only. Contact owner has the highest rate across all cities.

9.0 Pair Plot and Correlation Heat map

In this section, we use the ggpairs() function from GGally packages, to draw a simple plot based on two variables. In our hypothesis, house 'Rent' is the dependent variable. Hence, we use several meaningful categorical variables and plot them with 'Rent'. By using the ggpairs() function, a correlation of two variable will be computed. In the light of this, we use corrplot() to create a heat map of correlation for better visualisation. For GGpairs codes and results, please refers to Appendix 1.0.

9.1 Kolkata City Correlation Heat Map

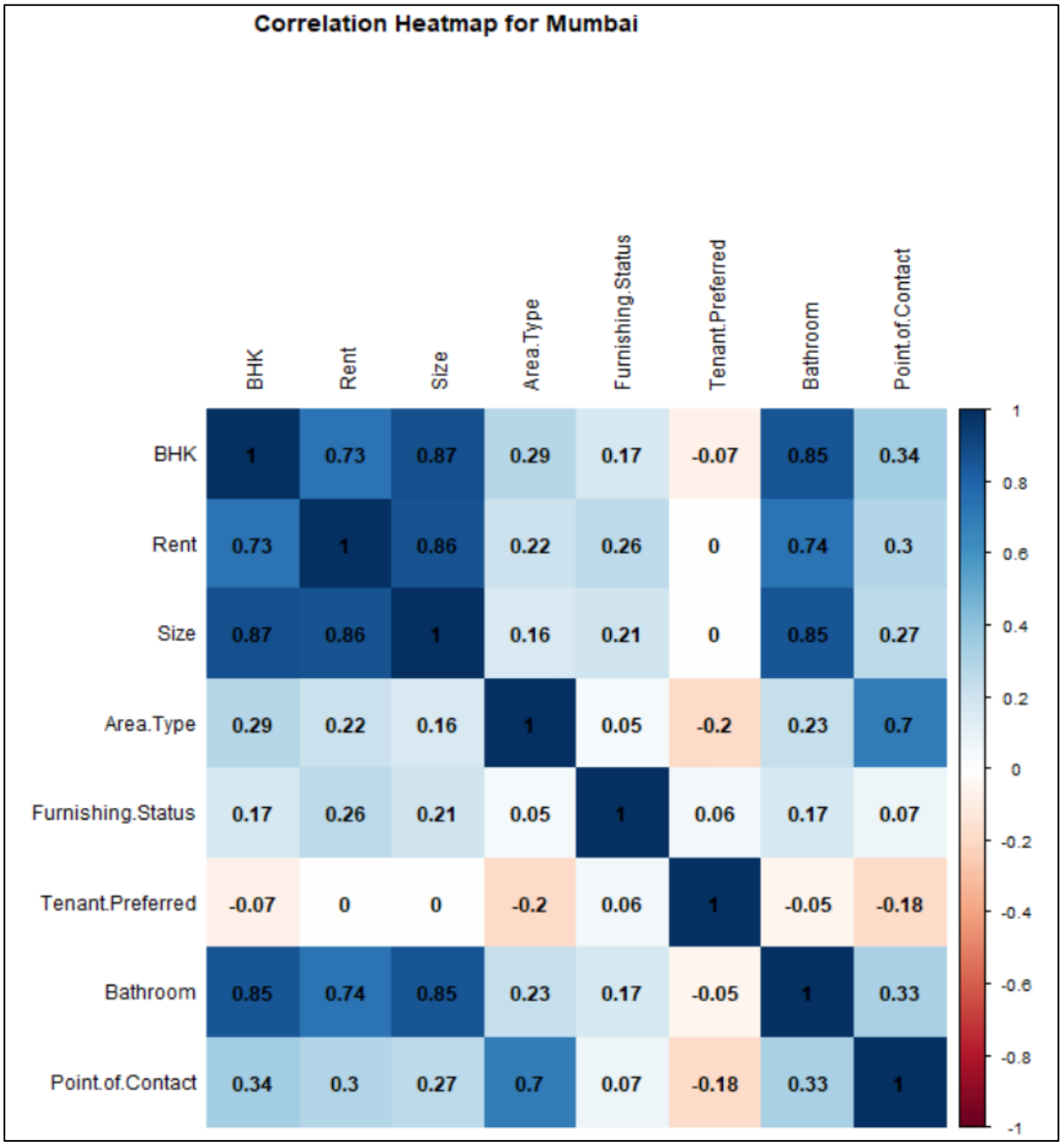
```
HouseRent_Cleaned_Preprocessed_Kolkata = city_subsets_preprocessed$Kolkata
HouseRent_Cleaned_Preprocessed_Kolkata <- select(HouseRent_Cleaned_Preprocessed_Kolkata, -City)

cor_matrix_Kolkata = cor(HouseRent_Cleaned_Preprocessed_Kolkata)

corrplot(cor_matrix_Kolkata, method = "color", addCoef.col = "black", tl.col = "black")

title("Correlation Heatmap for Kolkata", line = 3)
```



By using the correlation calculation, we understand that in Kolkata City, 'Size', 'BHK' and 'Bathroom' are the first 3 high attribute that impact the 'Rent'.

9.2 Mumbai City Correlation Heat Map

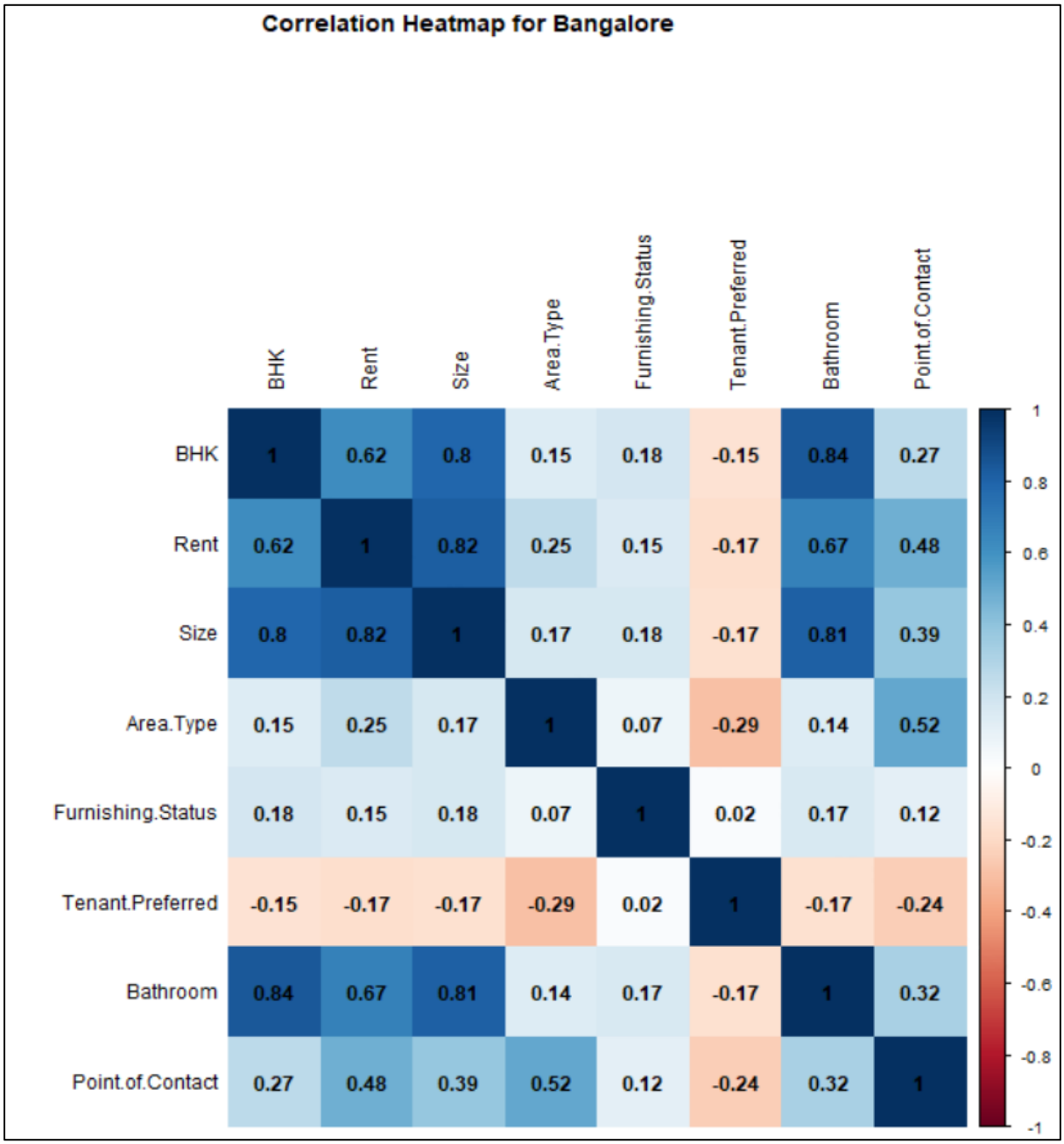
```
HouseRent_Cleaned_Preprocessed_Mumbai = city_subsets_preprocessed$Mumbai
HouseRent_Cleaned_Preprocessed_Mumbai <- select(HouseRent_Cleaned_Preprocessed_Mumbai, -City)

cor_matrix_Mumbai = cor(HouseRent_Cleaned_Preprocessed_Mumbai)

corrplot(cor_matrix_Mumbai, method = "color", addCoef.col = "black", tl.col = "black")

title("Correlation Heatmap for Mumbai", line = 3)
```



Based on the Heat Map above, we can understand that in Mumbai city, ‘BHK’, ‘Size’ and ‘Bathroom’ have the highest impact to ‘Rent’.

9.3 Bangalore City Correlation Heat Map

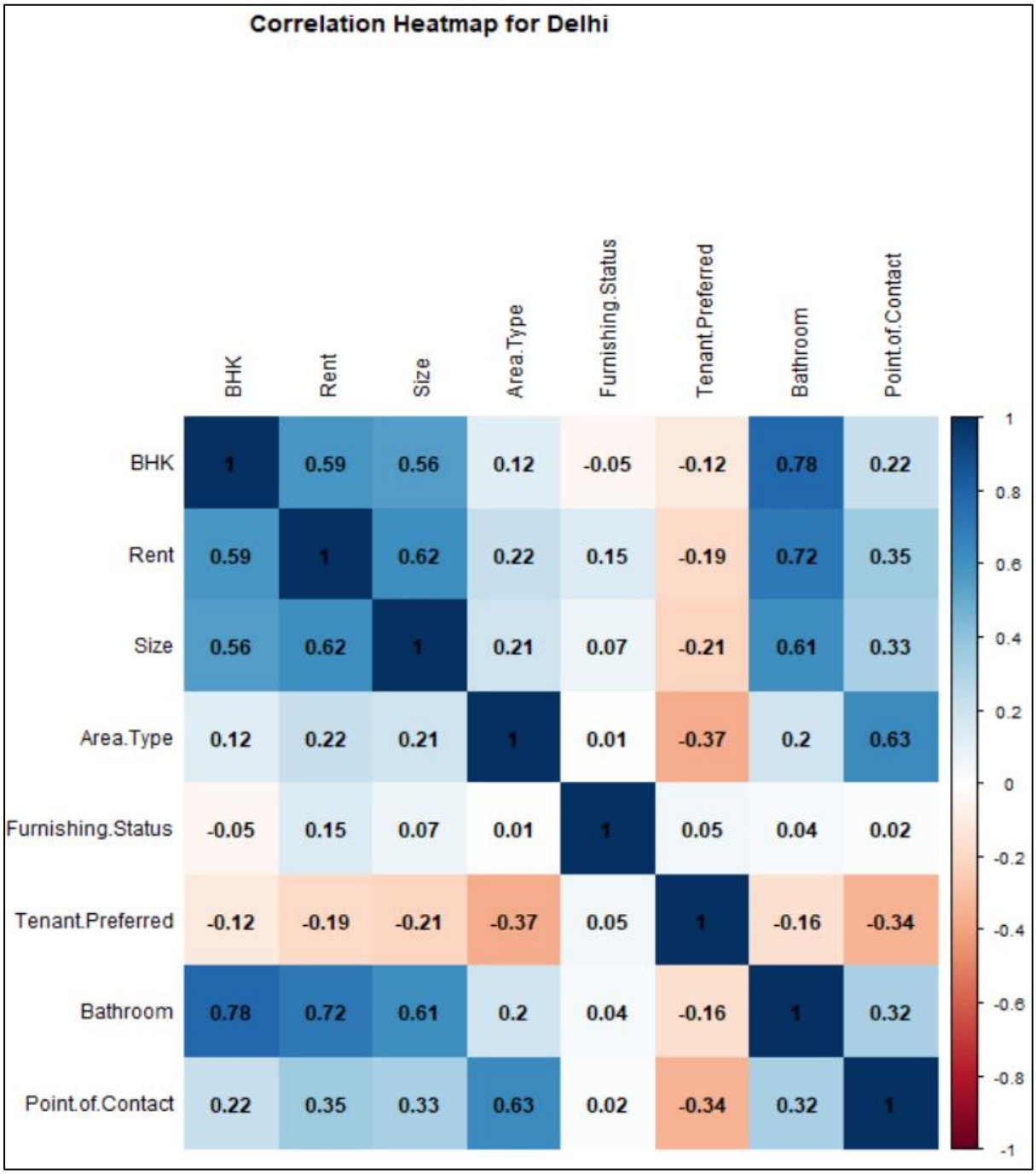
```
HouseRent_Cleaned_Preprocessed_Bangalore = city_subsets_preprocessed$Bangalore
HouseRent_Cleaned_Preprocessed_Bangalore <- select(HouseRent_Cleaned_Preprocessed_Bangalore, -City)

cor_matrix_Bangalore = cor(HouseRent_Cleaned_Preprocessed_Bangalore)

corrplot(cor_matrix_Bangalore, method = "color", addCoef.col = "black", tl.col = "black")

title("Correlation Heatmap for Bangalore", line = 3)
```



Based on figure above, ‘BHK’, ‘Size’ and ‘Bathroom’ have a high score in correlation with ‘Rent’ in Bangalore City.

9.4 Delhi City Correlation Heat Map

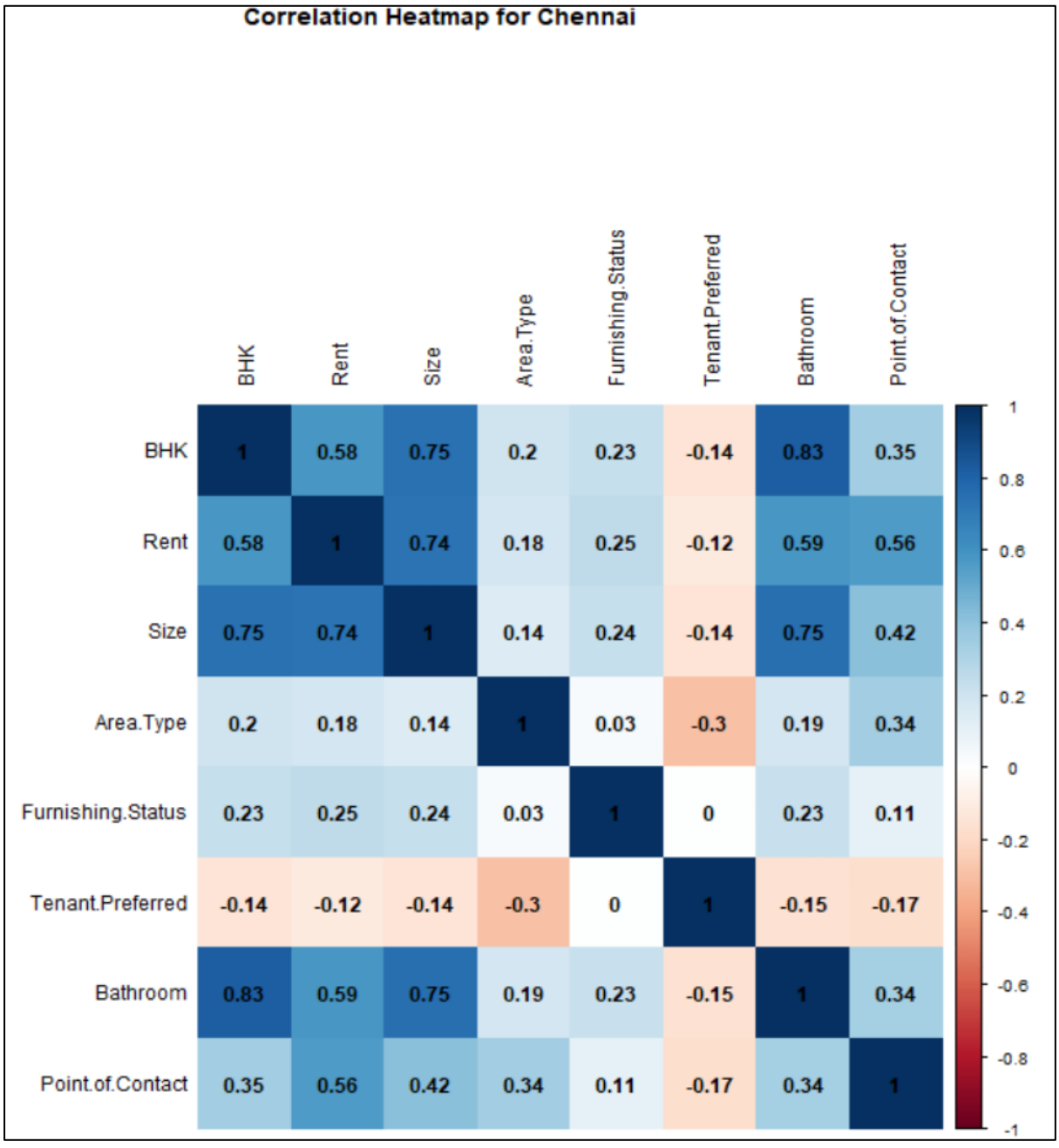
```
HouseRent_Cleaned_Preprocessed_Delhi = city_subsets_preprocessed$Delhi
HouseRent_Cleaned_Preprocessed_Delhi <- select(HouseRent_Cleaned_Preprocessed_Delhi, -City)

cor_matrix_Delhi = cor(HouseRent_Cleaned_Preprocessed_Delhi)

corrplot(cor_matrix_Delhi, method = "color", addCoef.col = "black", tl.col = "black")

title("Correlation Heatmap for Delhi", line = 3)
```



In Delhi City, ‘BHK’, ‘Rent’ and ‘Bathroom’ remain the highest three attribute that impact ‘Rent’.

9.5 Chennai City Correlation Heat Map

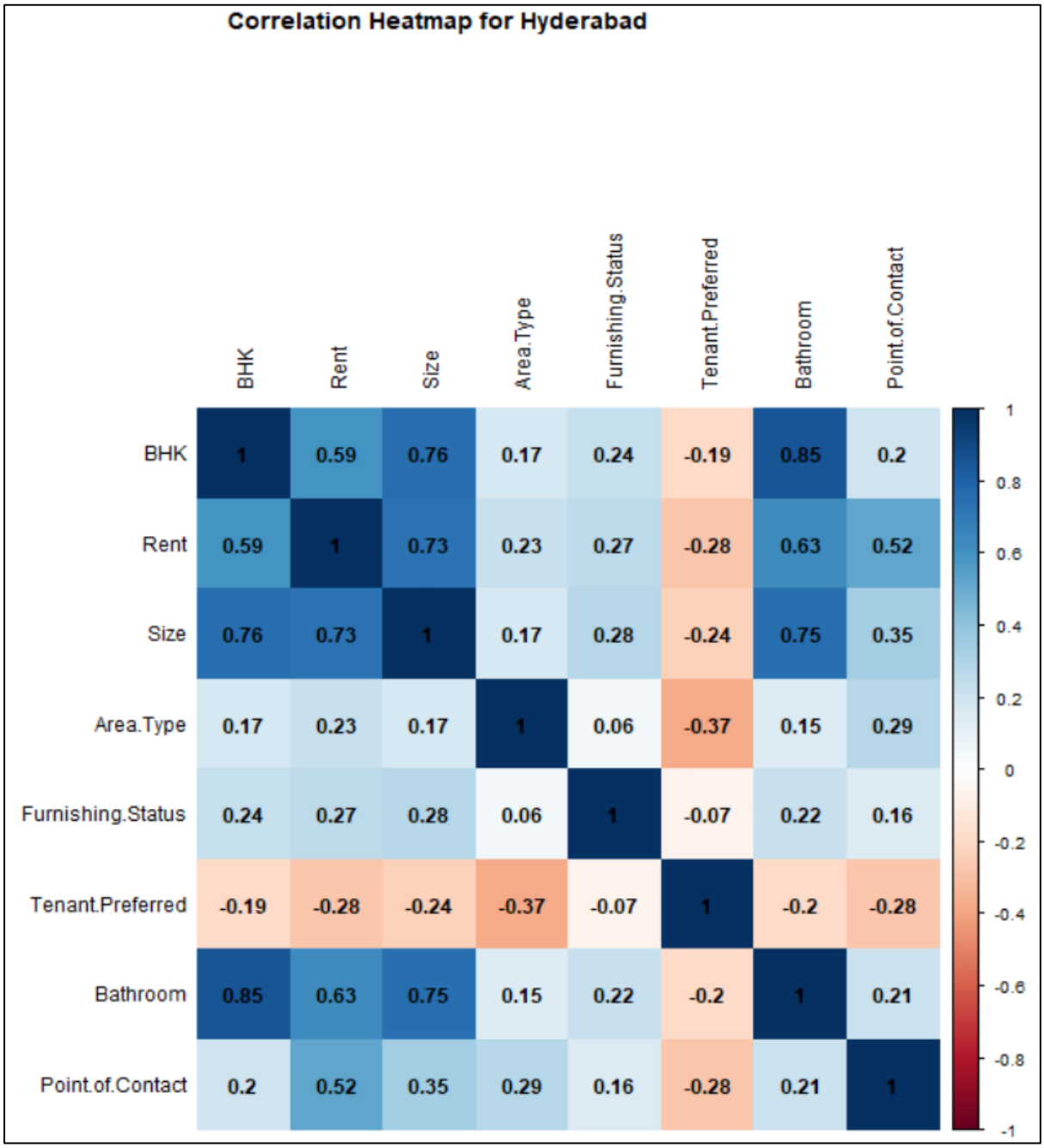
```
HouseRent_Cleaned_Preprocessed_Chennai = city_subsets_preprocessed$Chennai
HouseRent_Cleaned_Preprocessed_Chennai <- select(HouseRent_Cleaned_Preprocessed_Chennai, -City)

cor_matrix_Chennai = cor(HouseRent_Cleaned_Preprocessed_Chennai)

corrplot(cor_matrix_Chennai, method = "color", addCoef.col = "black", tl.col = "black")

title("Correlation Heatmap for Chennai", line = 3)
```



Heat map above shows in Chennai city, ‘BHK’, ‘Size’, ‘Bathroom’ and ‘Point of Contact’ have the similar high impact to ‘Rent’.

9.6 Hyderabad City Correlation Heat Map

```
HouseRent_Cleaned_Preprocessed_Hyderabad = city_subsets_preprocessed$Hyderabad
HouseRent_Cleaned_Preprocessed_Hyderabad <- select(HouseRent_Cleaned_Preprocessed_Hyderabad, -City)

cor_matrix_Hyderabad = cor(HouseRent_Cleaned_Preprocessed_Hyderabad)

corrplot(cor_matrix_Hyderabad, method = "color", addCoef.col = "black", tl.col = "black")

title("Correlation Heatmap for Hyderabad", line = 3)
}
```



Based on heat map above, in Hyderabad city, ‘BHK’, ‘Size’, ‘Bathroom’ and ‘Point of Contact’ have the similar high impact to ‘Rent’

10.0 ANOVA Test

This section we will introduce ANOVA testing to test our model accuracy with the p-value. We will assume a high accuracy of two variable if the p-value is lower than 0.05 (<0.05). The lower the p-value is, the higher the accuracy between two variables. Same as the correlation above, we will use the meaningful categorical variable to have the ANOVA test with 'Rent' price.

For the ANOVA testing, we are only focusing on the Pr(>F) column, which pointed to the p-value. For better understanding, we also make a bar chart for clear view. Do take note that for actual code and result will be included in Appendix 2.0. This section will only show the result.

10.1 Kolkata City ANOVA Testing

Summary of the ANOVA p-value in Kolkata City

Attribute	P-value	Impact?
BHK	<0.001	High
Size	<0.001	High
Bathroom	<0.001	High
Point of Contact	<0.001	High
Furnishing Status	<0.01	Upper Medium
Tenant Preferred	>0.05	Low
Area Type	>0.05	Low

10.2 Mumbai City ANOVA Testing

Summary of the ANOVA p-value in Mumbai City

Attribute	P-value	Impact?
BHK	<0.001	High
Size	<0.001	High
Bathroom	<0.001	High
Point of Contact	<0.001	High
Furnishing Status	<0.001	High
Area Type	<0.001	High
Tenant Preferred	>0.05	Low

10.3 Bangalore City ANOVA Testing

Summary of the ANOVA p-value in Bangalore City

Attribute	P-value	Impact?
BHK	<0.001	High
Size	<0.001	High
Bathroom	<0.001	High
Point of Contact	<0.001	High
Furnishing Status	<0.001	High
Area Type	<0.001	High
Tenant Preferred	<0.001	High

10.4 Delhi City ANOVA Testing

Summary of the ANOVA p-value in Delhi City

Attribute	P-value	Impact?
BHK	<0.001	High
Size	<0.001	High
Bathroom	<0.001	High
Point of Contact	<0.001	High
Furnishing Status	<0.001	High
Area Type	<0.001	High
Tenant Preferred	<0.001	High

10.5 Chennai City ANOVA Testing

Summary of the ANOVA p-value in Chennai City

Attribute	P-value	Impact?
BHK	<0.001	High
Size	<0.001	High
Bathroom	<0.001	High
Point of Contact	<0.001	High
Furnishing Status	<0.001	High
Area Type	<0.001	High
Tenant Preferred	<0.001	High

10.6 Hyderabad City ANOVA Testing

Summary of the ANOVA p-value in Hyderabad City

Attribute	P-value	Impact?
BHK	<0.001	High
Size	<0.001	High
Bathroom	<0.001	High
Point of Contact	<0.001	High
Furnishing Status	<0.001	High
Area Type	<0.001	High
Tenant Preferred	<0.001	High

11.0 Conclusion

In this report, we have made a hypothesis for a dataset, and we did relevant process to prove the hypothesis such as data cleaning, data pre-processing, data analysis and data visualisation. As a conclusion, we have proven that our hypothesis, which is 'BHK, house size and house furnishing status will affect the house rental more than the other variable in each city' is partially correct. To further elaborate that, all cities will have BHK, House Size and Bathroom as the highest three attribute that affecting house Rent.

We have performed various data analysis techniques and data visualisation methods to explore the relationship between rent and other factors such as BHK, size, furnishing status, area type, etc. We have also built and evaluated different predictive models such as linear regression, decision tree, and random forest to estimate the rent based on the features. Our findings are summarized as follows:

- BHK, size, and furnishing status are the most important factors that affect the rent price in each city, followed by point of contact and bathroom. Area type and tenant preferred to have relatively lower impact on the rent price.
- The rent price varies significantly across different cities, with Mumbai having the highest average rent and Kolkata having the lowest average rent. The rent price also increases with the number of BHKs, the size of the house, and the number of bathrooms.
- The furnishing status of the house has a positive effect on the rent price, with furnished houses having higher rent than semi-furnished or unfurnished houses. However, the furnishing status is not a strong predictor of the rent price compared to other factors.
- The random forest model has the best predictive performance among the three models we tested, with an R-squared score of 98.59%, indicating that it captures most of the variance in the data. The decision tree model offers a balance between performance and interpretability, but its performance is notably lower than the random forest model. The linear regression model has the lowest performance and suggests that there might be non-linear relationships or interactions in the data that it is not capturing.

Some of the limitations and recommendations of our analysis are:

- The dataset we used may not be representative of the entire house rent market in India, as it only covers six cities and may have some selection bias or sampling errors.

Therefore, we recommend collecting more data from other cities and regions to increase the generalizability and validity of our findings.

- Other relevant features that affect the rent price are not included, such as location, amenities, age of the property, etc. Therefore, we recommend adding more features to capture more information and improve the accuracy of our models.
- Outliers or errors could affect our analysis results. Therefore, we recommend conducting more data cleaning and validation steps to ensure the quality and reliability of our data.
- The models we used may have some assumptions or limitations that could affect their performance or interpretation. Therefore, it is essential to test a few more different models or parameters to find the best fit for our data and objectives.

References

Arbor Custom Analytics. (2021, October 27). Random forests: A tutorial with Forestry Data. https://arbor-analytics.com/post/2021-09-26-random-forests-a-tutorial-with-forestry-data/

Chu X., Ilyas I. F., Krishnan S. & Wang J. (2016). Data Cleaning. Proceeding of the 2016 International Conference on Management of Data. https://sci-hub.se/https://doi.org/10.1145/2882903.2912574

Kenton, W. (2022, December 31). *Homoskedastic: What it means in regression modeling, with example*. Investopedia. https://www.investopedia.com/terms/h/homoskedastic.asp

Soetewey, A. (2020, October 12). *ANOVA in R*. Stats and R. https://statsandr.com/blog/anova-in-r/

Suad A. Alasadi & Wesam S. Bhaya (2017). Review of Data Preprocessing Techniques in Data Mining. Journal of Engineering and Applied Science. https://d1wqtxts1xzle7.cloudfront.net/54509277/4102-4107-libre.pdf?1506113528=&response-content-disposition=inline%3B+filename%3DReview_of_Data_Preprocessing_Techniques.pdf&Expires=1692552664&Signature=HvUXNiu6hzlSLM8Qbu-x2m80ygGQOKeBVPDzP6ilaCYVToMkbarXlWXfK-Lc3j0rB-4lxvKuAwwvfzBVVI3u2qfE65RgRL9NVHz0Uiq7WXAm3TJuXMXkDWz7t2Yli~HQ9oMqJtRlJNsoTw7Uh0PhtXsq5ei~UM7ebLWN7tFlajHF02d7m~T5xZZNOhDfI8MDmEiP3s7kf1ZQsRp8yhPBa6WoLM4x05BKlm4fVnTny9LxOMQr35H1Ck0XA0wO2VR5etSYE1Ue8vBrV5asEWMoxOuW4F0he2m6tiWJ4g7-TTY29e-bRp~UTagvN~-23FfugfJzakFxhFqmqvcWnHgBXw__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA

Zach. (2021, July 12). *How to use Q-Q plots to check normality*. Statology. https://www.statology.org/q-q-plot-normality/

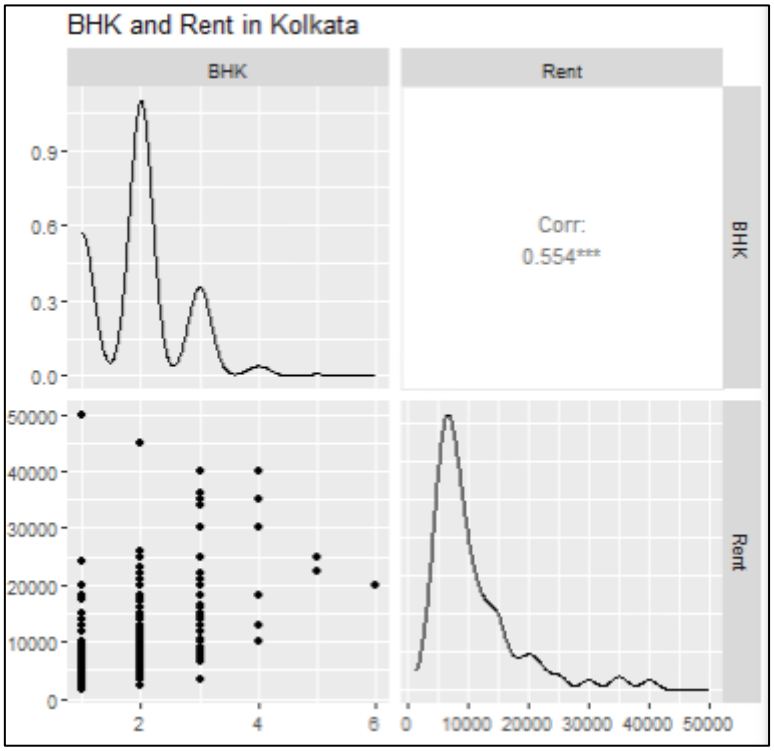
Appendix

1.0 Ggpair plot

1.1 Kolkata City – Pair plot

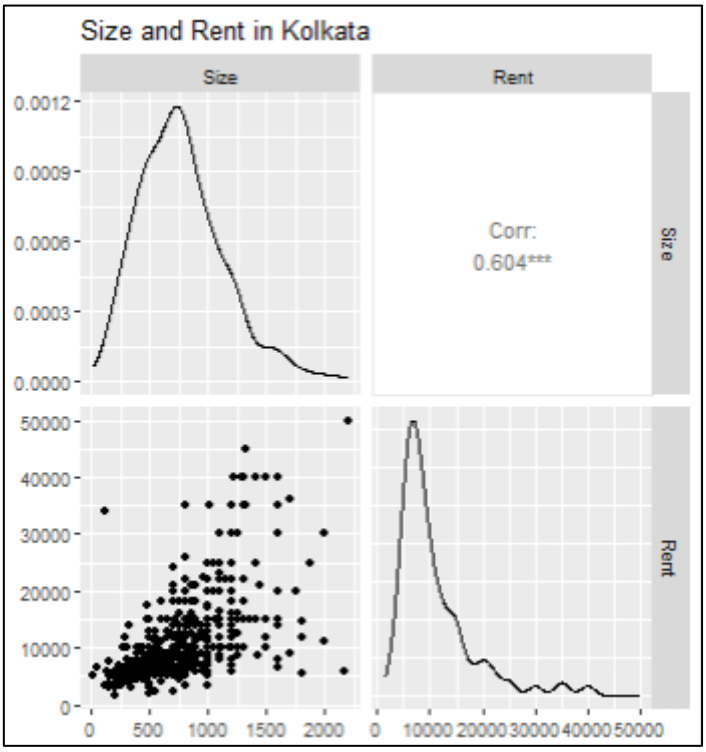
1.1.1 BHK and Rent

```
ggpairs(city_subsets_preprocessed$Kolkata, title = "BHK and Rent in Kolkata",columns = c("BHK", "Rent"))#0.554
```



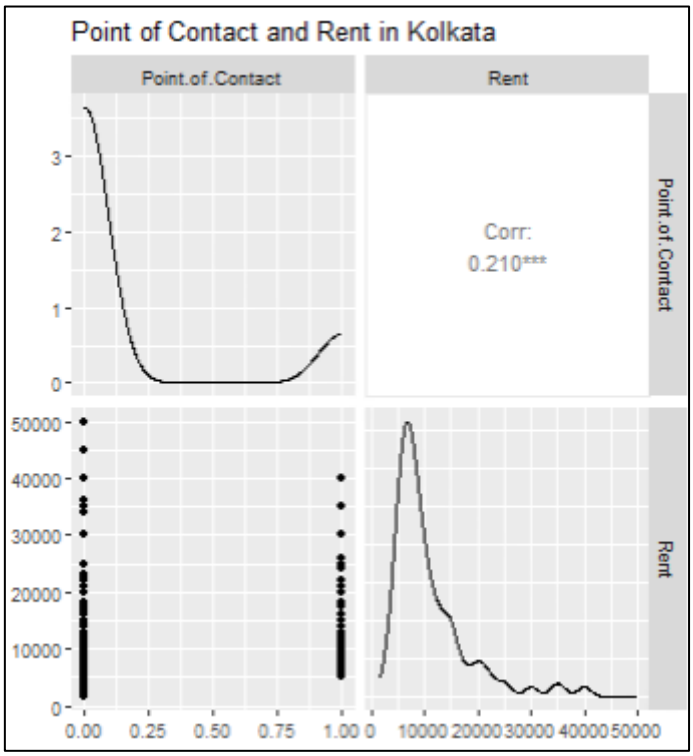
1.1.2 Size and Rent

```
ggpairs(city_subsets_preprocessed$Kolkata, title = "Size and Rent in Kolkata",columns = c("Size", "Rent"))#0.604
```



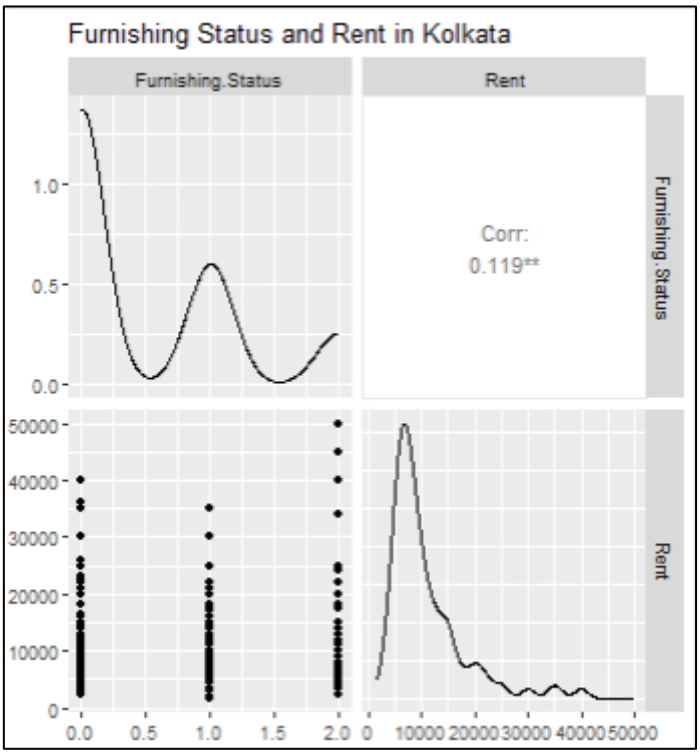
1.1.3 Point of Contact and Rent

```
ggpairs(city_subsets_preprocessed$Kolkata, title = "Point of Contact and Rent in Kolkata",columns = c("Point.of.Contact", "Rent"))#0.210
```



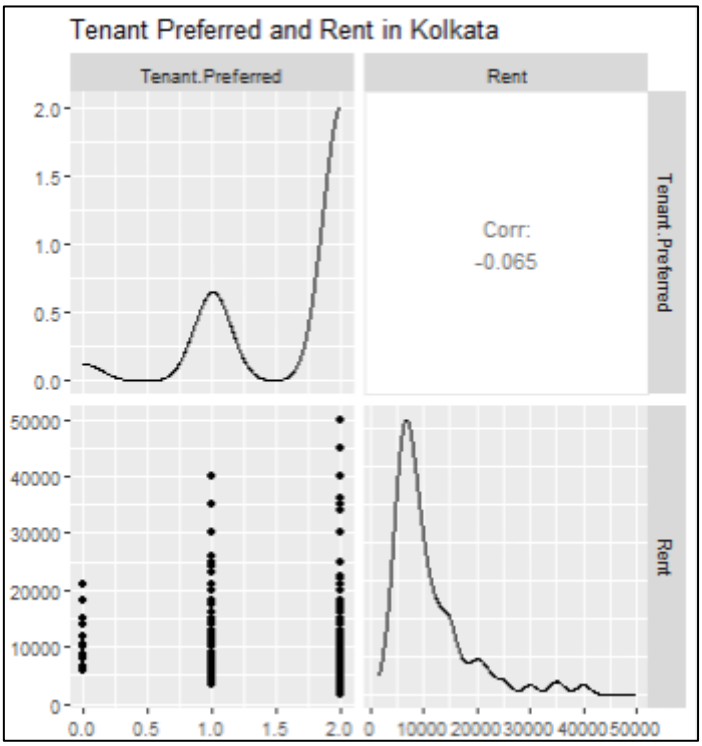
1.1.4 Furnishing Status and Rent

```
ggpairs(city_subsets_preprocessed$Kolkata, title = "Furnishing Status and Rent in Kolkata",columns = c("Furnishing.Status", "Rent"))#0.119
```



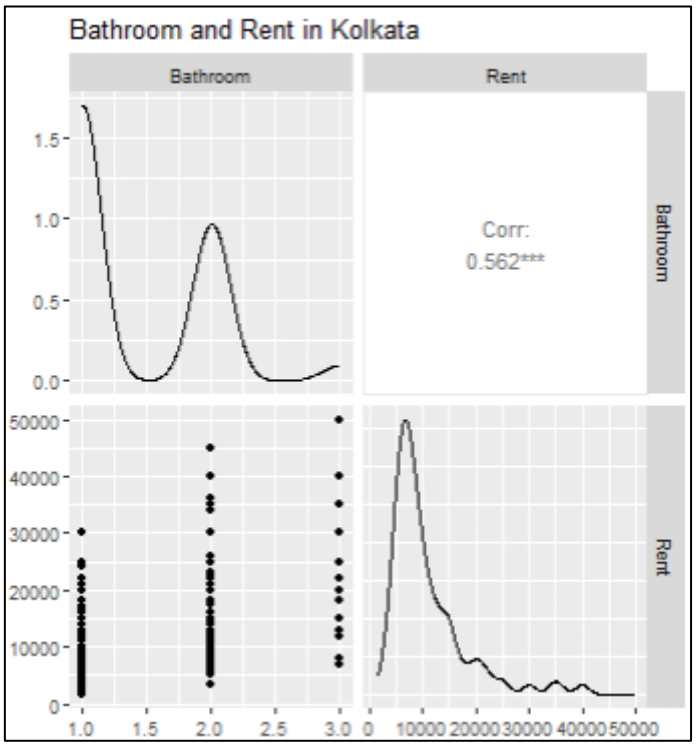
1.1.5 Tenant Preferred and Rent

```
ggpairs(city_subsets_preprocessed$Kolkata, title = "Tenant Preferred and Rent in Kolkata",columns = c("Tenant.Preferred", "Rent"))#-0.065
```



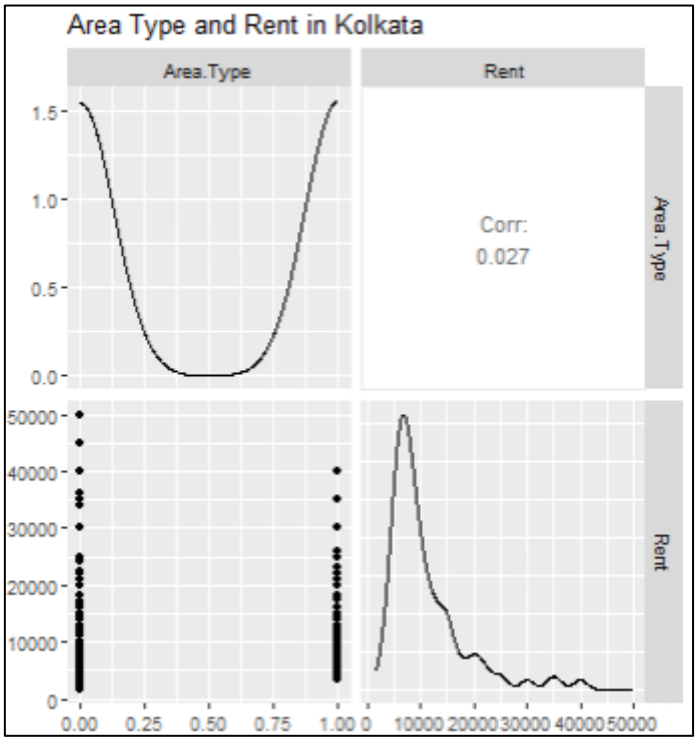
1.1.6 Bathroom and Rent

```
ggpairs(city_subsets_preprocessed$Kolkata, title = "Bathroom and Rent in Kolkata",columns = c("Bathroom", "Rent"))#0.562
```



1.1.7 Area Type and Rent

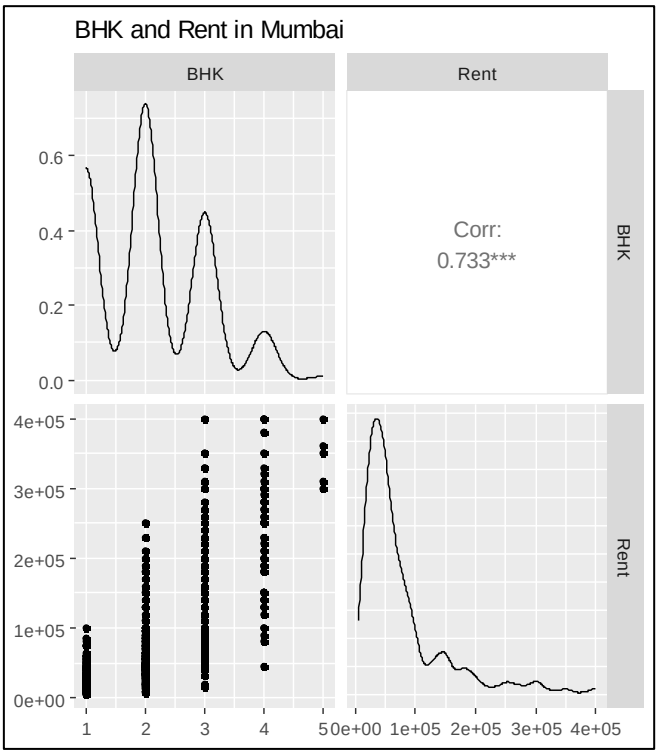
```
ggpairs(city_subsets_preprocessed$Kolkata, title = "Area Type and Rent in Kolkata",columns = c("Area.Type", "Rent"))#0.027
```



1.2 Mumbai City – Pair plot

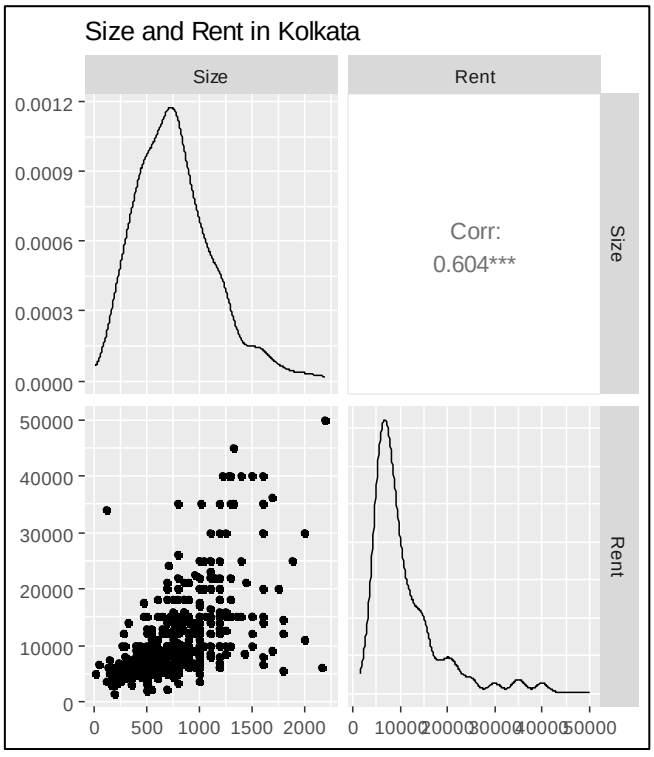
1.2.1 BHK and Rent

```
ggpairs(city_subsets_preprocessed$Mumbai, title = "BHK and Rent in Mumbai",columns = c("BHK", "Rent"))#0.733
```



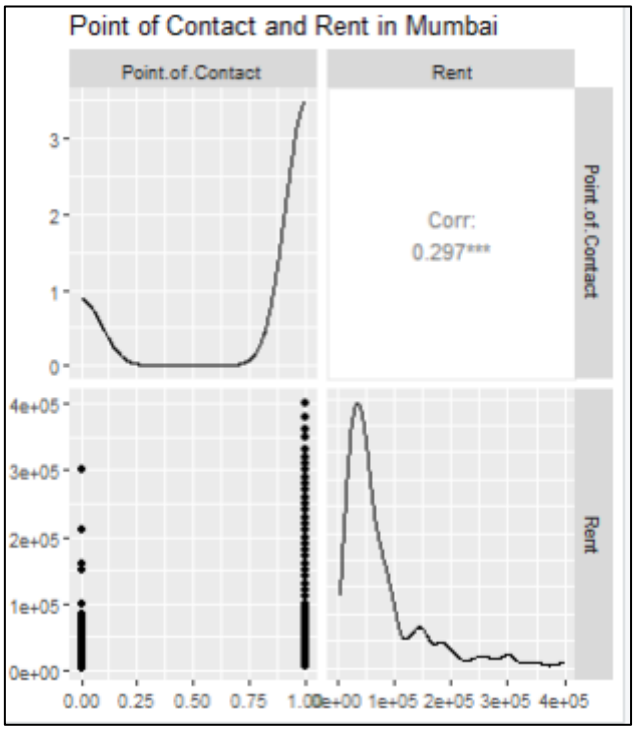
1.2.2 Size and Rent

```
ggpairs(city_subsets_preprocessed$Mumbai, title = "Size and Rent in Mumbai",columns = c("Size", "Rent"))#0.860
```



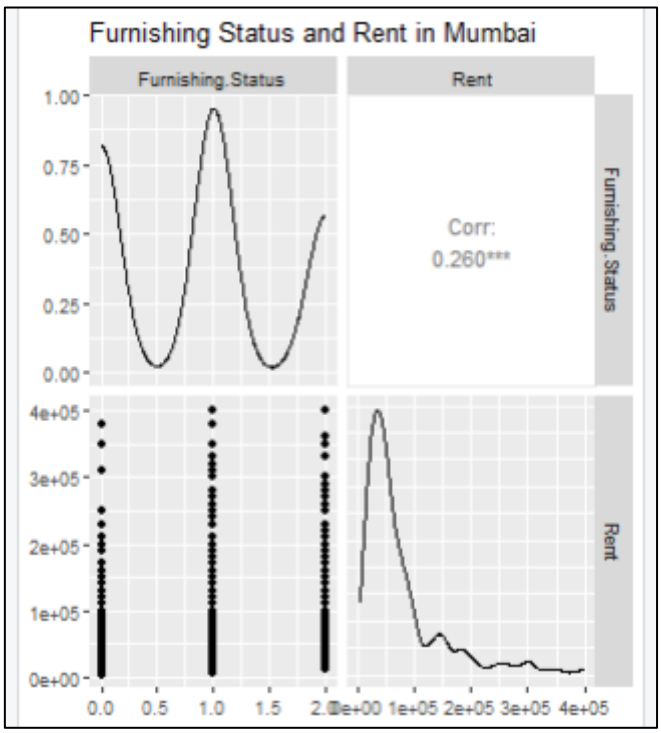
1.2.3 Point of Contact and Rent

```
ggpairs(city_subsets_preprocessed$Mumbai, title = "Point of Contact and Rent in Mumbai",columns = c("Point.of.Contact", "Rent"))#0.297
```



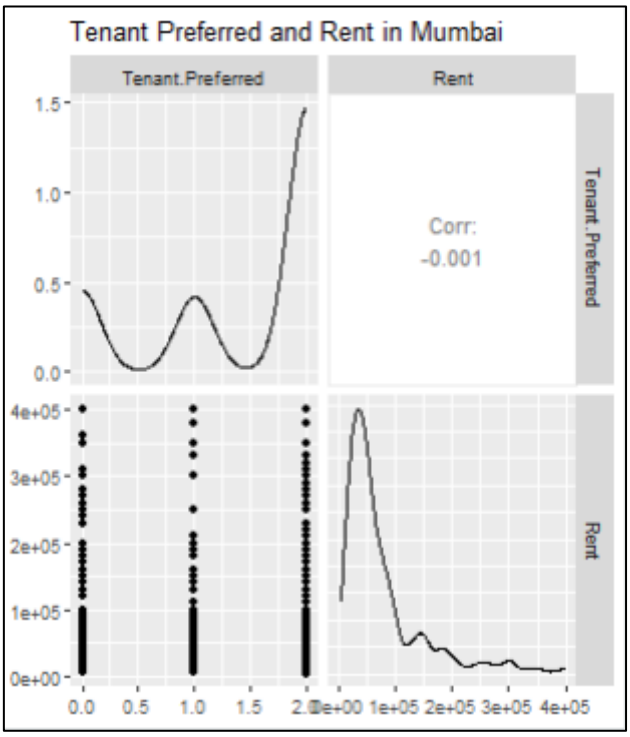
1.2.4 Furnishing Status and Rent

```
ggpairs(city_subsets_preprocessed$Mumbai, title = "Furnishing Status and Rent in Mumbai",columns = c("Furnishing.Status", "Rent"))#0.260
```



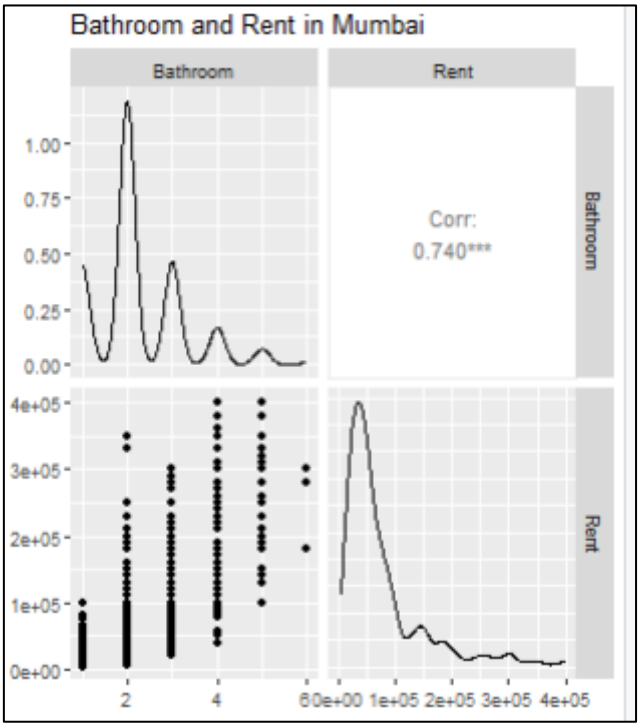
1.2.5 Tenant Preferred and Rent

```
ggpairs(city_subsets_preprocessed$Mumbai, title = "Tenant Preferred and Rent in Mumbai",columns = c("Tenant.Preferred", "Rent"))#-0.001
```



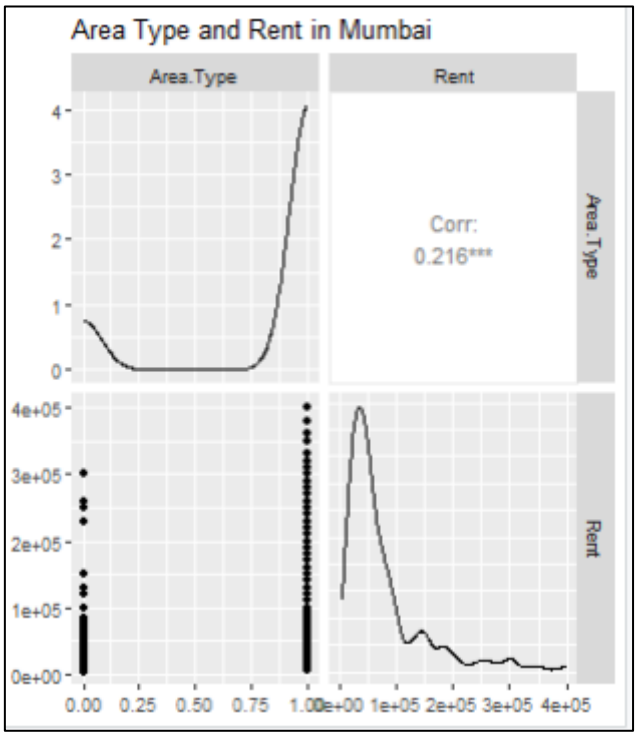
1.2.6 Bathroom and Rent

```
ggpairs(city_subsets_preprocessed$Mumbai, title = "Bathroom and Rent in Mumbai",columns = c("Bathroom", "Rent"))#0.740
```



1.2.7 Area Type and Rent

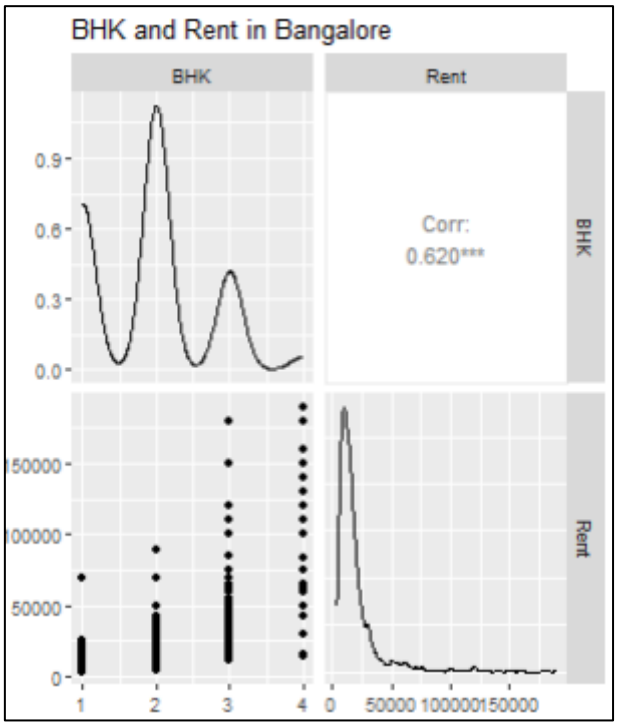
```
ggpairs(city_subsets_preprocessed$Mumbai, title = "Area Type and Rent in Mumbai",columns = c("Area.Type", "Rent"))#0.216
```



1.3 Bangalore City

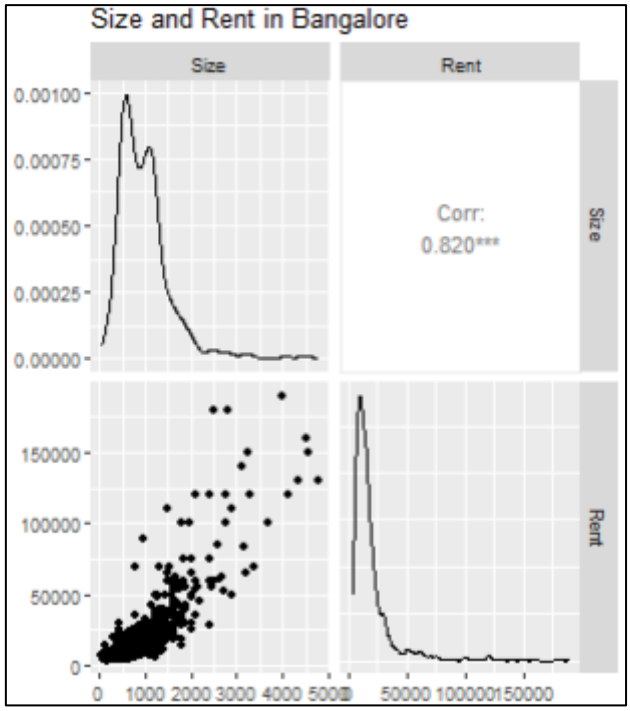
1.3.1 BHK and Rent

```
ggpairs(city_subsets_preprocessed$Bangalore, title = "BHK and Rent in Bangalore",columns = c("BHK", "Rent"))#0.620
```



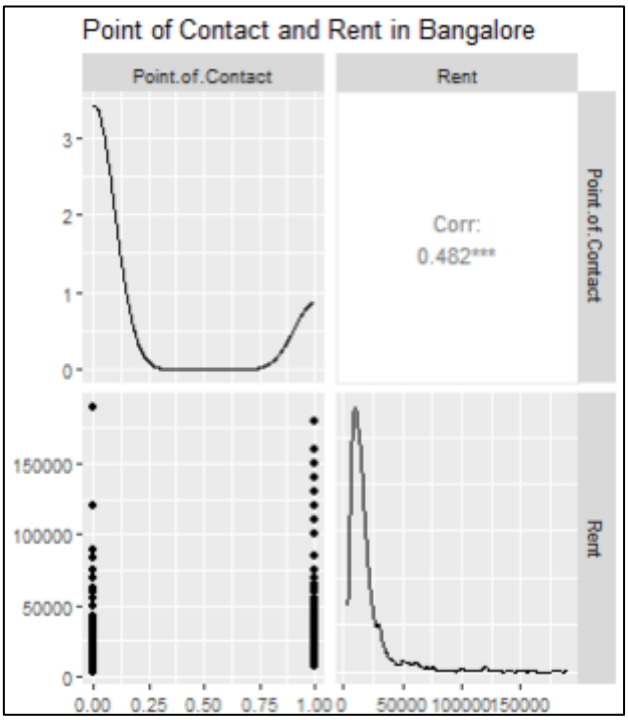
1.3.2 Size and Rent

```
ggpairs(city_subsets_preprocessed$Bangalore, title = "Size and Rent in Bangalore",columns = c("Size", "Rent"))#0.820
```



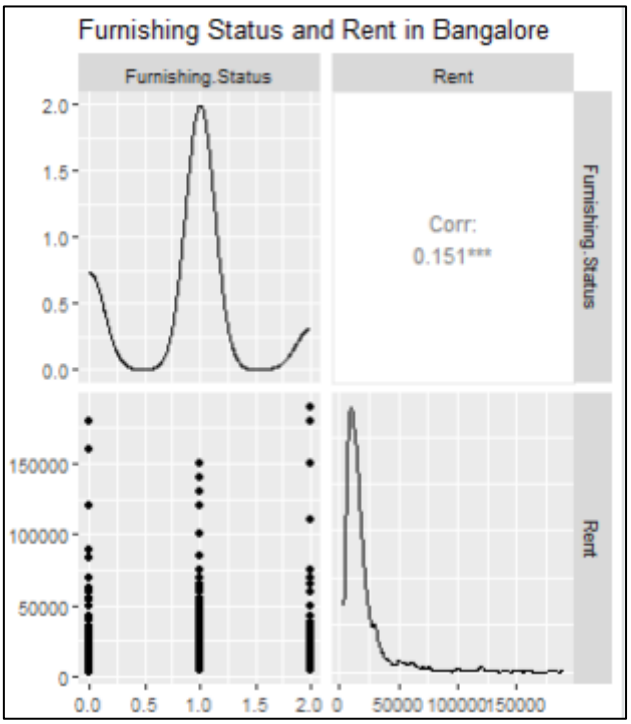
1.3.3 Point of Contact and Rent

```
ggpairs(city_subsets_preprocessed$Bangalore, title = "Point of Contact and Rent in Bangalore",columns = c("Point.of.Contact", "Rent"))#0.482
```



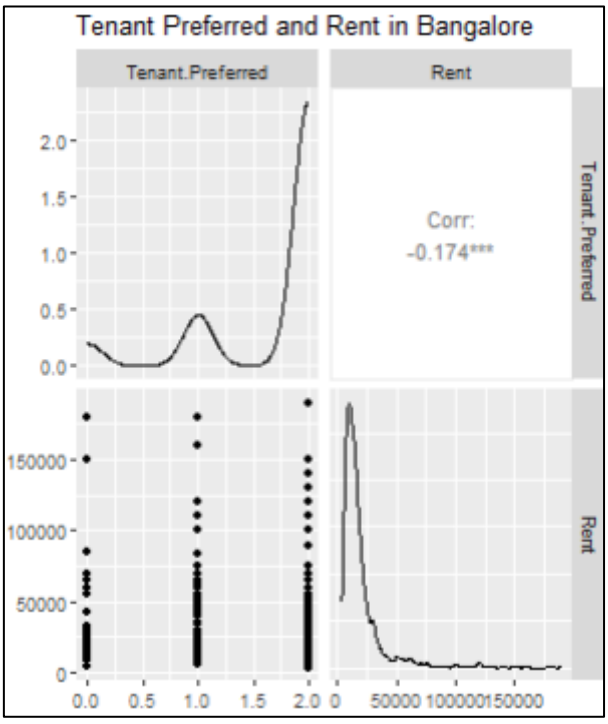
1.3.4 Furnishing Status and Rent

```
ggpairs(city_subsets_preprocessed$Bangalore, title = "Furnishing Status and Rent in Bangalore",columns = c("Furnishing.Status", "Rent"))#0.151
```



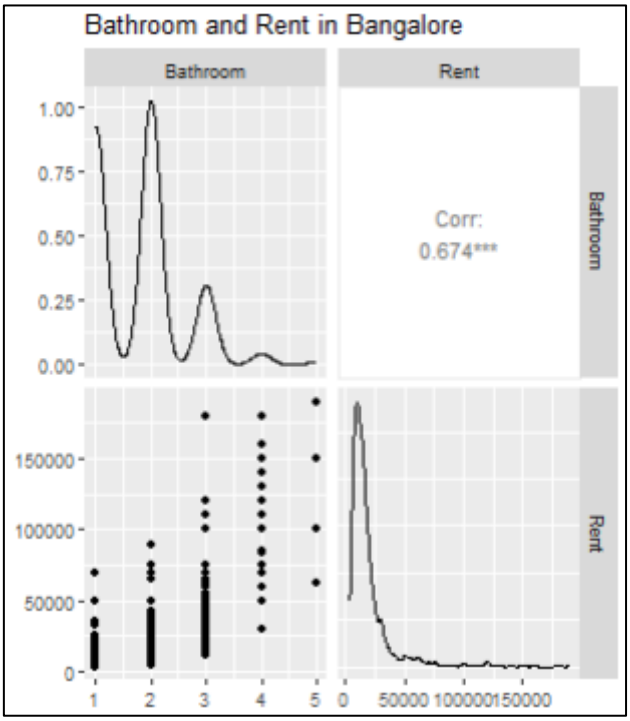
1.3.5 Tenant Preferred and Rent

```
ggpairs(city_subsets_preprocessed$Bangalore, title = "Tenant Preferred and Rent in Bangalore",columns = c("Tenant.Preferred", "Rent"))#-0.174
```



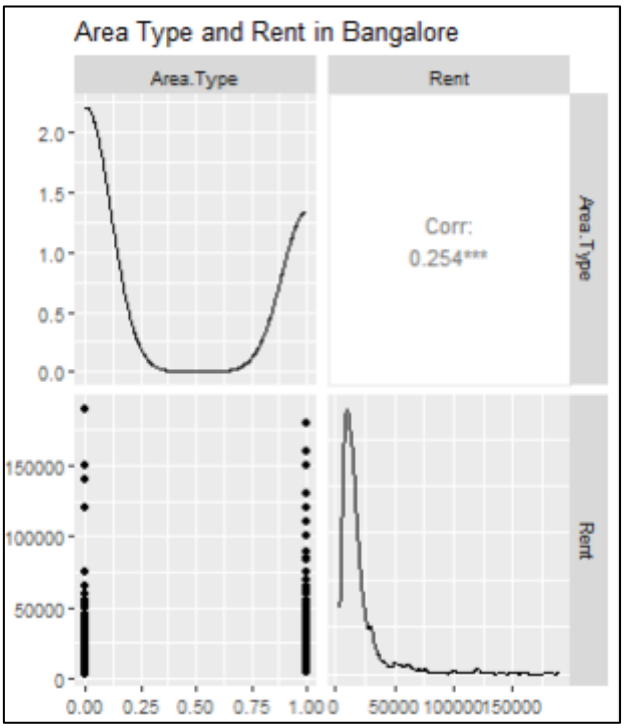
1.3.6 Bathroom and Rent

```
ggpairs(city_subsets_preprocessed$Bangalore, title = "Bathroom and Rent in Bangalore",columns = c("Bathroom", "Rent"))#0.674
```



1.3.7 Area Type and Rent

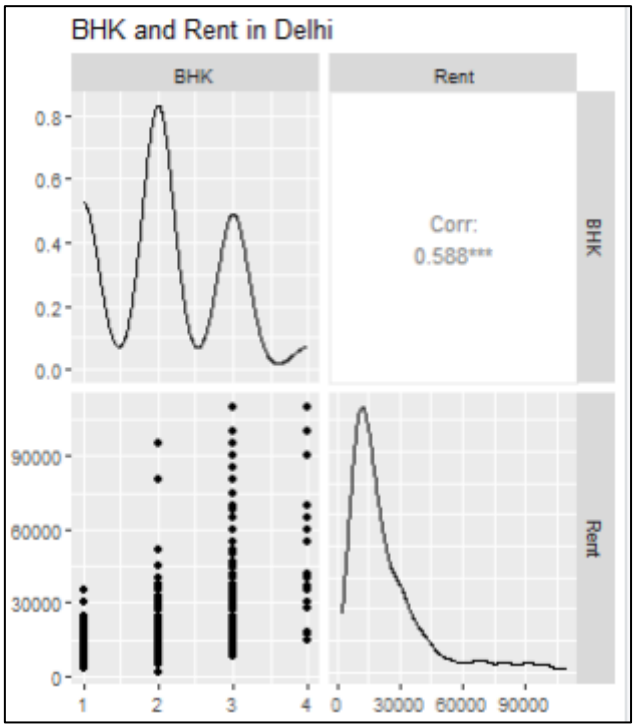
```
ggpairs(city_subsets_preprocessed$Bangalore, title = "Area Type and Rent in Bangalore",columns = c("Area.Type", "Rent"))#0.254
```



1.4 Delhi City

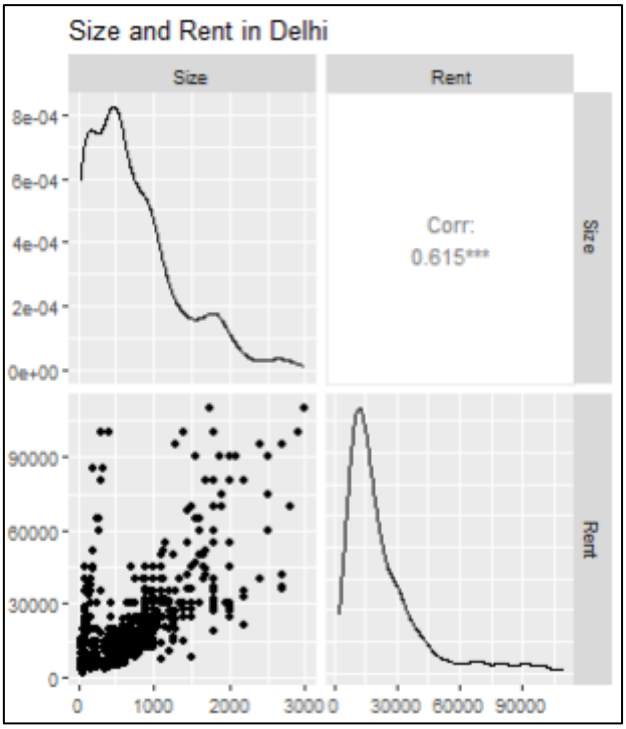
1.4.1 BHK and Rent

```
ggpairs(city_subsets_preprocessed$Delhi, title = "BHK and Rent in Delhi",columns = c("BHK", "Rent"))#0.588
```



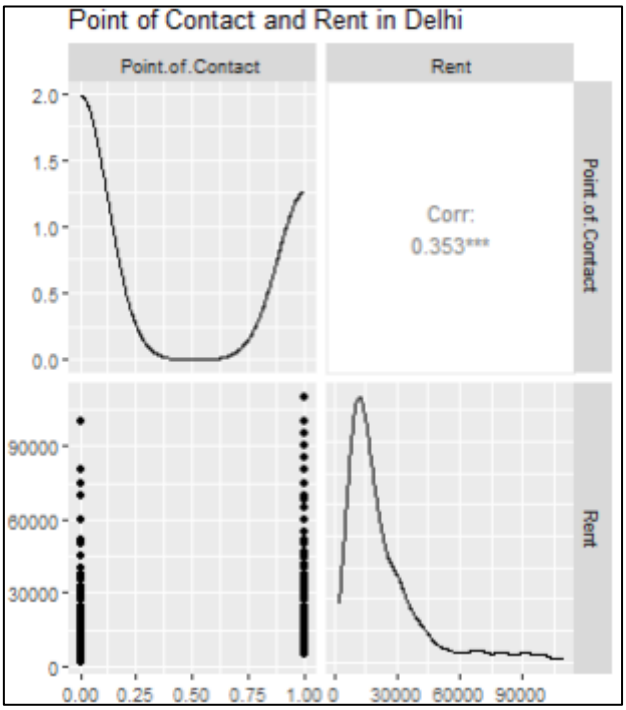
1.4.2 Size and Rent

```
ggpairs(city_subsets_preprocessed$Delhi, title = "Size and Rent in Delhi",columns = c("Size", "Rent"))#0.615
```



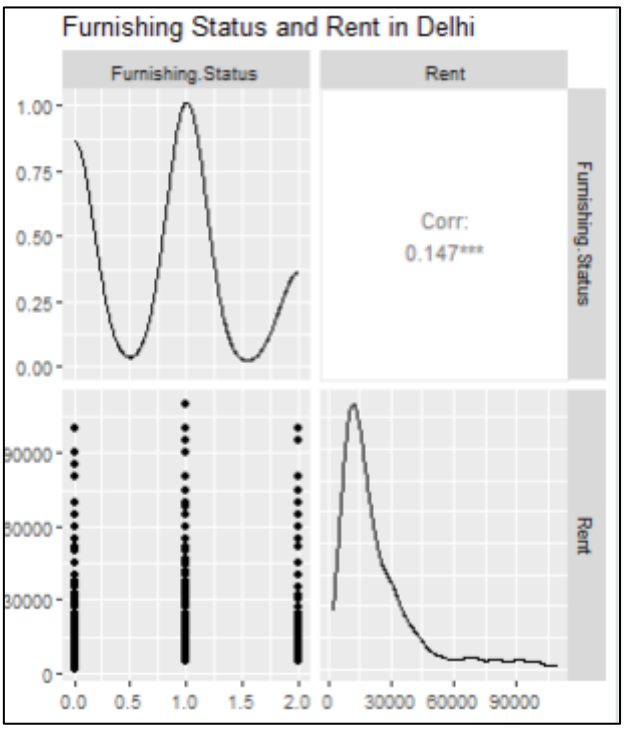
1.4.3 Point of Contact and Rent

```
ggpairs(city_subsets_preprocessed$Delhi, title = "Point of Contact and Rent in Delhi",columns = c("Point.of.Contact", "Rent"))#0.353
```



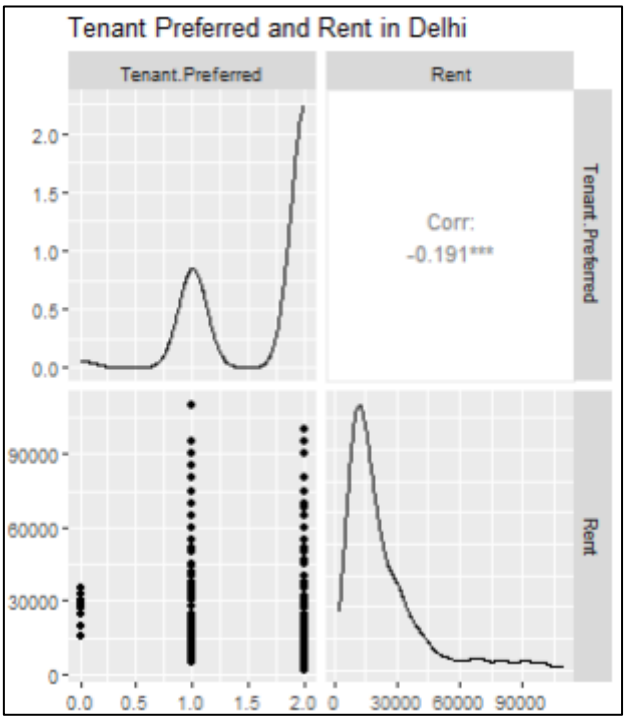
1.4.4 Furnishing Status and Rent

```
ggpairs(city_subsets_preprocessed$Delhi, title = "Furnishing Status and Rent in Delhi",columns = c("Furnishing.Status", "Rent"))#0.147
```



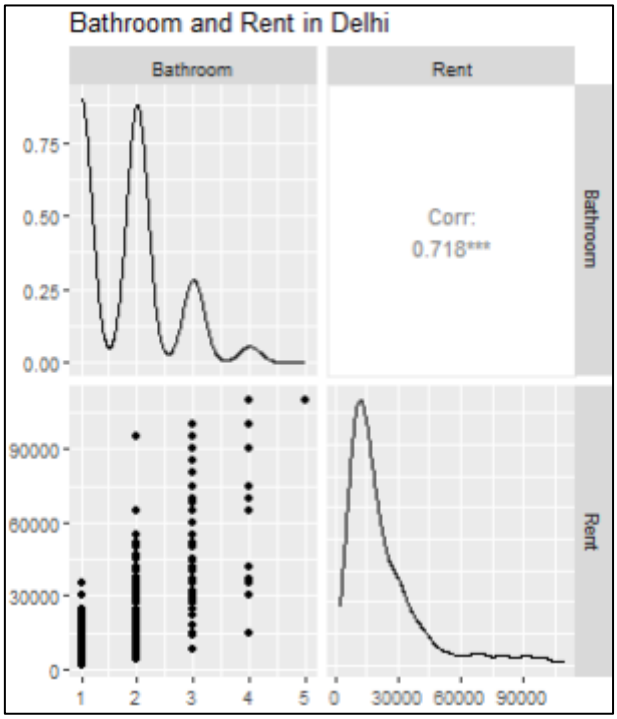
1.4.5 Tenant Preferred and Rent

```
ggpairs(city_subsets_preprocessed$Delhi, title = "Tenant Preferred and Rent in Delhi",columns = c("Tenant.Preferred", "Rent"))#-0.191
```



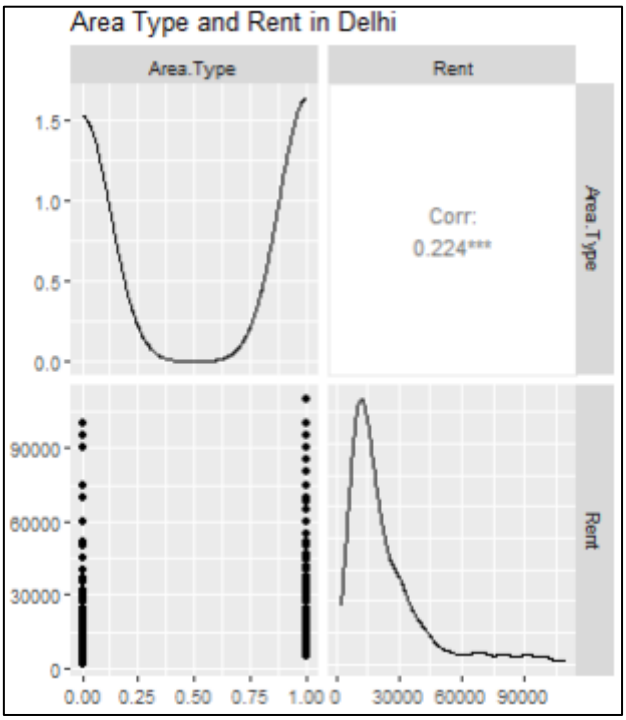
1.4.6 Bathroom and Rent

```
ggpairs(city_subsets_preprocessed$Delhi, title = "Bathroom and Rent in Delhi",columns = c("Bathroom", "Rent"))#0.718
```



1.4.7 Area Type and Rent

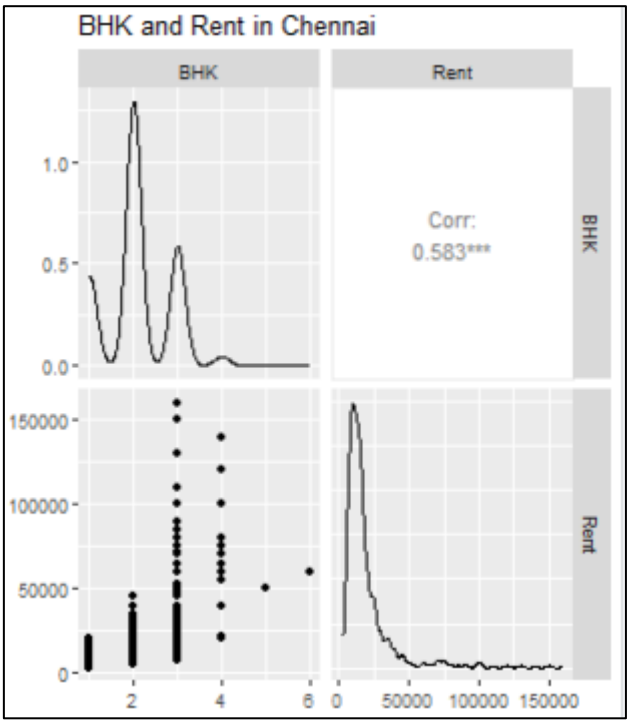
```
ggpairs(city_subsets_preprocessed$Delhi, title = "Area Type and Rent in Delhi",columns = c("Area.Type", "Rent"))#0.224
```



1.5 Chennai City

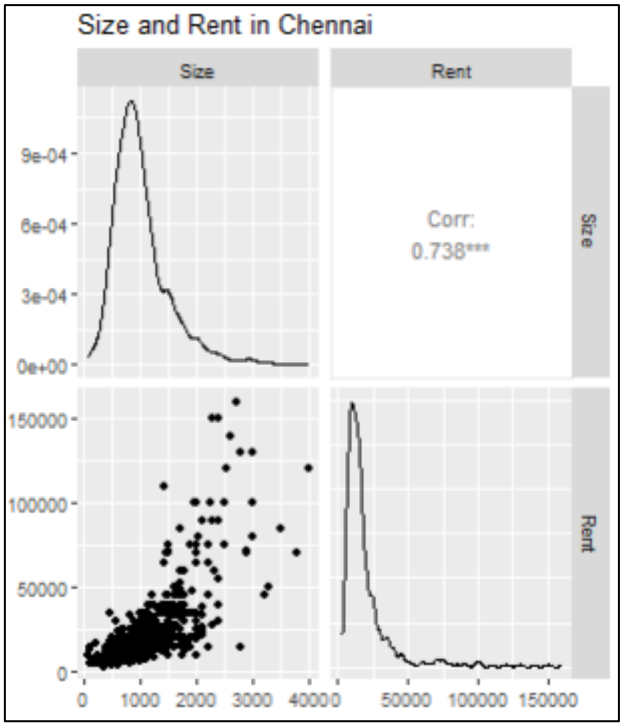
1.5.1 BHK and Rent

```
ggpairs(city_subsets_preprocessed$Chennai, title = "BHK and Rent in Chennai",columns = c("BHK", "Rent"))#0.583
```



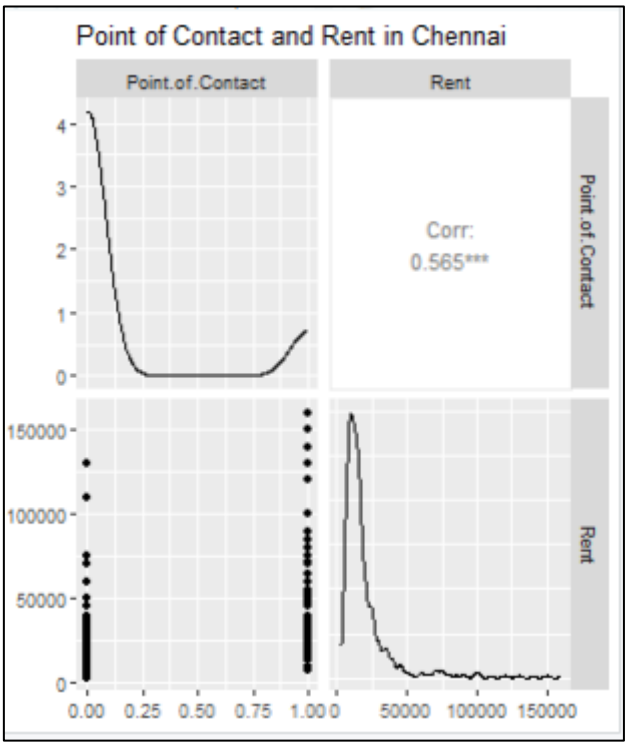
1.5.2 Size and Rent

```
ggpairs(city_subsets_preprocessed$Chennai, title = "Size and Rent in Chennai",columns = c("Size", "Rent"))#0.738
```



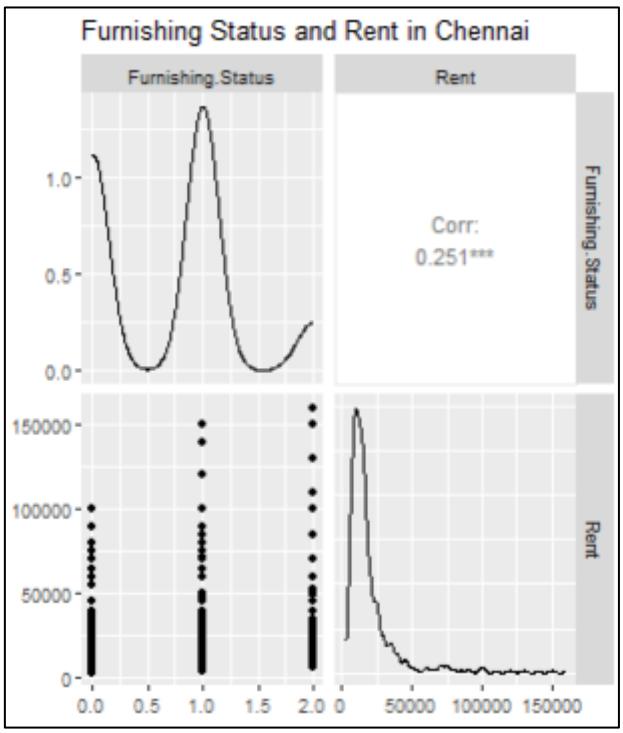
1.5.3 Point of Contact and Rent

```
ggpairs(city_subsets_preprocessed$Chennai, title = "Point of Contact and Rent in Chennai",columns = c("Point.of.Contact", "Rent"))#0.565
```



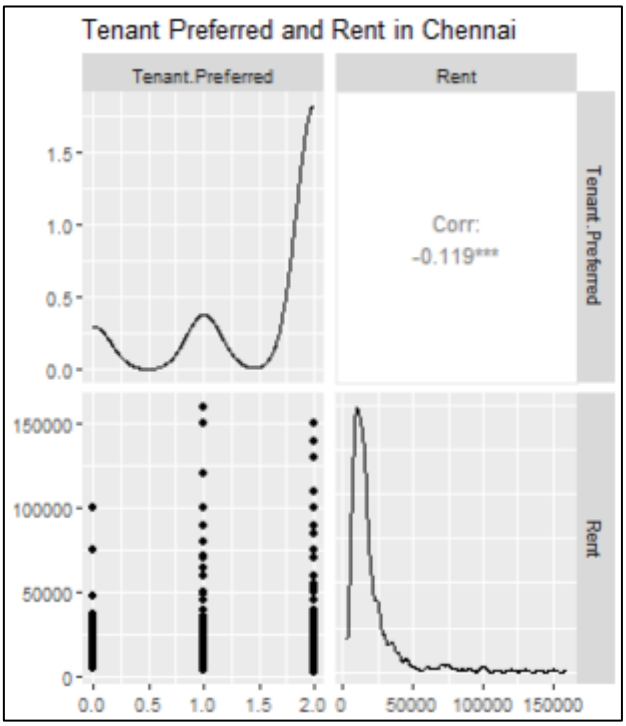
1.5.4 Furnishing Status and Rent

```
ggpairs(city_subsets_preprocessed$Chennai, title = "Furnishing Status and Rent in Chennai",columns = c("Furnishing.Status", "Rent"))#0.251
```



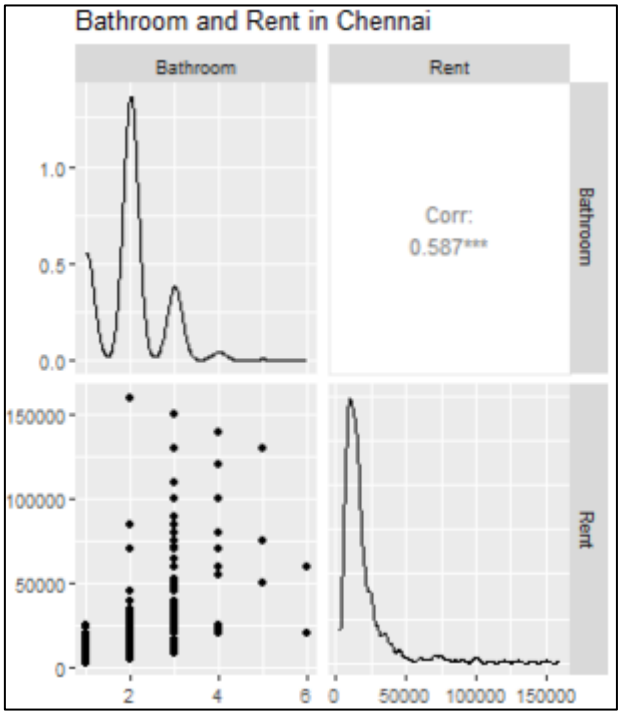
1.5.5 Tenant Preferred and Rent

```
ggpairs(city_subsets_preprocessed$Chennai, title = "Tenant Preferred and Rent in Chennai",columns = c("Tenant.Preferred", "Rent"))#-0.119
```



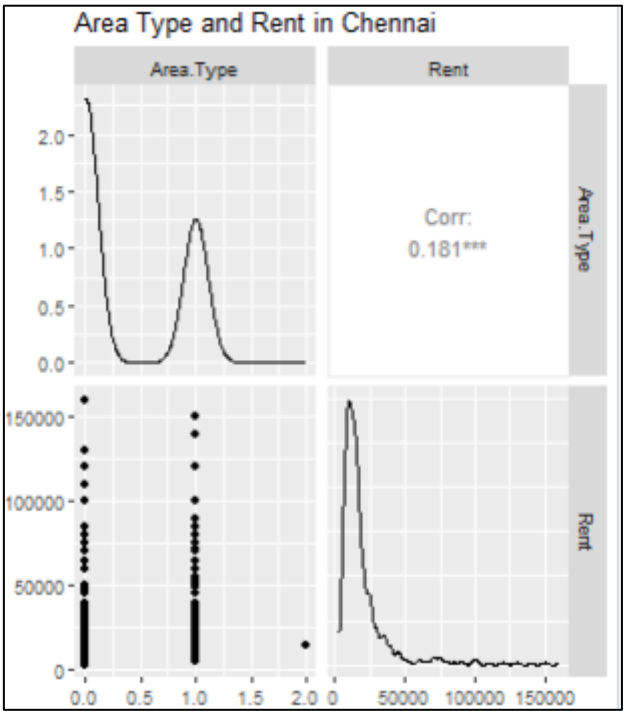
1.5.6 Bathroom and Rent

```
ggpairs(city_subsets_preprocessed$Chennai, title = "Bathroom and Rent in Chennai",columns = c("Bathroom", "Rent"))#0.587
```



1.5.7 Area Type and Rent

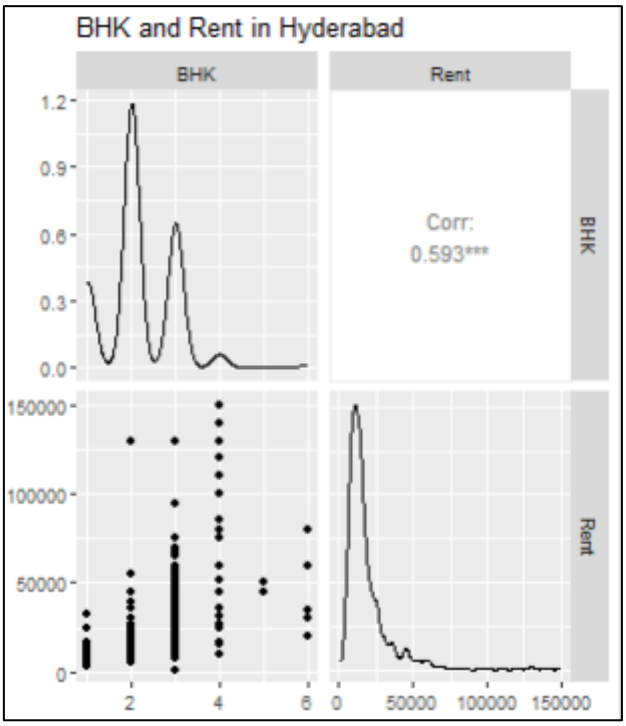
```
ggpairs(city_subsets_preprocessed$Chennai, title = "Area Type and Rent in Chennai",columns = c("Area.Type", "Rent"))#0.181
```



1.6 Hyderabad City

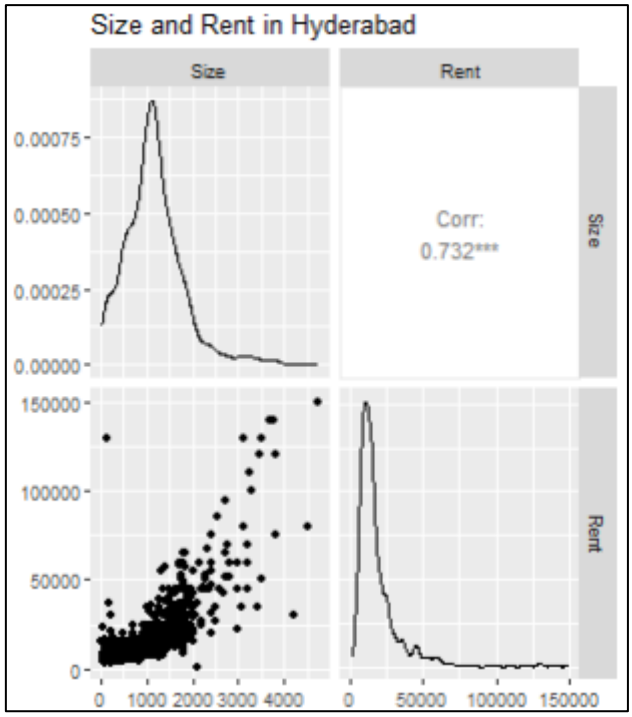
1.6.1 BHK and Rent

```
ggpairs(city_subsets_preprocessed$Hyderabad, title = "BHK and Rent in Hyderabad",columns = c("BHK", "Rent"))#0.593
```



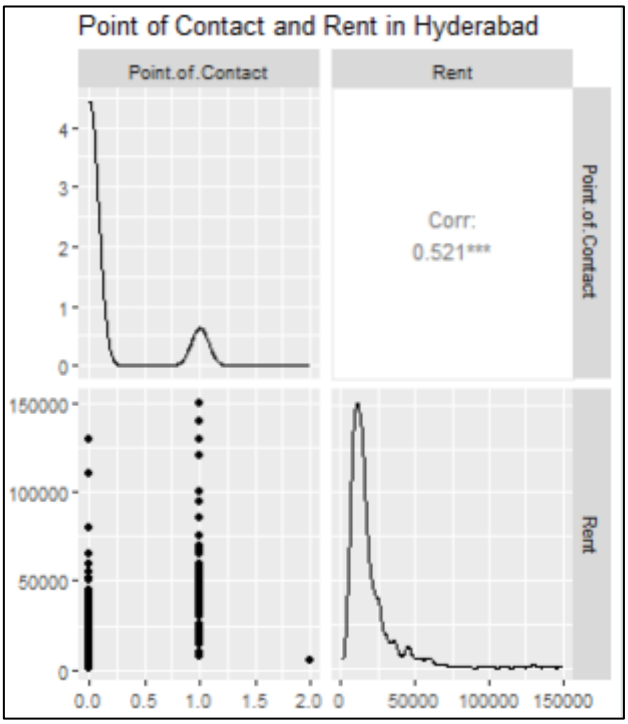
1.6.2 Size and Rent

```
ggpairs(city_subsets_preprocessed$Hyderabad, title = "Size and Rent in Hyderabad",columns = c("Size", "Rent"))#0.732
```



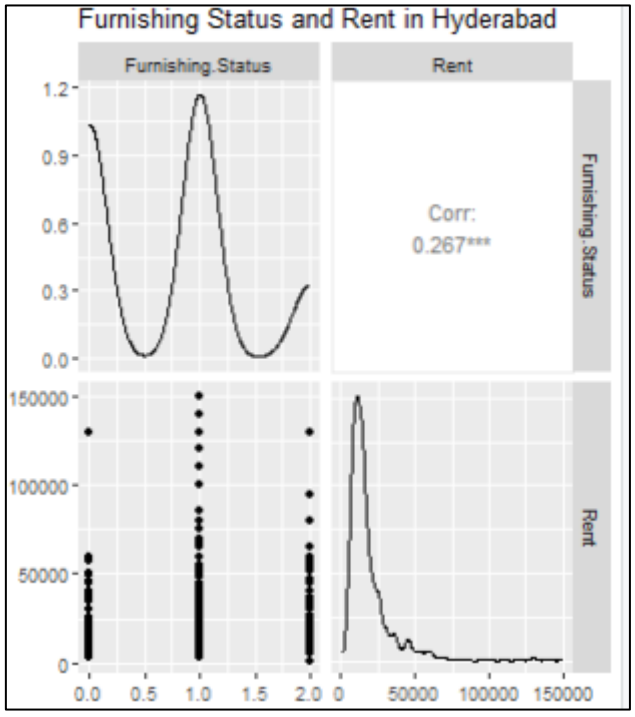
1.6.3 Point of Contact and Rent

```
ggpairs(city_subsets_preprocessed$Hyderabad, title = "Point of Contact and Rent in Hyderabad",columns = c("Point.of.Contact", "Rent"))#0.521
```



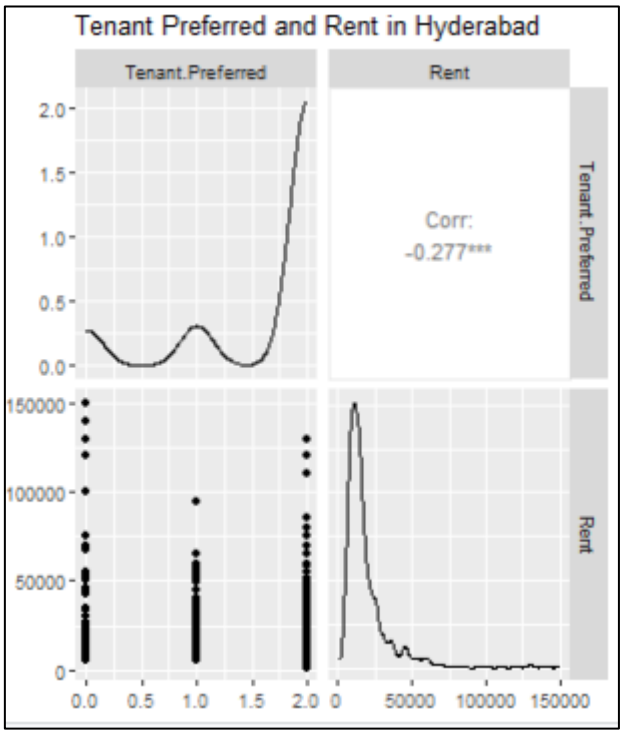
1.6.4 Furnishing Status and Rent

```
ggpairs(city_subsets_preprocessed$Hyderabad, title = "Furnishing Status and Rent in Hyderabad",columns = c("Furnishing.Status", "Rent"))#0.267
```



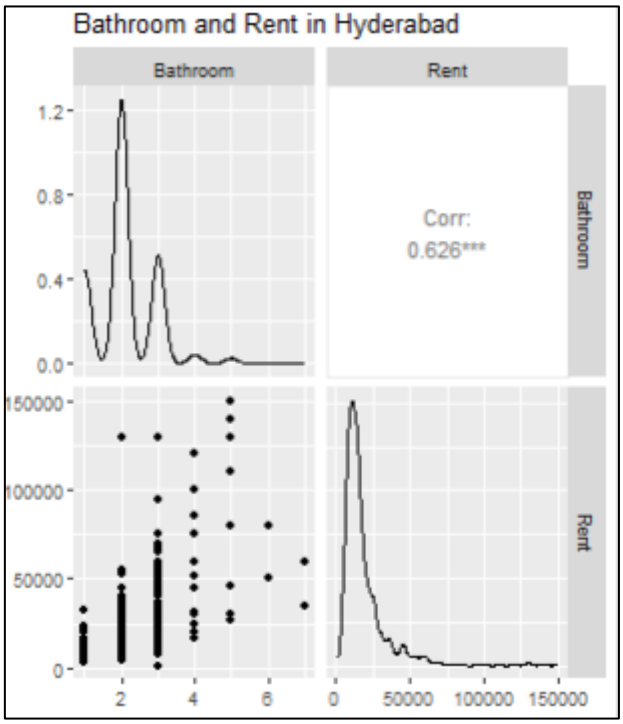
1.6.5 Tenant Preferred and Rent

```
ggpairs(city_subsets_preprocessed$Hyderabad, title = "Tenant Preferred and Rent in Hyderabad",columns = c("Tenant.Preferred", "Rent"))#-0.227
```



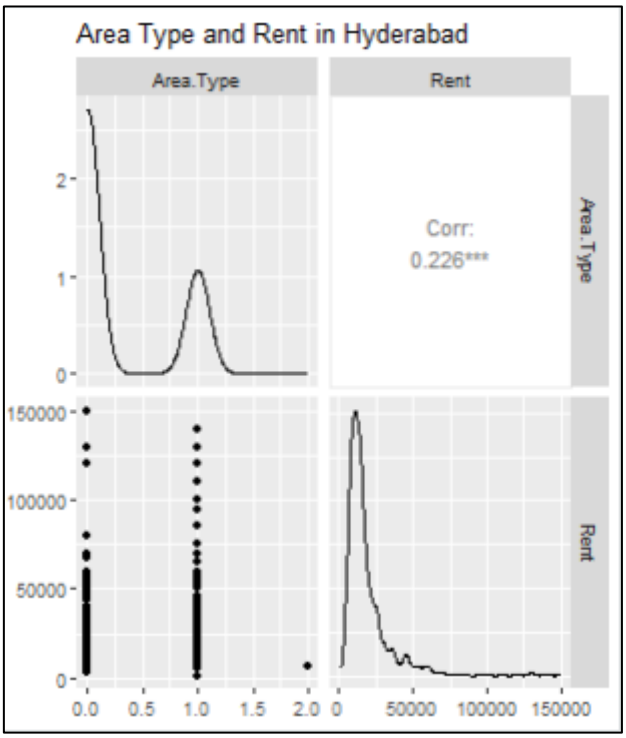
1.6.6 Bathroom and Rent

```
ggpairs(city_subsets_preprocessed$Hyderabad, title = "Bathroom and Rent in Hyderabad",columns = c("Bathroom", "Rent"))#0.626
```



1.6.7 Area Type and Rent

```
ggpairs(city_subsets_preprocessed$Hyderabad, title = "Area Type and Rent in Hyderabad",columns = c("Area.Type", "Rent"))#0.226
```



2.0 ANOVA Testing

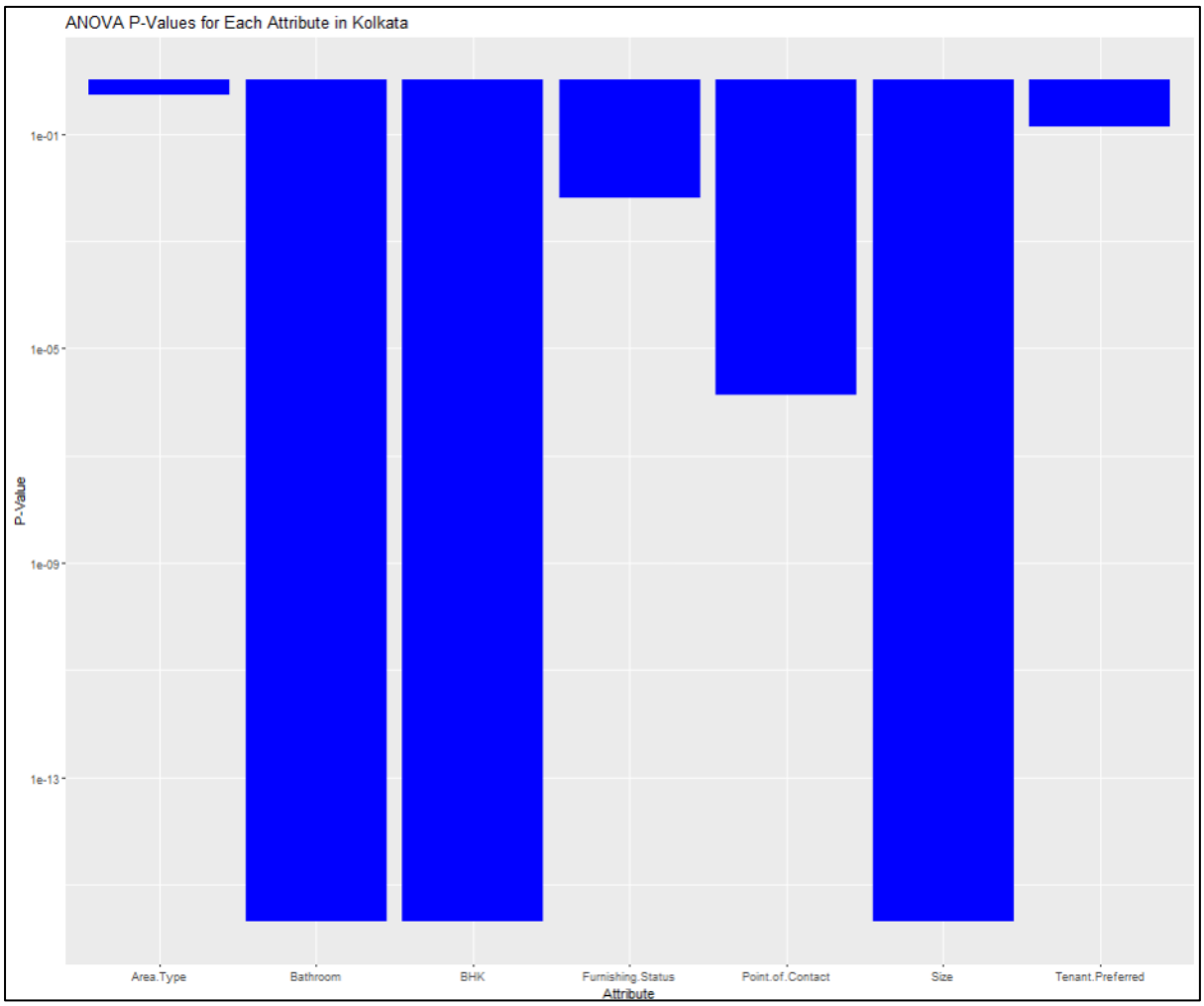
2.1 Kolkata City

```
#ANOVA for Kolkata
#Result for p-value in Kolkata
summary(aov(BHK~Rent, data = HouseRent_Cleaned_Preprocessed_Kolkata))#2*10^(-16)
summary(aov(Size~Rent, data = HouseRent_Cleaned_Preprocessed_Kolkata))#2*10^(-16)
summary(aov(Point.of.Contact~Rent, data = HouseRent_Cleaned_Preprocessed_Kolkata))#1.38*10^(-6)
summary(aov(Furnishing.Status~Rent, data = HouseRent_Cleaned_Preprocessed_Kolkata))#0.00642
summary(aov(Tenant.Preferred~Rent, data = HouseRent_Cleaned_Preprocessed_Kolkata))#0.137
summary(aov(Bathroom~Rent, data = HouseRent_Cleaned_Preprocessed_Kolkata))#2*10^(-16)
summary(aov(Area.Type~Rent, data = HouseRent_Cleaned_Preprocessed_Kolkata))# 0.545
```

```
> summary(aov(BHK~Rent, data = HouseRent_Cleaned_Preprocessed_Kolkata))#2*10^(-16)
             Df Sum Sq Mean Sq F value Pr(>F)
Rent          1  93.28   93.28   228.9 <2e-16 ***
Residuals   517 210.70    0.41
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> summary(aov(Size~Rent, data = HouseRent_Cleaned_Preprocessed_Kolkata))#2*10^(-16)
             Df   Sum Sq  Mean Sq F value Pr(>F)
Rent          1 26015967 26015967   296.9 <2e-16 ***
Residuals   517 45306460    87633
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> summary(aov(Point.of.Contact~Rent, data = HouseRent_Cleaned_Preprocessed_Kolkata))#1.38*10^(-6)
             Df Sum Sq Mean Sq F value   Pr(>F)
Rent          1   2.96  2.9559   23.87 1.38e-06 ***
Residuals   517  64.02  0.1238
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> summary(aov(Furnishing.Status~Rent, data = HouseRent_Cleaned_Preprocessed_Kolkata))#0.00642
             Df Sum Sq Mean Sq F value  Pr(>F)
Rent          1   3.57   3.566   7.489 0.00642 **
Residuals   517 246.18   0.476
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> summary(aov(Tenant.Preferred~Rent, data = HouseRent_Cleaned_Preprocessed_Kolkata))#0.137
             Df Sum Sq Mean Sq F value Pr(>F)
Rent          1   0.68  0.6793   2.215  0.137
Residuals   517 158.58  0.3067
> summary(aov(Bathroom~Rent, data = HouseRent_Cleaned_Preprocessed_Kolkata))#2*10^(-16)
             Df Sum Sq Mean Sq F value Pr(>F)
Rent          1  51.25   51.25   238.3 <2e-16 ***
Residuals   517 111.18    0.22
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> summary(aov(Area.Type~Rent, data = HouseRent_Cleaned_Preprocessed_Kolkata))# 0.545
             Df Sum Sq Mean Sq F value Pr(>F)
Rent          1   0.09 0.09188   0.366  0.545
Residuals   517 129.65 0.25078
```

Above shows the ANOVA computation results, and we are only focusing on the Pr(>F) column, which pointed to the p-value. For better understanding, we also make a bar chart for clear view.

```
ANOVA_attribute <- c("BHK", "Size", "Point.of.Contact", "Furnishing.Status", "Tenant.Preferred", "Bathroom", "Area.Type")
ANOVA_p_scores_Kolkata <- c(2 * 10^(-16), 2 * 10^(-16), 1.38 * 10^(-6), 0.00642, 0.137, 2 * 10^(-16), 0.545)
ANOVA_p_scores_df_Kolkata <- data.frame(Attribute = ANOVA_attribute, P_Value = ANOVA_p_scores_Kolkata)

ggplot(ANOVA_p_scores_df_Kolkata, aes(x = Attribute, y = P_Value)) +
  geom_bar(stat = "identity", fill = "blue") +
  labs(title = "ANOVA P-Values for Each Attribute in Kolkata",
       x = "Attribute", y = "P-Value") +
  scale_y_log10()
```



Summary of the ANOVA p-value in Kolkata City

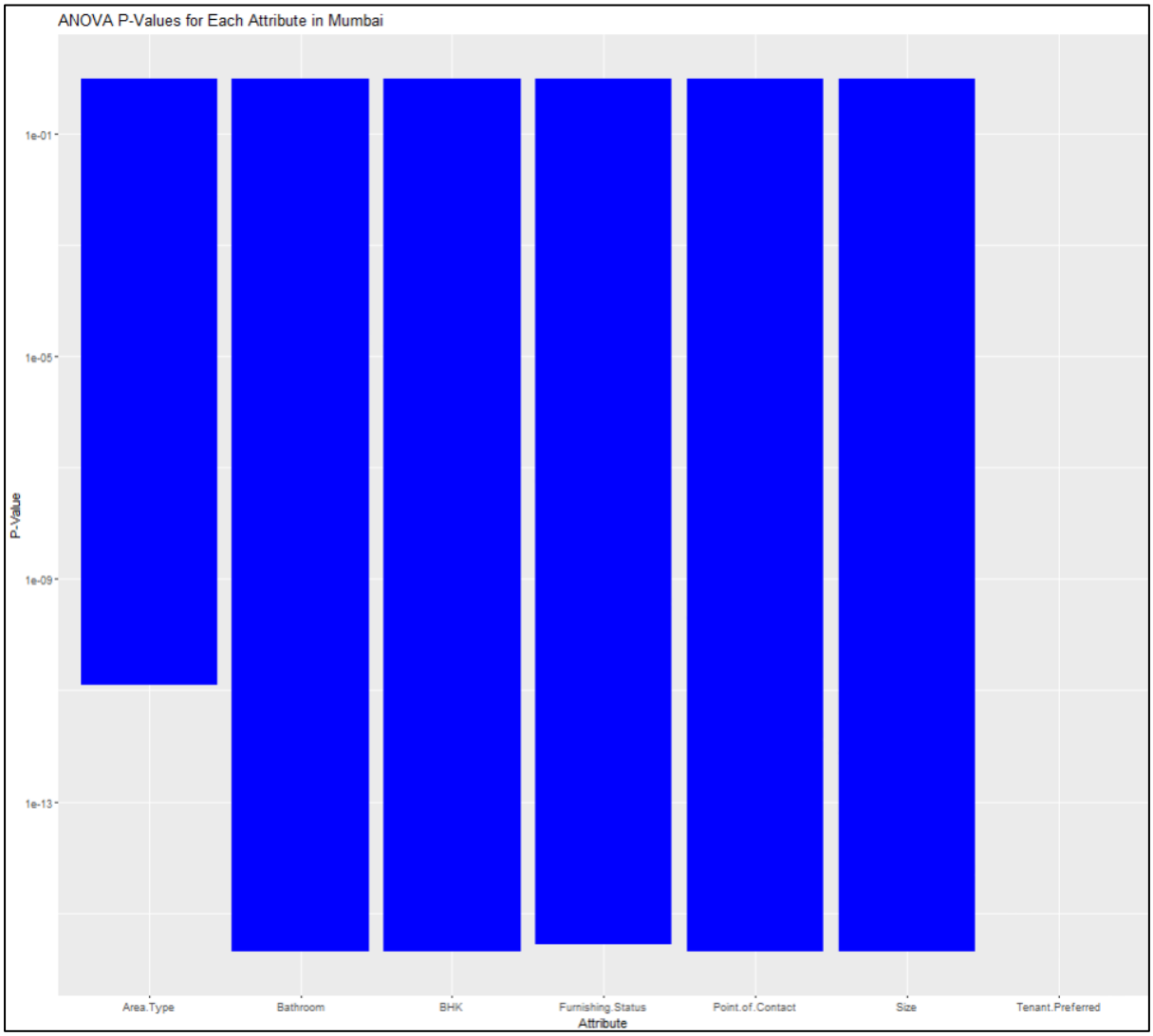
Attribute	P-value	Impact?
BHK	<0.001	High
Size	<0.001	High
Bathroom	<0.001	High
Point of Contact	<0.001	High
Furnishing Status	<0.01	Upper Medium
Tenant Preferred	>0.05	Low
Area Type	>0.05	Low

2.2 Mumbai City

```
summary(aov(BHK~Rent, data = HouseRent_Cleaned_Preprocessed_Mumbai))#2*10^(-16)
summary(aov(Size~Rent, data = HouseRent_Cleaned_Preprocessed_Mumbai))#2*10^(-16)
summary(aov(Point.of.Contact~Rent, data = HouseRent_Cleaned_Preprocessed_Mumbai))#2*10^(-16)
summary(aov(Furnishing.Status~Rent, data = HouseRent_Cleaned_Preprocessed_Mumbai))#2.75*10^(-16)
summary(aov(Tenant.Preferred~Rent, data = HouseRent_Cleaned_Preprocessed_Mumbai))#0.98
summary(aov(Bathroom~Rent, data = HouseRent_Cleaned_Preprocessed_Mumbai))#2*10^(-16)
summary(aov(Area.Type~Rent, data = HouseRent_Cleaned_Preprocessed_Mumbai))#1.26*10^(-11)
```

```
> summary(aov(BHK~Rent, data = HouseRent_Cleaned_Preprocessed_Mumbai))#2*10^(-16)
             Df Sum Sq Mean Sq F value Pr(>F)
Rent          1  438.1   438.1    1112 <2e-16 ***
Residuals   957  377.0     0.4
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> summary(aov(Size~Rent, data = HouseRent_Cleaned_Preprocessed_Mumbai))#2*10^(-16)
             Df    Sum Sq   Mean Sq F value Pr(>F)
Rent          1 167519179 167519179    2719 <2e-16 ***
Residuals   957  58967688     61617
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> summary(aov(Point.of.Contact~Rent, data = HouseRent_Cleaned_Preprocessed_Mumbai))#2*10^(-16)
             Df Sum Sq Mean Sq F value Pr(>F)
Rent          1   13.5  13.502   92.25 <2e-16 ***
Residuals   957  140.1   0.146
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> summary(aov(Furnishing.Status~Rent, data = HouseRent_Cleaned_Preprocessed_Mumbai))#2.75*10^(-16)
             Df Sum Sq Mean Sq F value   Pr(>F)
Rent          1   37.8   37.76   69.41 2.75e-16 ***
Residuals   957  520.6    0.54
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> summary(aov(Tenant.Preferred~Rent, data = HouseRent_Cleaned_Preprocessed_Mumbai))#0.98
             Df Sum Sq Mean Sq F value Pr(>F)
Rent          1    0.0  0.0004   0.001   0.98
Residuals   957  604.1  0.6312
> summary(aov(Bathroom~Rent, data = HouseRent_Cleaned_Preprocessed_Mumbai))#2*10^(-16)
             Df Sum Sq Mean Sq F value Pr(>F)
Rent          1  494.9   494.9    1155 <2e-16 ***
Residuals   957  410.0     0.4
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> summary(aov(Area.Type~Rent, data = HouseRent_Cleaned_Preprocessed_Mumbai))#1.26*10^(-11)
             Df Sum Sq Mean Sq F value   Pr(>F)
Rent          1   5.93   5.926   47.02 1.26e-11 ***
Residuals   957 120.61   0.126
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
>
```

```
ANOVA_p_scores_Mumbai <- c(2 * 10^(-16), 2 * 10^(-16), 2 * 10^(-16), 2.75 * 10^(-16), 0.98, 2 * 10^(-16), 1.26 * 10^(-11))
ANOVA_p_scores_df_Mumbai <- data.frame(Attribute = ANOVA_attribute, P_Value = ANOVA_p_scores_Mumbai)

ggplot(ANOVA_p_scores_df_Mumbai, aes(x = Attribute, y = P_Value)) +
  geom_bar(stat = "identity", fill = "blue") +
  labs(title = "ANOVA P-Values for Each Attribute in Mumbai",
       x = "Attribute", y = "P-Value") +
  scale_y_log10()
```



Summary of the ANOVA p-value in Mumbai City

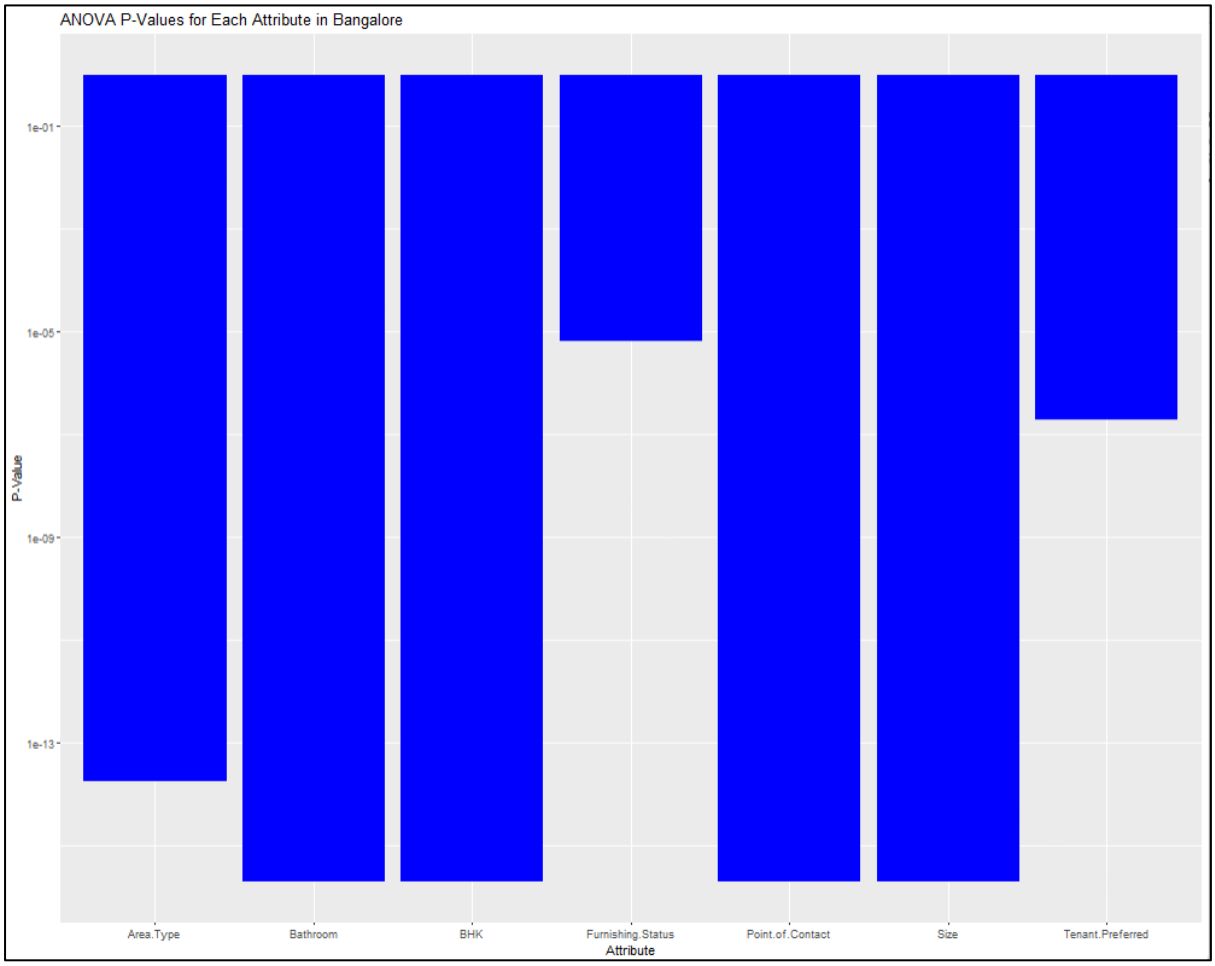
Attribute	P-value	Impact?
BHK	<0.001	High
Size	<0.001	High
Bathroom	<0.001	High
Point of Contact	<0.001	High
Furnishing Status	<0.001	High
Area Type	<0.001	High
Tenant Preferred	>0.05	Low

2.3 Bangalore City

```
summary(aov(BHK~Rent, data = HouseRent_Cleaned_Preprocessed_Bangalore))#2*10^(-16)
summary(aov(Size~Rent, data = HouseRent_Cleaned_Preprocessed_Bangalore))#2*10^(-16)
summary(aov(Point.of.Contact~Rent, data = HouseRent_Cleaned_Preprocessed_Bangalore))#2*10^(-16)
summary(aov(Furnishing.Status~Rent, data = HouseRent_Cleaned_Preprocessed_Bangalore))#6.42*10^(-6)
summary(aov(Tenant.Preferred~Rent, data = HouseRent_Cleaned_Preprocessed_Bangalore))#1.95*10^(-7)
summary(aov(Bathroom~Rent, data = HouseRent_Cleaned_Preprocessed_Bangalore))#2*10^(-16)
summary(aov(Area.Type~Rent, data = HouseRent_Cleaned_Preprocessed_Bangalore))#1.78*10^(-14)
```

```
> #ANOVA for Bangalore
> #Result for p-value in Bangalore
> summary(aov(BHK~Rent, data = HouseRent_Cleaned_Preprocessed_Bangalore))#2*10^(-16)
             Df Sum Sq Mean Sq F value Pr(>F)
Rent          1  195.8  195.76   550.4 <2e-16 ***
Residuals   880  313.0    0.36
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> summary(aov(Size~Rent, data = HouseRent_Cleaned_Preprocessed_Bangalore))#2*10^(-16)
             Df    Sum Sq   Mean Sq F value Pr(>F)
Rent          1 204892932 204892932    1809 <2e-16 ***
Residuals   880  99668858    113260
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> summary(aov(Point.of.Contact~Rent, data = HouseRent_Cleaned_Preprocessed_Bangalore))#2*10^(-16)
             Df Sum Sq Mean Sq F value Pr(>F)
Rent          1  33.01   33.01   266.3 <2e-16 ***
Residuals   880 109.07    0.12
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> summary(aov(Furnishing.Status~Rent, data = HouseRent_Cleaned_Preprocessed_Bangalore))#6.42*10^(-6)
             Df Sum Sq Mean Sq F value   Pr(>F)
Rent          1   6.51   6.507   20.61 6.42e-06 ***
Residuals   880 277.89   0.316
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> summary(aov(Tenant.Preferred~Rent, data = HouseRent_Cleaned_Preprocessed_Bangalore))#1.95*10^(-7)
             Df Sum Sq Mean Sq F value   Pr(>F)
Rent          1   8.85   8.848   27.52 1.95e-07 ***
Residuals   880 282.98   0.322
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> summary(aov(Bathroom~Rent, data = HouseRent_Cleaned_Preprocessed_Bangalore))#2*10^(-16)
             Df Sum Sq Mean Sq F value Pr(>F)
Rent          1  240.4  240.42     733 <2e-16 ***
Residuals   880  288.6    0.33
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> summary(aov(Area.Type~Rent, data = HouseRent_Cleaned_Preprocessed_Bangalore))#1.78*10^(-14)
             Df Sum Sq Mean Sq F value   Pr(>F)
Rent          1  13.38   13.38   60.81 1.78e-14 ***
Residuals   880 193.65    0.22
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
ANOVA_p_scores_Bangalore <- c(2 * 10^(-16), 2 * 10^(-16), 2 * 10^(-16), 6.42*10^(-6), 1.95*10^(-7), 2*10^(-16), 1.78*10^(-14))
ANOVA_p_scores_df_Bangalore <- data.frame(Attribute = ANOVA_attribute, P_Value = ANOVA_p_scores_Bangalore)

ggplot(ANOVA_p_scores_df_Bangalore, aes(x = Attribute, y = P_Value)) +
  geom_bar(stat = "identity", fill = "blue") +
  labs(title = "ANOVA P-Values for Each Attribute in Bangalore",
       x = "Attribute", y = "P-Value") +
  scale_y_log10()
```



Summary of the ANOVA p-value in Bangalore City

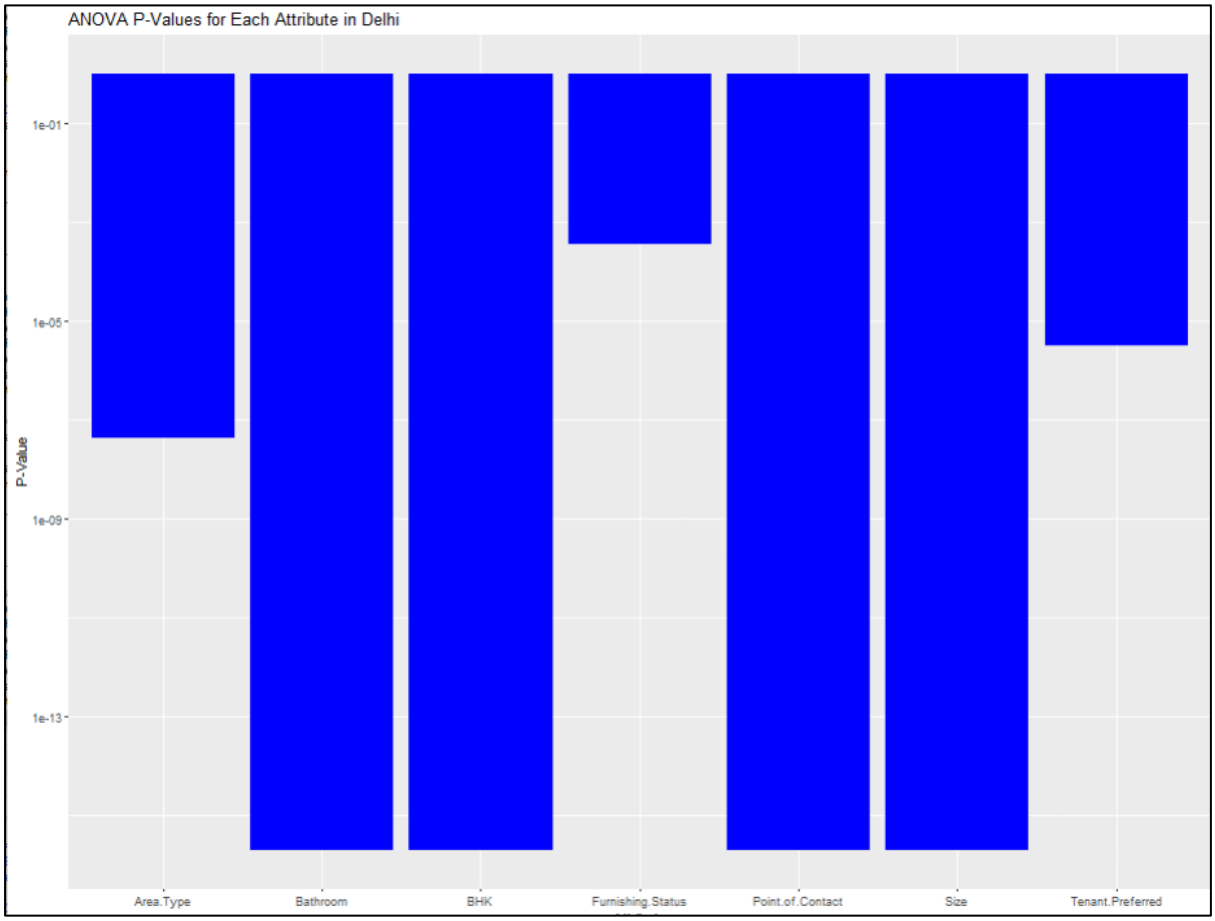
Attribute	P-value	Impact?
BHK	<0.001	High
Size	<0.001	High
Bathroom	<0.001	High
Point of Contact	<0.001	High
Furnishing Status	<0.001	High
Area Type	<0.001	High
Tenant Preferred	<0.001	High

2.4 Delhi City

```
summary(aov(BHK~Rent, data = HouseRent_Cleaned_Preprocessed_Delhi))#2*10^(-16)
summary(aov(Size~Rent, data = HouseRent_Cleaned_Preprocessed_Delhi))#2*10^(-16)
summary(aov(Point.of.Contact~Rent, data = HouseRent_Cleaned_Preprocessed_Delhi))#2*10^(-16)
summary(aov(Furnishing.Status~Rent, data = HouseRent_Cleaned_Preprocessed_Delhi))#0.00035
summary(aov(Tenant.Preferred~Rent, data = HouseRent_Cleaned_Preprocessed_Delhi))#3.22*10^(-6)
summary(aov(Bathroom~Rent, data = HouseRent_Cleaned_Preprocessed_Delhi))#2*10^(-16)
summary(aov(Area.Type~Rent, data = HouseRent_Cleaned_Preprocessed_Delhi))#4.29*10^(-8)
```

```
> #-------------------------------------------------------------------------------------------------------------
> #ANOVA for Delhi
> #Result for p-value in Delhi
> summary(aov(BHK~Rent, data = HouseRent_Cleaned_Preprocessed_Delhi))#2*10^(-16)
             Df Sum Sq Mean Sq F value Pr(>F)
Rent          1  136.5  136.49   307.1 <2e-16 ***
Residuals   582  258.6    0.44
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> summary(aov(Size~Rent, data = HouseRent_Cleaned_Preprocessed_Delhi))#2*10^(-16)
             Df    Sum Sq  Mean Sq F value Pr(>F)
Rent          1  79511137 79511137   354.9 <2e-16 ***
Residuals   582 130388432   224035
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> summary(aov(Point.of.Contact~Rent, data = HouseRent_Cleaned_Preprocessed_Delhi))#2*10^(-16)
             Df Sum Sq Mean Sq F value Pr(>F)
Rent          1  17.34  17.340   83.11 <2e-16 ***
Residuals   582 121.43   0.209
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> summary(aov(Furnishing.Status~Rent, data = HouseRent_Cleaned_Preprocessed_Delhi))#0.00035
             Df Sum Sq Mean Sq F value  Pr(>F)
Rent          1   6.33   6.327   12.93 0.00035 ***
Residuals   582 284.73   0.489
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> summary(aov(Tenant.Preferred~Rent, data = HouseRent_Cleaned_Preprocessed_Delhi))#3.22*10^(-6)
             Df Sum Sq Mean Sq F value   Pr(>F)
Rent          1   5.33   5.334   22.11 3.22e-06 ***
Residuals   582 140.41   0.241
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> summary(aov(Bathroom~Rent, data = HouseRent_Cleaned_Preprocessed_Delhi))#2*10^(-16)
             Df Sum Sq Mean Sq F value Pr(>F)
Rent          1  187.2   187.2   619.9 <2e-16 ***
Residuals   582  175.7     0.3
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> summary(aov(Area.Type~Rent, data = HouseRent_Cleaned_Preprocessed_Delhi))#4.29*10^(-8)
             Df Sum Sq Mean Sq F value   Pr(>F)
Rent          1   7.33   7.334   30.83 4.29e-08 ***
Residuals   582 138.46   0.238
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
ANOVA_p_scores_Delhi <- c(2 * 10^(-16), 2 * 10^(-16), 2 * 10^(-16), 0.00035, 3.22*10^(-6), 2*10^(-16), 4.29*10^(-8))
ANOVA_p_scores_df_Delhi <- data.frame(Attribute = ANOVA_attribute, P_Value = ANOVA_p_scores_Delhi)

ggplot(ANOVA_p_scores_df_Delhi, aes(x = Attribute, y = P_Value)) +
  geom_bar(stat = "identity", fill = "blue") +
  labs(title = "ANOVA P-Values for Each Attribute in Delhi",
       x = "Attribute", y = "P-Value") +
  scale_y_log10()
```



Summary of the ANOVA p-value in Delhi City

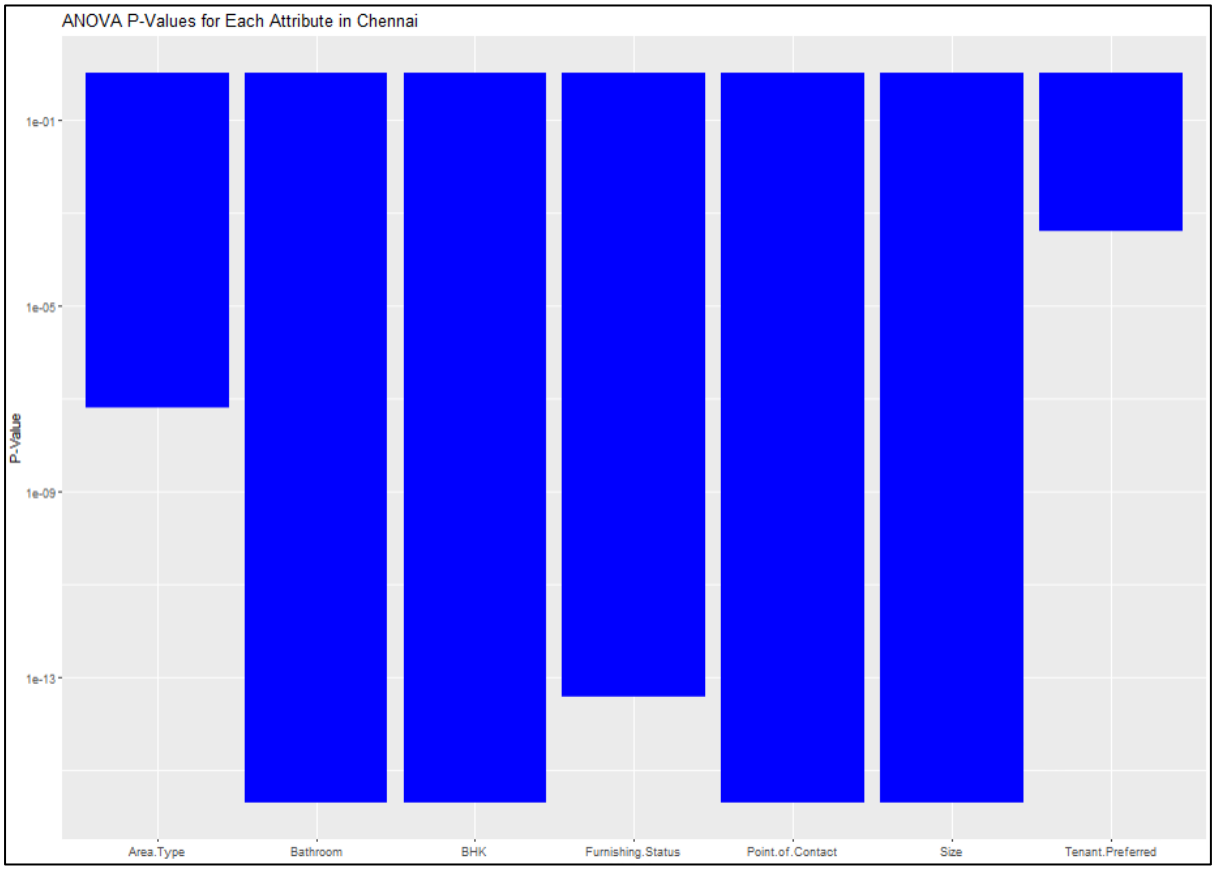
Attribute	P-value	Impact?
BHK	<0.001	High
Size	<0.001	High
Bathroom	<0.001	High
Point of Contact	<0.001	High
Furnishing Status	<0.001	High
Area Type	<0.001	High
Tenant Preferred	<0.001	High

2.5 Chennai City

```
summary(aov(BHK~Rent, data = HouseRent_Cleaned_Preprocessed_Chennai))#2*10^(-16)
summary(aov(Size~Rent, data = HouseRent_Cleaned_Preprocessed_Chennai))#2*10^(-16)
summary(aov(Point.of.Contact~Rent, data = HouseRent_Cleaned_Preprocessed_Chennai))#2*10^(-16)
summary(aov(Furnishing.Status~Rent, data = HouseRent_Cleaned_Preprocessed_Chennai))#3.83*10^(-14)
summary(aov(Tenant.Preferred~Rent, data = HouseRent_Cleaned_Preprocessed_Chennai))#0.000389
summary(aov(Bathroom~Rent, data = HouseRent_Cleaned_Preprocessed_Chennai))#2*10^(-16)
summary(aov(Area.Type~Rent, data = HouseRent_Cleaned_Preprocessed_Chennai))#6.39*10^(-8)
```

```
> #-------------------------------------------------------------------------------------------------------------
> #ANOVA for Chennai
> #Result for p-value in Chennai
> summary(aov(BHK~Rent, data = HouseRent_Cleaned_Preprocessed_Chennai))#2*10^(-16)
             Df Sum Sq Mean Sq F value Pr(>F)
Rent          1  157.6  157.62   453.9 <2e-16 ***
Residuals   881  305.9    0.35
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> summary(aov(Size~Rent, data = HouseRent_Cleaned_Preprocessed_Chennai))#2*10^(-16)
             Df    Sum Sq   Mean Sq F value Pr(>F)
Rent          1 125616765 125616765    1054 <2e-16 ***
Residuals   881 105000234    119183
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> summary(aov(Point.of.Contact~Rent, data = HouseRent_Cleaned_Preprocessed_Chennai))#2*10^(-16)
             Df Sum Sq Mean Sq F value Pr(>F)
Rent          1  34.71   34.71     413 <2e-16 ***
Residuals   881  74.03    0.08
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> summary(aov(Furnishing.Status~Rent, data = HouseRent_Cleaned_Preprocessed_Chennai))#3.83*10^(-14)
             Df Sum Sq Mean Sq F value   Pr(>F)
Rent          1   22.1  22.134   59.19 3.83e-14 ***
Residuals   881  329.4   0.374
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> summary(aov(Tenant.Preferred~Rent, data = HouseRent_Cleaned_Preprocessed_Chennai))#0.000389
             Df Sum Sq Mean Sq F value   Pr(>F)
Rent          1    5.9   5.928   12.68 0.000389 ***
Residuals   881  411.8   0.467
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> summary(aov(Bathroom~Rent, data = HouseRent_Cleaned_Preprocessed_Chennai))#2*10^(-16)
             Df Sum Sq Mean Sq F value Pr(>F)
Rent          1  163.0  163.04   463.8 <2e-16 ***
Residuals   881  309.7    0.35
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> summary(aov(Area.Type~Rent, data = HouseRent_Cleaned_Preprocessed_Chennai))#6.39*10^(-8)
             Df Sum Sq Mean Sq F value   Pr(>F)
Rent          1   6.66   6.656   29.75 6.39e-08 ***
Residuals   881 197.10   0.224
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
ANOVA_p_scores_Chennai <- c(2 * 10^(-16), 2 * 10^(-16), 2 * 10^(-16), 3.83*10^(-14), 0.000389, 2 * 10^(-16), 6.39*10^(-8))
ANOVA_p_scores_df_Chennai <- data.frame(Attribute = ANOVA_attribute, P_Value = ANOVA_p_scores_Chennai)

ggplot(ANOVA_p_scores_df_Chennai, aes(x = Attribute, y = P_Value)) +
  geom_bar(stat = "identity", fill = "blue") +
  labs(title = "ANOVA P-Values for Each Attribute in Chennai",
       x = "Attribute", y = "P-Value") +
  scale_y_log10()
```



Summary of the ANOVA p-value in Chennai City

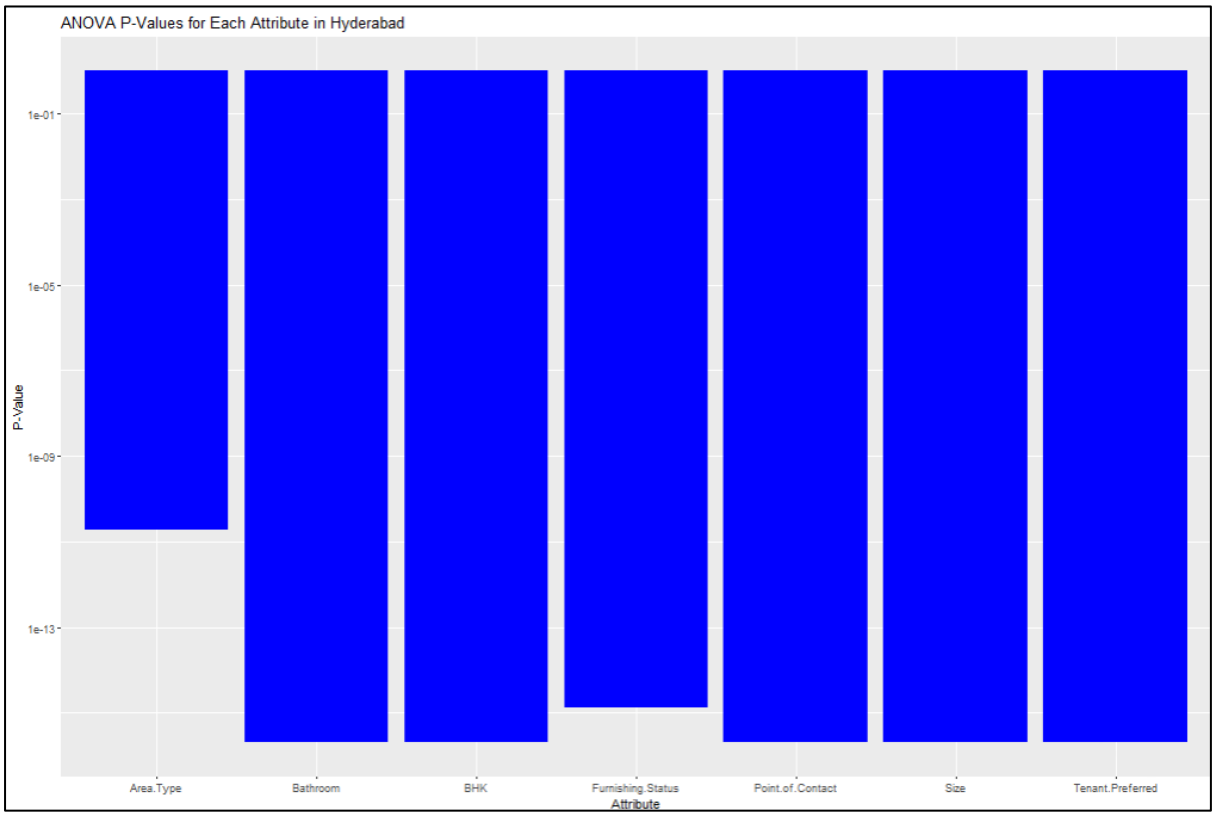
Attribute	P-value	Impact?
BHK	<0.001	High
Size	<0.001	High
Bathroom	<0.001	High
Point of Contact	<0.001	High
Furnishing Status	<0.001	High
Area Type	<0.001	High
Tenant Preferred	<0.001	High

2.6 Hyderabad City

```
summary(aov(BHK~Rent, data = HouseRent_Cleaned_Preprocessed_Hyderabad))#2*10^(-16)
summary(aov(Size~Rent, data = HouseRent_Cleaned_Preprocessed_Hyderabad))#2*10^(-16)
summary(aov(Point.of.Contact~Rent, data = HouseRent_Cleaned_Preprocessed_Hyderabad))#2*10^(-16)
summary(aov(Furnishing.Status~Rent, data = HouseRent_Cleaned_Preprocessed_Hyderabad))#1.34*10^(-15)
summary(aov(Tenant.Preferred~Rent, data = HouseRent_Cleaned_Preprocessed_Hyderabad))#2*10^(-16)
summary(aov(Bathroom~Rent, data = HouseRent_Cleaned_Preprocessed_Hyderabad))#2*10^(-16)
summary(aov(Area.Type~Rent, data = HouseRent_Cleaned_Preprocessed_Hyderabad))#1.85*10^(-11)
```

```
> #-----------------------------------------------------------------------------------------------------------
> #ANOVA for Hyderabad
> #Result for p-value in Hyderabad
> summary(aov(BHK~Rent, data = HouseRent_Cleaned_Preprocessed_Hyderabad))#2*10^(-16)
             Df Sum Sq Mean Sq F value Pr(>F)
Rent          1  190.8  190.76   466.1 <2e-16 ***
Residuals   861  352.4    0.41
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> summary(aov(Size~Rent, data = HouseRent_Cleaned_Preprocessed_Hyderabad))#2*10^(-16)
             Df    Sum Sq   Mean Sq F value Pr(>F)
Rent          1 197253218 197253218   995.3 <2e-16 ***
Residuals   861 170633734    198181
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> summary(aov(Point.of.Contact~Rent, data = HouseRent_Cleaned_Preprocessed_Hyderabad))#2*10^(-16)
             Df Sum Sq Mean Sq F value Pr(>F)
Rent          1  26.55  26.548     320 <2e-16 ***
Residuals   861  71.43   0.083
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> summary(aov(Furnishing.Status~Rent, data = HouseRent_Cleaned_Preprocessed_Hyderabad))#1.34*10^(-15)
             Df Sum Sq Mean Sq F value   Pr(>F)
Rent          1   28.3  28.334   66.33 1.34e-15 ***
Residuals   861  367.8   0.427
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> summary(aov(Tenant.Preferred~Rent, data = HouseRent_Cleaned_Preprocessed_Hyderabad))#2*10^(-16)
             Df Sum Sq Mean Sq F value Pr(>F)
Rent          1   28.1  28.051    71.3 <2e-16 ***
Residuals   861  338.8   0.393
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> summary(aov(Bathroom~Rent, data = HouseRent_Cleaned_Preprocessed_Hyderabad))#2*10^(-16)
             Df Sum Sq Mean Sq F value Pr(>F)
Rent          1  229.4  229.37   553.8 <2e-16 ***
Residuals   861  356.6    0.41
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> summary(aov(Area.Type~Rent, data = HouseRent_Cleaned_Preprocessed_Hyderabad))#1.85*10^(-11)
             Df Sum Sq Mean Sq F value   Pr(>F)
Rent          1   9.06   9.065   46.35 1.85e-11 ***
Residuals   861 168.38   0.196
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
>
```

```
ANOVA_p_scores_Hyderabad <- c(2 * 10^(-16), 2 * 10^(-16), 2 * 10^(-16), 1.34*10^(-15), 2*10^(-16), 2 * 10^(-16), 1.85*10^(-11))
ANOVA_p_scores_df_Hyderabad <- data.frame(Attribute = ANOVA_attribute, P_Value = ANOVA_p_scores_Hyderabad)

ggplot(ANOVA_p_scores_df_Hyderabad, aes(x = Attribute, y = P_Value)) +
  geom_bar(stat = "identity", fill = "blue") +
  labs(title = "ANOVA P-Values for Each Attribute in Hyderabad",
       x = "Attribute", y = "P-Value") +
  scale_y_log10()
```



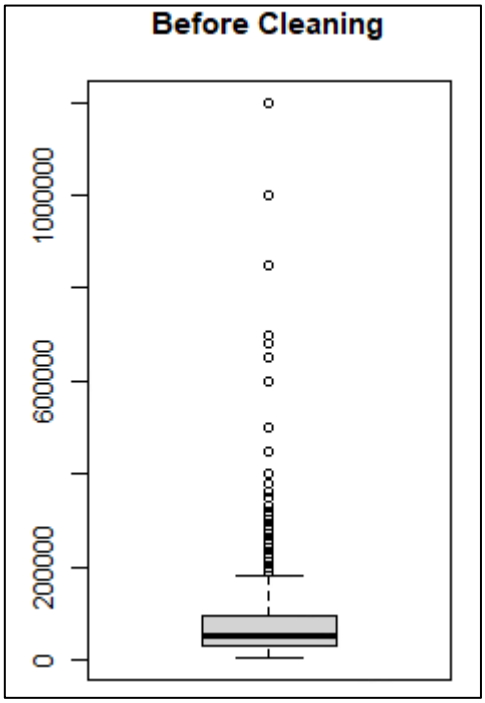
Summary of the ANOVA p-value in Hyderabad City

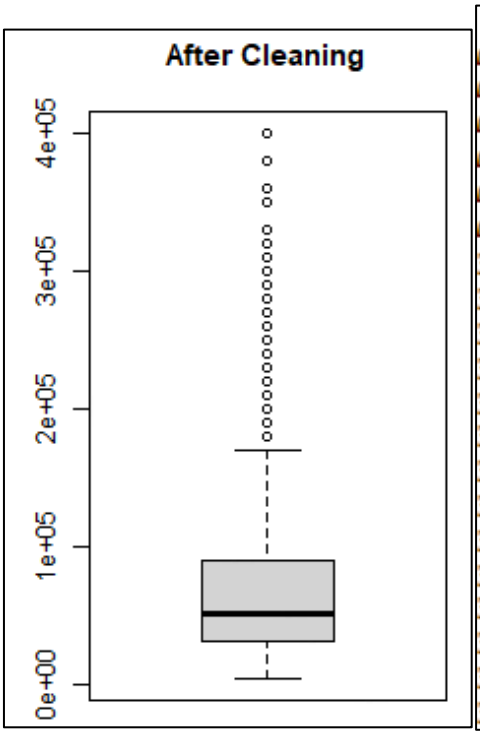
Attribute	P-value	Impact?
BHK	<0.001	High
Size	<0.001	High
Bathroom	<0.001	High
Point of Contact	<0.001	High
Furnishing Status	<0.001	High
Area Type	<0.001	High
Tenant Preferred	<0.001	High

3.0 Data Cleaning Coding

3.1 Rent Cleaning for Mumbai

```
#Cleaning for Mumbai Rent
max(city_subsets$Mumbai$Rent)
city_subsets$Mumbai = city_subsets$Mumbai[city_subsets$Mumbai$Rent !=1200000, ]
city_subsets$Mumbai = city_subsets$Mumbai[city_subsets$Mumbai$Rent !=1000000, ]
city_subsets$Mumbai = city_subsets$Mumbai[city_subsets$Mumbai$Rent !=850000, ]
city_subsets$Mumbai = city_subsets$Mumbai[city_subsets$Mumbai$Rent !=700000, ]
city_subsets$Mumbai = city_subsets$Mumbai[city_subsets$Mumbai$Rent !=680000, ]
city_subsets$Mumbai = city_subsets$Mumbai[city_subsets$Mumbai$Rent !=650000, ]
city_subsets$Mumbai = city_subsets$Mumbai[city_subsets$Mumbai$Rent !=600000, ]
city_subsets$Mumbai = city_subsets$Mumbai[city_subsets$Mumbai$Rent !=500000, ]
city_subsets$Mumbai = city_subsets$Mumbai[city_subsets$Mumbai$Rent !=450000, ]

#Comparison before and after cleaning
boxplot(city_subsets_unclean$Mumbai$Rent)
title("Before Cleaning")
boxplot(city_subsets$Mumbai$Rent)
title("After Cleaning")

head(city_subsets$Mumbai[order(city_subsets$Mumbai$Rent, decreasing = TRUE), ], 20)
```

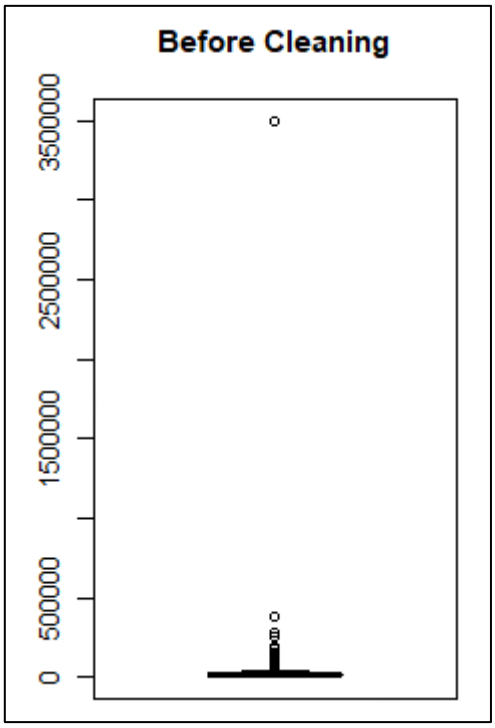


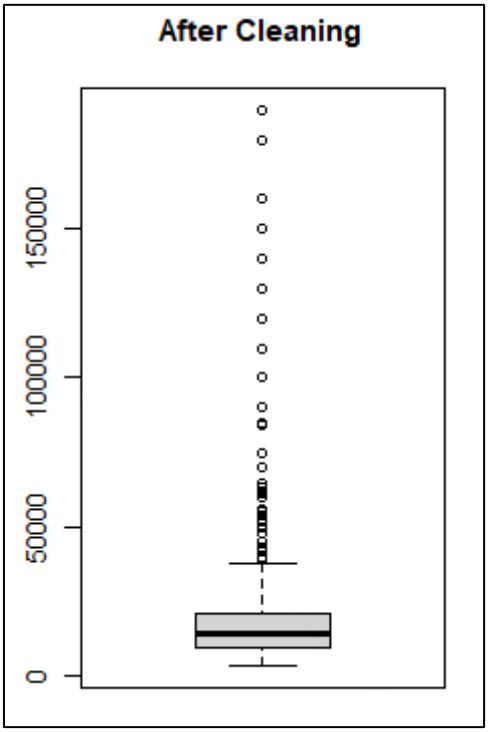



```
     Rent
400000
400000
400000
400000
400000
400000
380000
380000
360000
350000
350000
350000
350000
350000
350000
330000
330000
330000
320000
310000
310000
```

3.2 Rent Cleaning for Bangalore

```
#Cleaning for Bangalore Rent
max(city_subsets$Bangalore$Rent)
city_subsets$Bangalore = city_subsets$Bangalore[city_subsets$Bangalore$Rent !=3500000, ]
city_subsets$Bangalore = city_subsets$Bangalore[city_subsets$Bangalore$Rent !=380000, ]
city_subsets$Bangalore = city_subsets$Bangalore[city_subsets$Bangalore$Rent !=280000, ]
city_subsets$Bangalore = city_subsets$Bangalore[city_subsets$Bangalore$Rent !=250000, ]

#Comparison before and after cleaning
boxplot(city_subsets_unclean$Bangalore$Rent)
title("Before Cleaning")
boxplot(city_subsets$Bangalore$Rent)
title("After Cleaning")

head(city_subsets$Bangalore[order(city_subsets$Bangalore$Rent, decreasing = TRUE), ], 20)
```

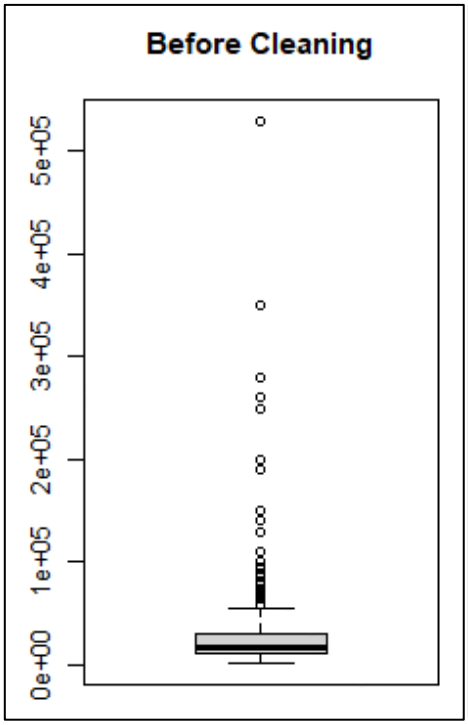


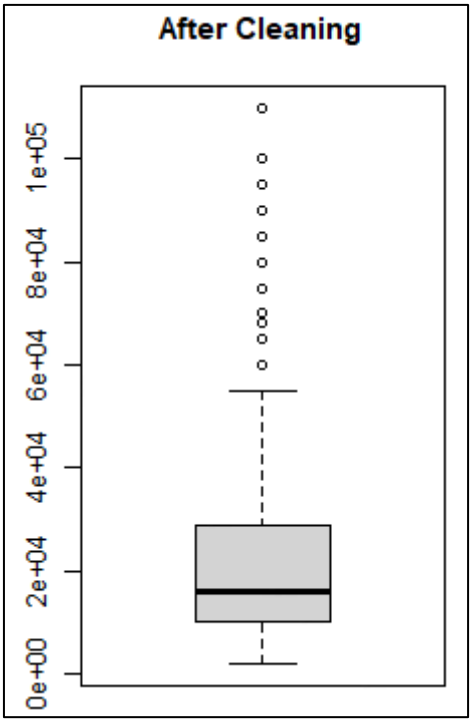



```
     Rent
190000
180000
180000
160000
150000
150000
140000
130000
130000
120000
120000
120000
120000
120000
110000
110000
100000
100000
100000
100000
```

3.3 Rent Cleaning for Delhi

```
#Cleaning for Delhi Rent
max(city_subsets$Delhi$Rent)
city_subsets$Delhi = city_subsets$Delhi[city_subsets$Delhi$Rent !=530000, ]
city_subsets$Delhi = city_subsets$Delhi[city_subsets$Delhi$Rent !=350000, ]
city_subsets$Delhi = city_subsets$Delhi[city_subsets$Delhi$Rent !=280000, ]
city_subsets$Delhi = city_subsets$Delhi[city_subsets$Delhi$Rent !=260000, ]
city_subsets$Delhi = city_subsets$Delhi[city_subsets$Delhi$Rent !=250000, ]
city_subsets$Delhi = city_subsets$Delhi[city_subsets$Delhi$Rent !=200000, ]
city_subsets$Delhi = city_subsets$Delhi[city_subsets$Delhi$Rent !=190000, ]
city_subsets$Delhi = city_subsets$Delhi[city_subsets$Delhi$Rent !=150000, ]
city_subsets$Delhi = city_subsets$Delhi[city_subsets$Delhi$Rent !=140000, ]
city_subsets$Delhi = city_subsets$Delhi[city_subsets$Delhi$Rent !=130000, ]

#Comparison before and after cleaning
boxplot(city_subsets_unclean$Delhi$Rent)
title("Before Cleaning")
boxplot(city_subsets$Delhi$Rent)
title("After Cleaning")

head(city_subsets$Delhi[order(city_subsets$Delhi$Rent, decreasing = TRUE), ], 20)
```

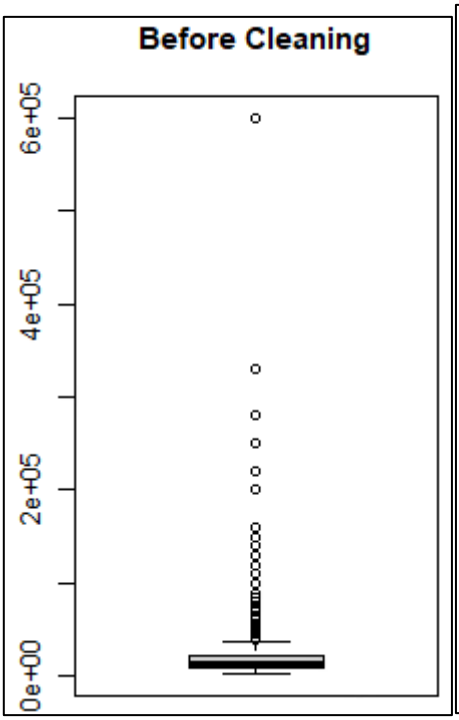


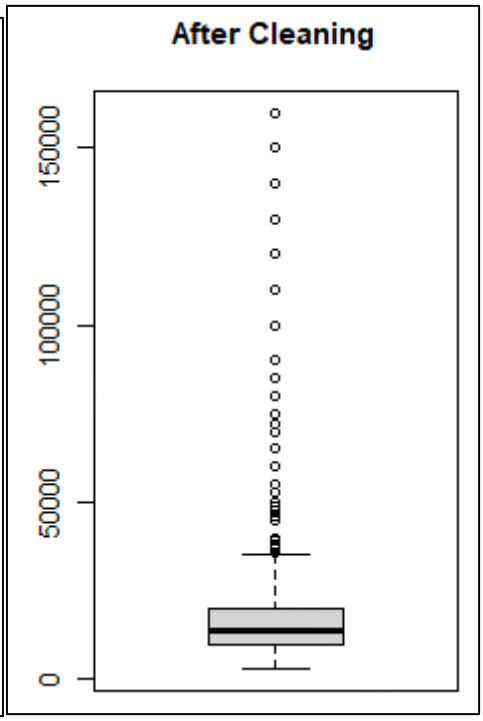



```
  Rent
110000
110000
100000
100000
100000
100000
100000
 95000
 95000
 95000
 90000
 90000
 90000
 90000
 90000
 85000
 85000
 80000
 80000
 80000
```

3.4 Rent Cleaning for Chennai

```
#Cleaning for Chennai Rent
max(city_subsets$Chennai$Rent)
city_subsets$Chennai = city_subsets$Chennai[city_subsets$Chennai$Rent !=600000, ]
city_subsets$Chennai = city_subsets$Chennai[city_subsets$Chennai$Rent !=330000, ]
city_subsets$Chennai = city_subsets$Chennai[city_subsets$Chennai$Rent !=280000, ]
city_subsets$Chennai = city_subsets$Chennai[city_subsets$Chennai$Rent !=250000, ]
city_subsets$Chennai = city_subsets$Chennai[city_subsets$Chennai$Rent !=220000, ]
city_subsets$Chennai = city_subsets$Chennai[city_subsets$Chennai$Rent !=200000, ]

#Comparison before and after cleaning
boxplot(city_subsets_unclean$Chennai$Rent)
title("Before Cleaning")
boxplot(city_subsets$Chennai$Rent)
title("After Cleaning")

head(city_subsets$Chennai[order(city_subsets$Chennai$Rent, decreasing = TRUE), ], 20)
```

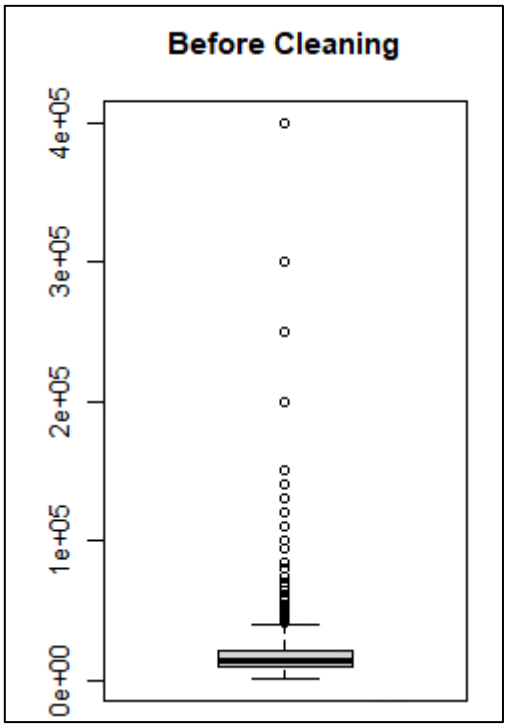


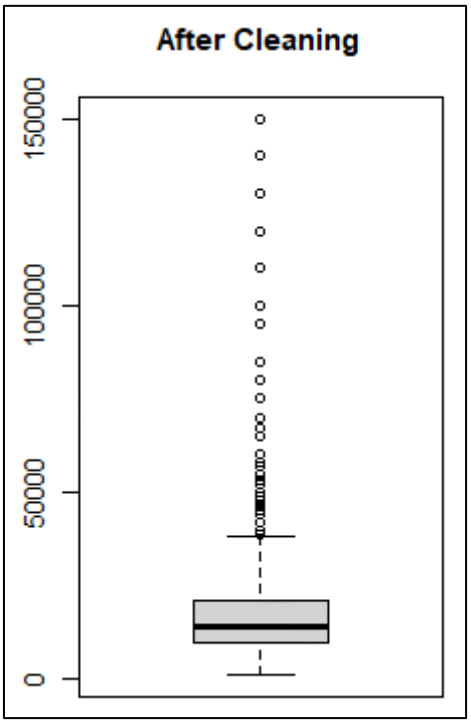


Rent
160000
150000
150000
140000
130000
130000
120000
120000
110000
100000
100000
100000
100000
100000
90000
90000
90000
85000
85000
80000

3.5 Rent Cleaning for Hyderabad

```
#Cleaning for Hyderabad Rent
max(city_subsets$Hyderabad$Rent)
city_subsets$Hyderabad = city_subsets$Hyderabad[city_subsets$Hyderabad$Rent !=400000, ]
city_subsets$Hyderabad = city_subsets$Hyderabad[city_subsets$Hyderabad$Rent !=300000, ]
city_subsets$Hyderabad = city_subsets$Hyderabad[city_subsets$Hyderabad$Rent !=250000, ]
city_subsets$Hyderabad = city_subsets$Hyderabad[city_subsets$Hyderabad$Rent !=200000, ]

#Comparison before and after cleaning
boxplot(city_subsets_unclean$Hyderabad$Rent)
title("Before Cleaning")
boxplot(city_subsets$Hyderabad$Rent)
title("After Cleaning")

head(city_subsets$Hyderabad[order(city_subsets$Hyderabad$Rent, decreasing = TRUE), ], 20)
}
```



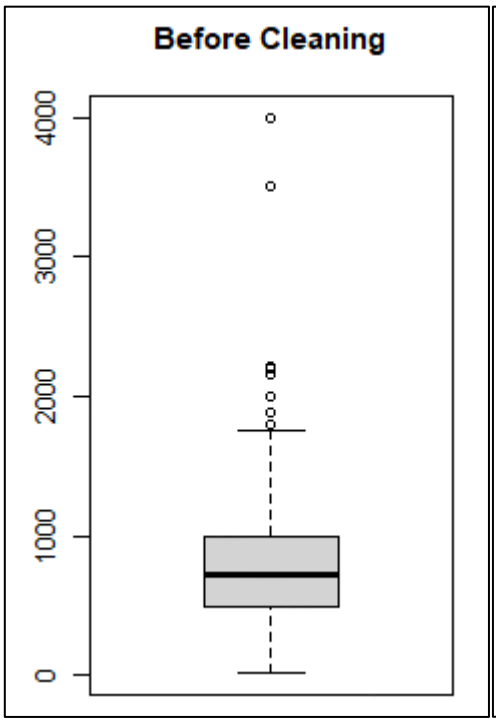


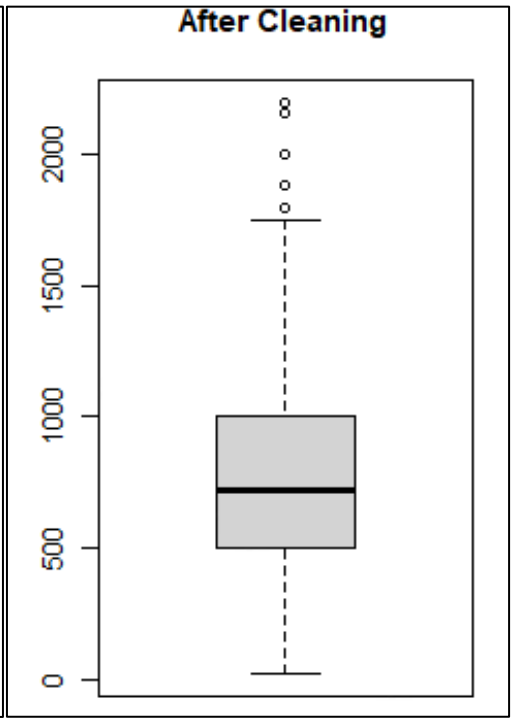
Rent
150000
140000
140000
130000
130000
130000
120000
120000
110000
100000
95000
85000
80000
80000
75000
75000
70000
70000
67000
65000

3.6 Size Cleaning for Kolkata

```
#Cleaning for Kolkata Size
max(city_subsets$Kolkata$Size)
city_subsets$Kolkata = city_subsets$Kolkata[city_subsets$Kolkata$Size!=4000, ]
city_subsets$Kolkata = city_subsets$Kolkata[city_subsets$Kolkata$Size!=3500, ]

#Comparison before and after cleaning
boxplot(city_subsets_unclean$Kolkata$Size)
title("Before Cleaning")
boxplot(city_subsets$Kolkata$Size)
title("After Cleaning")

head(city_subsets$Kolkata[order(city_subsets$Kolkata$Size, decreasing = TRUE), ], 20)
```

Above shows the code for data cleaning in 'Size' attribute. The cleaning process are similar with the 'Rent' cleaning.

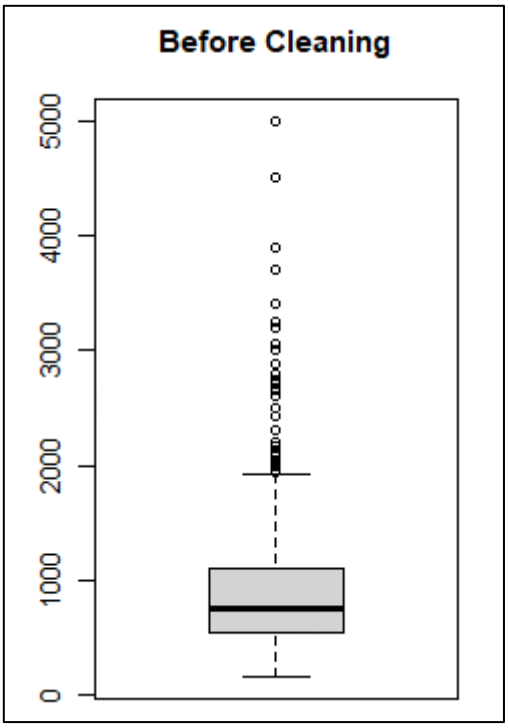


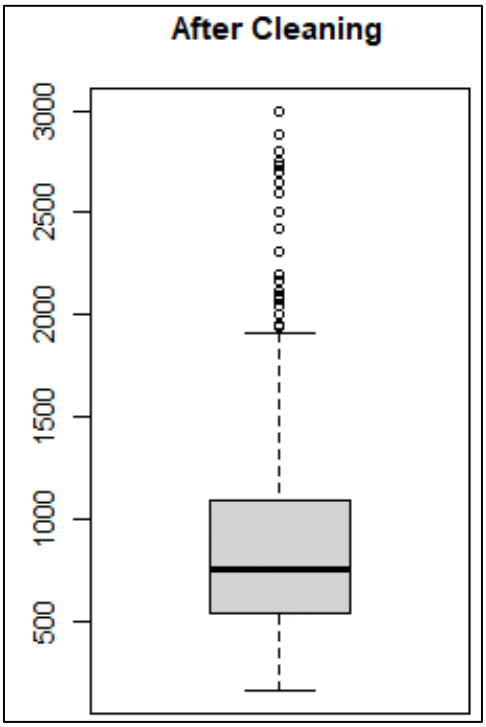


Size
2200
2160
2000
2000
2000
1880
1800
1800
1800
1750
1700
1700
1600
1600
1600
1600
1600
1600
1600
1600

3.6 Size Cleaning for Mumbai

```
#Cleaning for Mumbai Size
max(city_subsets$Mumbai$Size)
city_subsets$Mumbai = city_subsets$Mumbai[city_subsets$Mumbai$Size !=3700, ]
city_subsets$Mumbai = city_subsets$Mumbai[city_subsets$Mumbai$Size !=3250, ]

#Comparison before and after cleaning
boxplot(city_subsets_unclean$Mumbai$Size)
title("Before Cleaning")
boxplot(city_subsets$Mumbai$Size)
title("After Cleaning")

head(city_subsets$Mumbai[order(city_subsets$Mumbai$Size, decreasing = TRUE), ], 20)
```

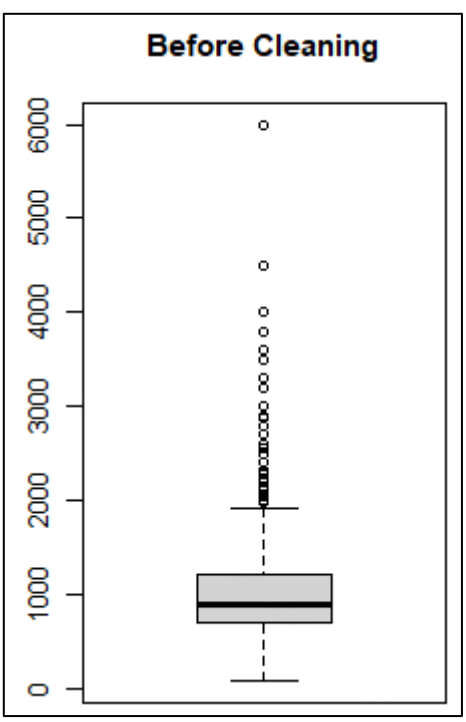


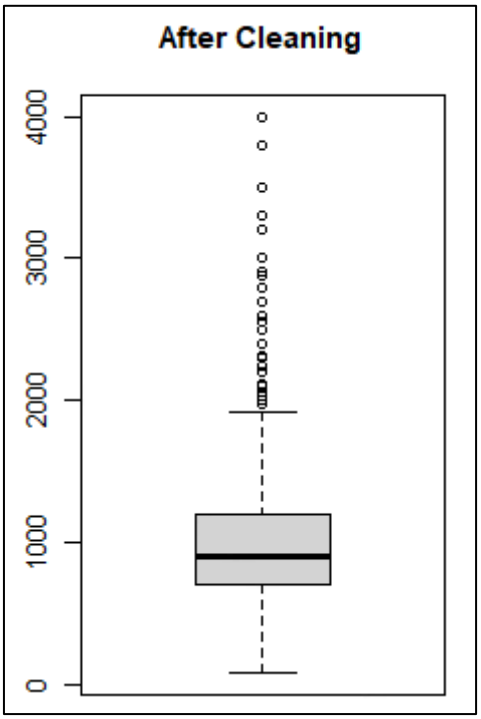



```
   Size
3000
3000
2880
2800
2800
2800
2800
2800
2750
2729
2700
2650
2600
2500
2500
2500
2500
2500
2500
2500
2424
```

3.7 Size Cleaning for Chennai

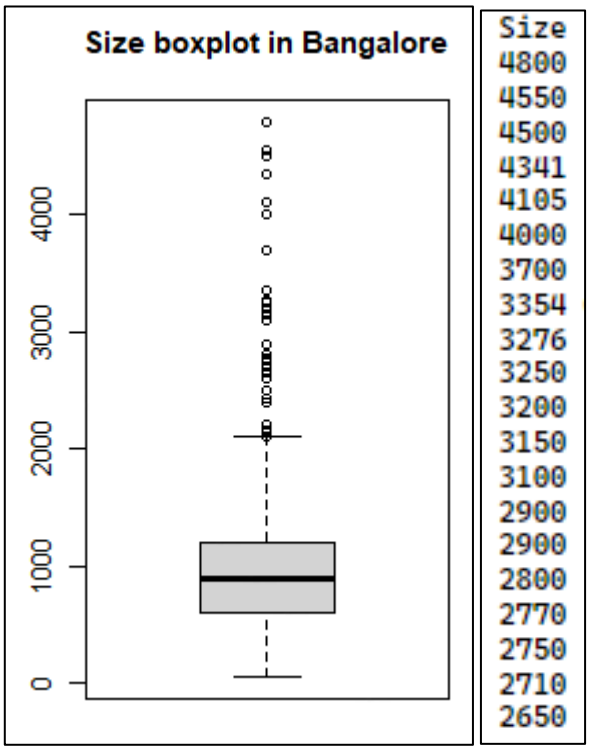
```
#Cleaning for Chennai Size
max(city_subsets$Chennai$Size)
city_subsets$Chennai = city_subsets$Chennai[city_subsets$Chennai$Size !=6000, ]

#Comparison before and after cleaning
boxplot(city_subsets_unclean$Chennai$Size)
title("Before Cleaning")
boxplot(city_subsets$Chennai$Size)
title("After Cleaning")

head(city_subsets$Chennai[order(city_subsets$Chennai$Size, decreasing = TRUE), ], 20)
```



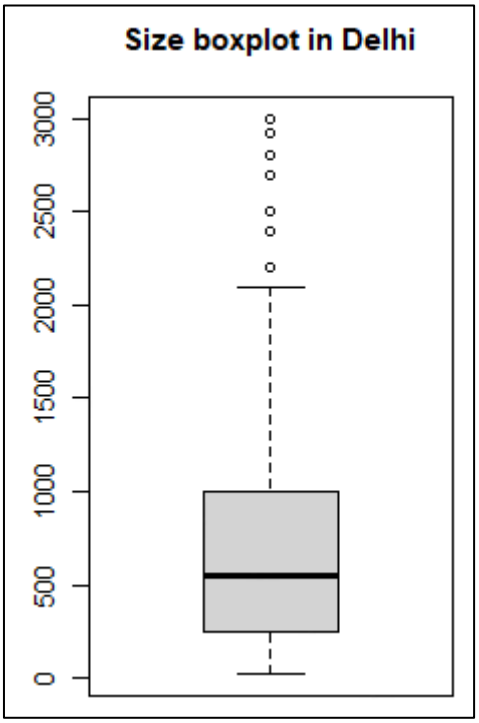


Size
4000
3800
3500
3300
3200
3000
3000
3000
2900
2880
2800
2792
2700
2600
2550
2500
2500
2400
2400
2400

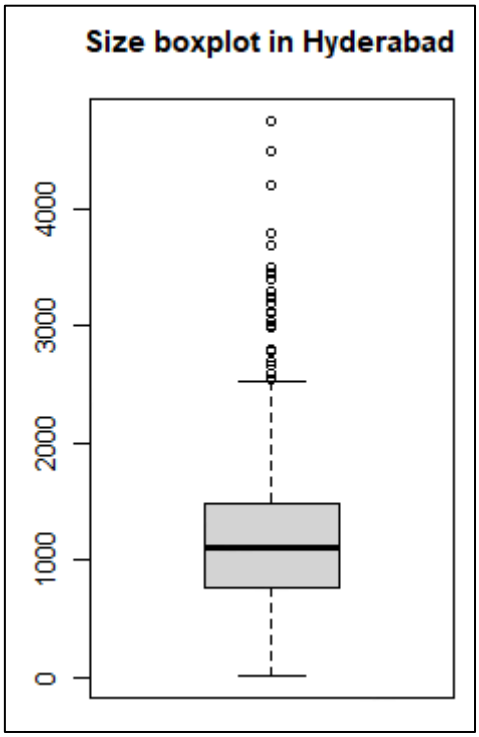
3.8 Size Cleaning in Bangalore, Delhi & Hyderabad City



Size
4800
4550
4500
4341
4105
4000
3700
3354
3276
3250
3200
3150
3100
2900
2900
2800
2770
2750
2710
2650



Size
3000
2925
2800
2700
2700
2700
2700
2700
2500
2500
2500
2400
2400
2200
2200
2200
2200
2100
2000
2000



Size
4761
4500
4200
3800
3800
3789
3698
3500
3500
3455
3400
3300
3250
3200
3200
3200
3133
3115
3050
3000

As the three boxplots shown above, there is no significant outlier in ‘Size’ attribute. Hence, there is no need data cleaning for ‘Size’ attribute in Hyderabad, Bangalore, and Delhi city.

4.0 Building and Evaluating Predictive Models

4.1 Linear Regression Model

```
# Split the data into training and testing sets
train_index <- sample(seq_len(nrow(HouseRent_Cleaned_Preprocessed)), 0.7 * nrow(HouseRent_Cleaned_Preprocessed))
train_data <- HouseRent_Cleaned_Preprocessed[train_index, ]
test_data <- HouseRent_Cleaned_Preprocessed[-train_index, ]

# Fit the linear regression model
lm_model <- lm(Rent ~ ., data = train_data)

# Model summary
summary(lm_model)
```

```
Call:
lm(formula = Rent ~ ., data = train_data)

Residuals:
   Min     1Q Median     3Q    Max
-97954 -13042  -1551   9513 279295

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)       -38514.722   2193.415 -17.559  < 2e-16 ***
BHK                 3567.276   1069.739   3.335 0.000863 ***
Size                  18.074      1.357  13.321  < 2e-16 ***
Area.Type           1179.477   1176.095   1.003 0.315995
CityChennai        -2711.681   1530.412  -1.772 0.076510 .
CityDelhi           5568.554   1792.837   3.106 0.001913 **
CityHyderabad      -7330.997   1558.640  -4.703 2.66e-06 ***
CityKolkata         2104.332   1854.256   1.135 0.256515
CityMumbai         44456.631   1751.389  25.384  < 2e-16 ***
Furnishing.Status   4357.101    710.204   6.135 9.54e-10 ***
Tenant.Preferred    2221.573    744.740   2.983 0.002875 **
Bathroom           13071.679   1077.925  12.127  < 2e-16 ***
Point.of.Contact   11483.739   1423.308   8.068 9.91e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 26770 on 3270 degrees of freedom
Multiple R-squared:  0.6081,    Adjusted R-squared:  0.6066
F-statistic: 422.8 on 12 and 3270 DF,  p-value: < 2.2e-16
```

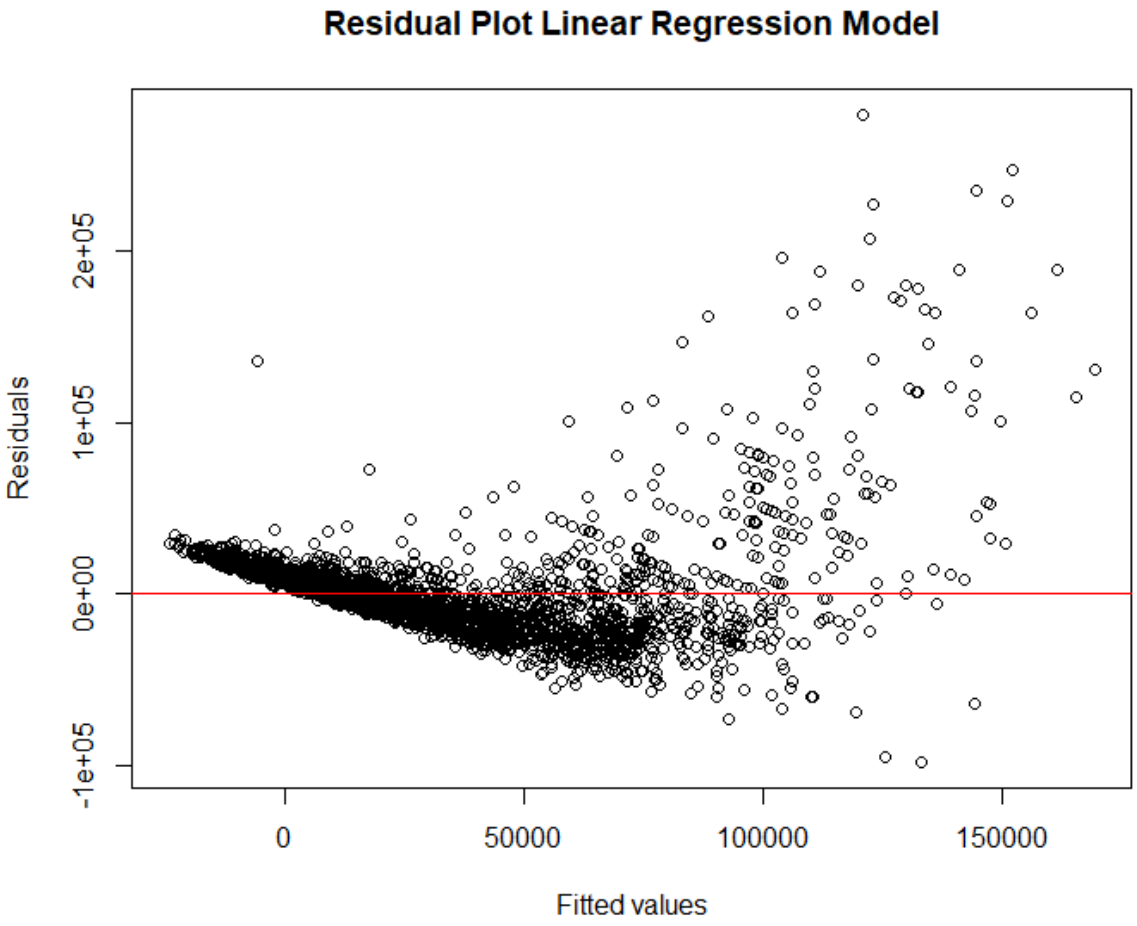
The Multiple R-squared value of 0.6081 suggests that the model explains about 60.81% of the variability in the rents. The Adjusted R-squared (which accounts for the number of predictors in the model) is slightly lower at 60.66%, indicating that most predictors are relevant.

Conclusions:

- Size, number of bedrooms (BHK), furnishing status, and the number of bathrooms is strong predictors of rent.
- The city where a property is located plays a significant role in determining its rent; Mumbai has substantially higher rents compared to other cities.

- The type of area measurement (Super Area, Carpet Area, Built Area) is not a statistically significant predictor of rent in this model, suggesting that other factors may be more influential.
- The model is a good fit for the data, but there's still around 40% of variability in rents that it does not explain.

```
# Extract residuals from the model
residuals <- resid(lm_model)

# Create a plot of residuals
plot(predict(lm_model), residuals,
     xlab = "Fitted values",
     ylab = "Residuals",
     main = "Residual Plot")
abline(h = 0, col = "red")  # Add a horizontal line at y = 0
```



1. **Zero Line**: The horizontal red dashed line represents zero residual or no error. The points that are above the line represent over-predictions (predicted value > actual value), while points below the line represent under-predictions (predicted value < actual value).

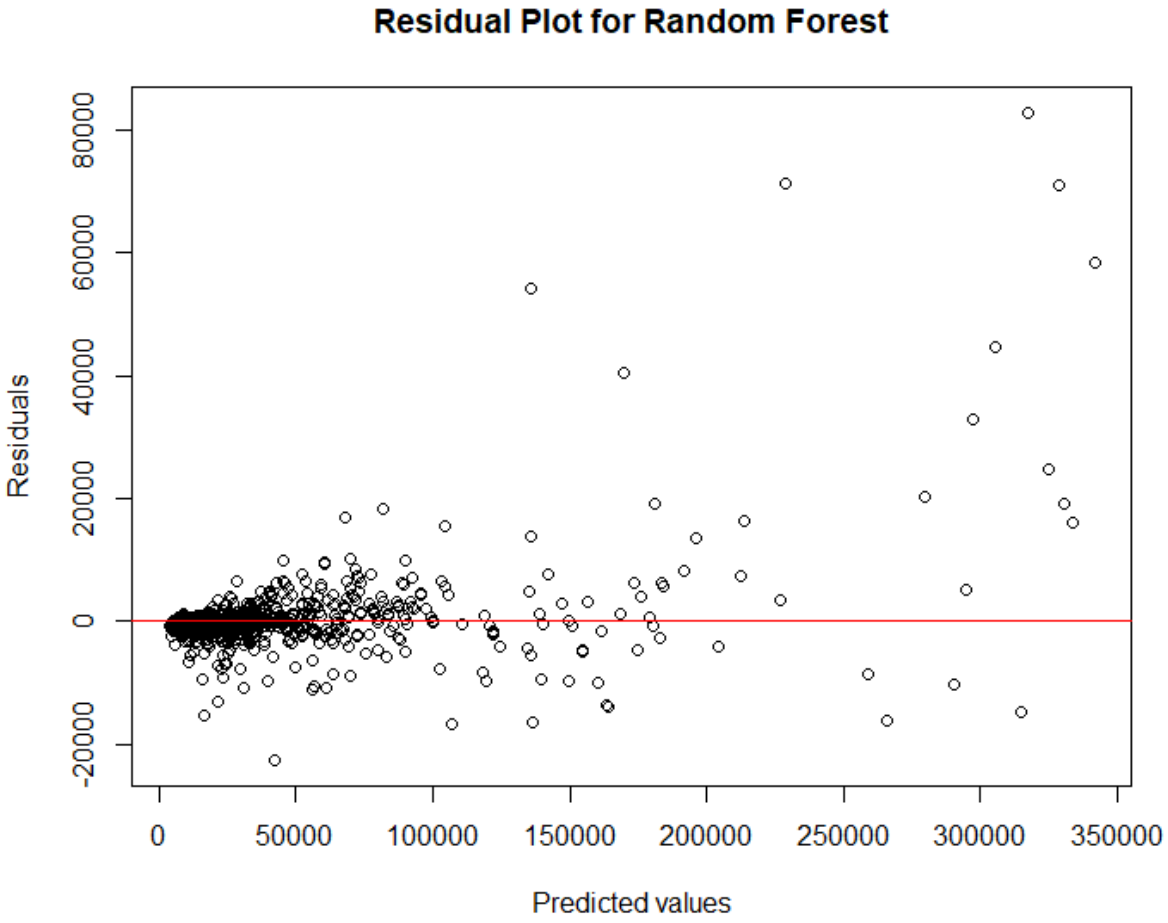
2. **Scatter**: According to Kenton (2022), a good model will have residuals scattered randomly around the zero line. However, the plot shows a pattern where residuals are not entirely scattered randomly around zero, meaning the model might not capture some underlying patterns in the data or there may be some non-linear relationships that a linear model cannot capture.

3. **Outliers**: There are some points far from the zero line, indicating large residuals. These could be potential outliers or influential points that the model has trouble predicting.

4.2 Random Forest Model

```
# Create a vector of random indices for training data
train_indices <- sample(1:nrow(HouseRent_Cleaned_Preprocessed), size = 0.7 * nrow(HouseRent_Cleaned_Preprocessed))

# Split the data
X_train <- HouseRent_Cleaned_Preprocessed[train_indices,]
y_train <- HouseRent_Cleaned_Preprocessed$Rent[train_indices]
X_test <- HouseRent_Cleaned_Preprocessed[-train_indices,]
y_test <- HouseRent_Cleaned_Preprocessed$Rent[-train_indices]
# Train the random forest model
rf_model <- randomForest(y_train ~ ., data = X_train)
```

```
# Make predictions
rf_prediction <- predict(rf_model, newdata = X_test)

rf_residuals <- y_test - rf_prediction

# Create a plot of residuals
plot(rf_prediction, rf_residuals,
     xlab = "Predicted values",
     ylab = "Residuals",
     main = "Residual Plot for Random Forest")
abline(h = 0, col = "red")  # Add a horizontal line at y = 0
```



From the graph above, the residuals are scattered around the zero line, which is good as it indicates homoscedasticity (constant variance of residuals across all levels of the independent variables) (Kenton, 2022). Additionally, there are no obvious trends or patterns in the residuals, which means that the model is reasonably well-specified for the data.

```
# Calculate Mean Absolute Error (MAE)
mae_rf <- mean(abs(rf_prediction - y_test))

# Calculate Mean Squared Error (MSE)
mse_rf <- mean((rf_prediction - y_test)^2)

# Calculate Root Mean Squared Error (RMSE)
rmse_rf <- sqrt(mse_rf)

# Calculate R-squared
r_squared_rf <- 1 - sum((rf_prediction - y_test)^2) / sum((y_test - mean(y_test))^2)

# Print evaluation metrics
cat("Random Forest Model Evaluation:\n")
cat("Mean Absolute Error:", mae_rf, "\n")
cat("Mean Squared Error:", mse_rf, "\n")
cat("Root Mean Squared Error:", rmse_rf, "\n")
cat("R-squared:", r_squared_rf, "\n")
```

```
> cat("Random Forest Model Evaluation:\n")
Random Forest Model Evaluation:
> cat("Mean Absolute Error:", mae_rf, "\n")
Mean Absolute Error: 1635.71
> cat("Mean Squared Error:", mse_rf, "\n")
Mean Squared Error: 28586043
> cat("Root Mean Squared Error:", rmse_rf, "\n")
Root Mean Squared Error: 5346.592
> cat("R-squared:", r_squared_rf, "\n")
R-squared: 0.9858915
```

Mean Absolute Error (MAE) - 1635.71:

On average, the model's predictions are off by 1635.71 units (in the currency of the rent). Therefore, this signifies that the anticipated rent values differ from the actual rent prices by this amount on average.

Mean Squared Error (MSE) - 28,586,043:

This metric provides the average squared differences between predicted and actual values. Higher values indicate worse performance. Since it's squared, favours large errors over small ones, making it sensitive to outliers.

Root Mean Squared Error (RMSE) - 5346.592:

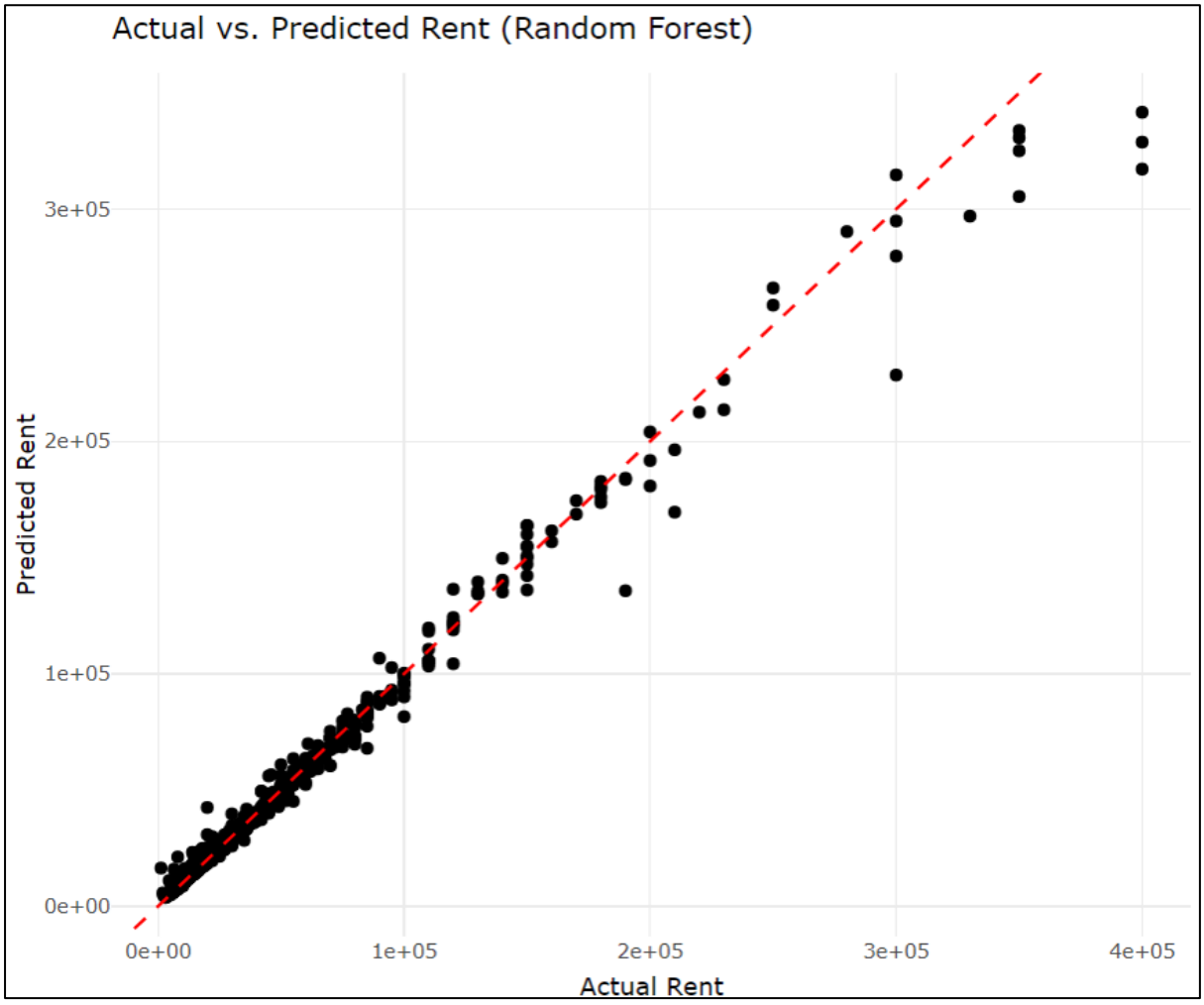
This is MSE's square root. It stands for the residuals' standard deviation (prediction mistakes). An increased RMSE is the outcome of predictions that are further off than the actual values. In this case, the anticipated rent values differ by an average of 5346.592 units from the actual rent values. When RMSE and MAE are compared, the model occasionally has higher mistakes because the RMSE is a great deal larger than the MAE.

R-squared (R^2) - 0.9858915:

This metric represents the proportion of the variance in the dependent variable (Rent) that is predictable from the independent variables (features). An R^2 value of 0.9858915 means that the model explains approximately 98.59% of the variance in the rent prices. This is a very high

R^2 value, indicating a strong performance by the model. However, very high R^2 values may sometimes indicate overfitting. Thus, it is best to test the model on new, unseen data to ensure its robustness and generalisation capabilities (Frost, 2023).

```
# Create a data frame with actual and predicted values
actual_vs_predicted <- data.frame(Actual = y_test, Predicted = rf_prediction)

# Create a scatter plot
plot <- ggplot(actual_vs_predicted, aes(x = Actual, y = Predicted)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1, color = "red", linetype = "dashed") +  # Add a reference line
  labs(title = "Actual vs. Predicted Rent (Random Forest)",
       x = "Actual Rent",
       y = "Predicted Rent") +
  theme_minimal()

p <- ggplotly(plot)
p
```



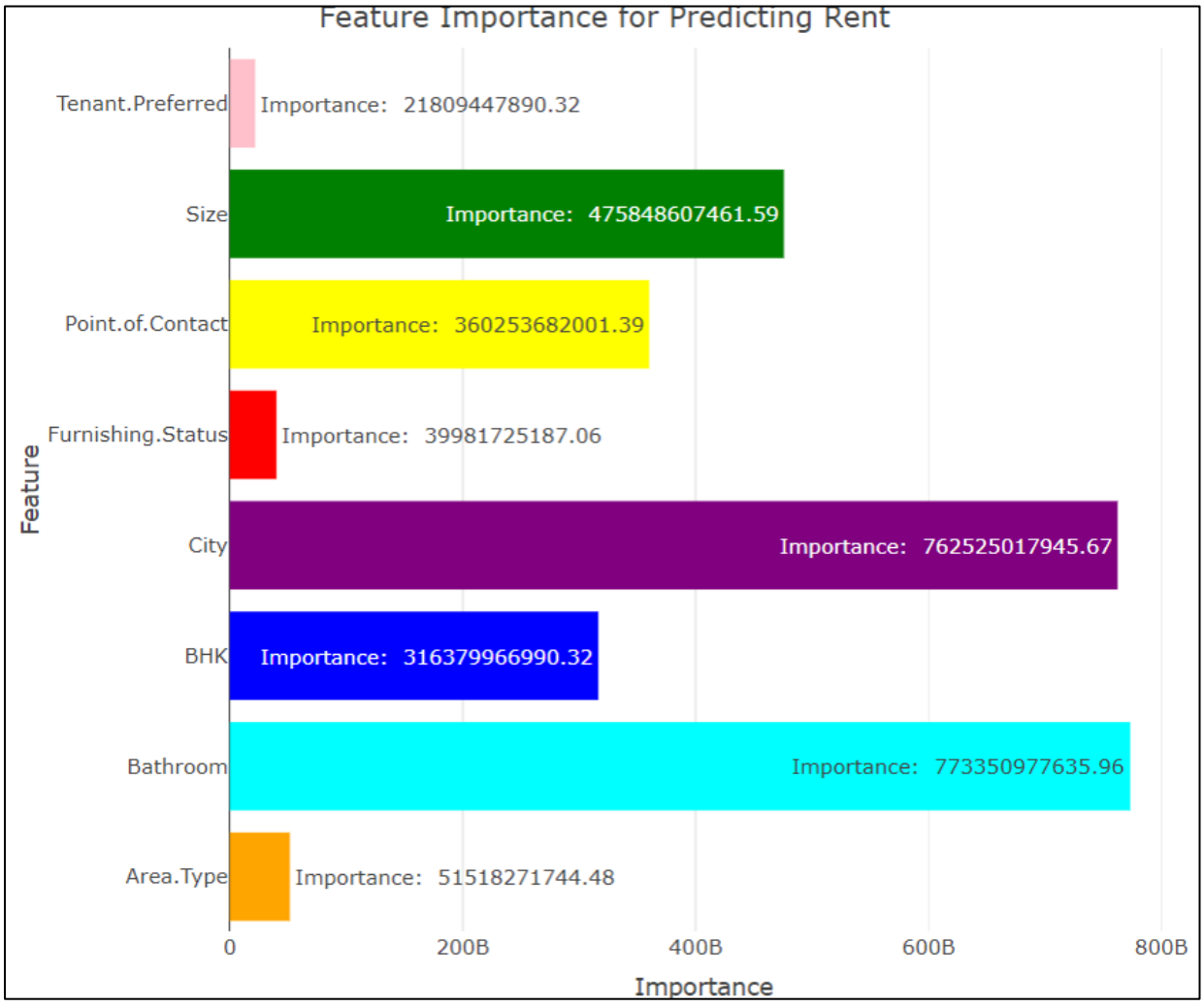
Above is a scatter plot of actual vs. predicted rent values using for the random forest model. This is a common approach to visually assess how well a regression model's predictions match the actual values. The red dashed line represents the line of perfect prediction where actual and

predicted values are equal. Although the plot does not directly demonstrate accuracy, it helps see how closely model's predictions match the actual numbers. The plot helps find patterns and differences in the model's predictions by comparing projected and actual values. In this case, the model is not very accurate when it comes to predicting high rent values. This could be due to a variety of reasons:

1. The model may emphasise accuracy for lower values since certain features may have a greater impact on lower rent values than higher ones.

2. Model Complexity: It's possible that the model overfits to lower rent values while undergeneralizing to larger ones, and it can happen if the model is overly complicated.

```r
# Get feature importance scores
feature_importance <- importance(rf_model)

# Convert feature importance scores to a data frame for easier manipulation
feature_importance_df <- data.frame(Feature = rownames(feature_importance), Importance = rowMeans(feature_importance))

# Exclude the "Rent" column from the dataframe
feature_importance_df <- feature_importance_df[feature_importance_df$Feature != "Rent", ]

# Define a palette of 8 colors
color_palette <- c("blue", "green", "orange", "purple", "red", "pink", "cyan", "yellow")

plot_fImportance <- plot_ly(data = feature_importance_df, x = ~Importance, y = ~Feature, type = "bar", orientation = 'h',
                marker = list(color = color_palette),
                text = ~paste("Importance: ", round(Importance, 2))) %>%
  layout(title = "Feature Importance for Predicting Rent",
         xaxis = list(title = "Importance"),
         yaxis = list(title = "Feature"),
         showlegend = FALSE)

# Show the plot
plot_fImportance
```



The random forest model's feature significance scores are illustrated in the image above. Each feature (predictor) in the dataset is ranked according to its relative importance using these scores. According to Arbor Analytics (2020), the “IncNodePurity” score is a measurement of how much the feature helps to lower the impurity (such as Gini impurity) of the nodes in the decision trees that make up the random forest. The importance rating for “Rent” is the highest, followed by "City" and "Bathroom." In contrast, other factors like “Furnishing.Status”,

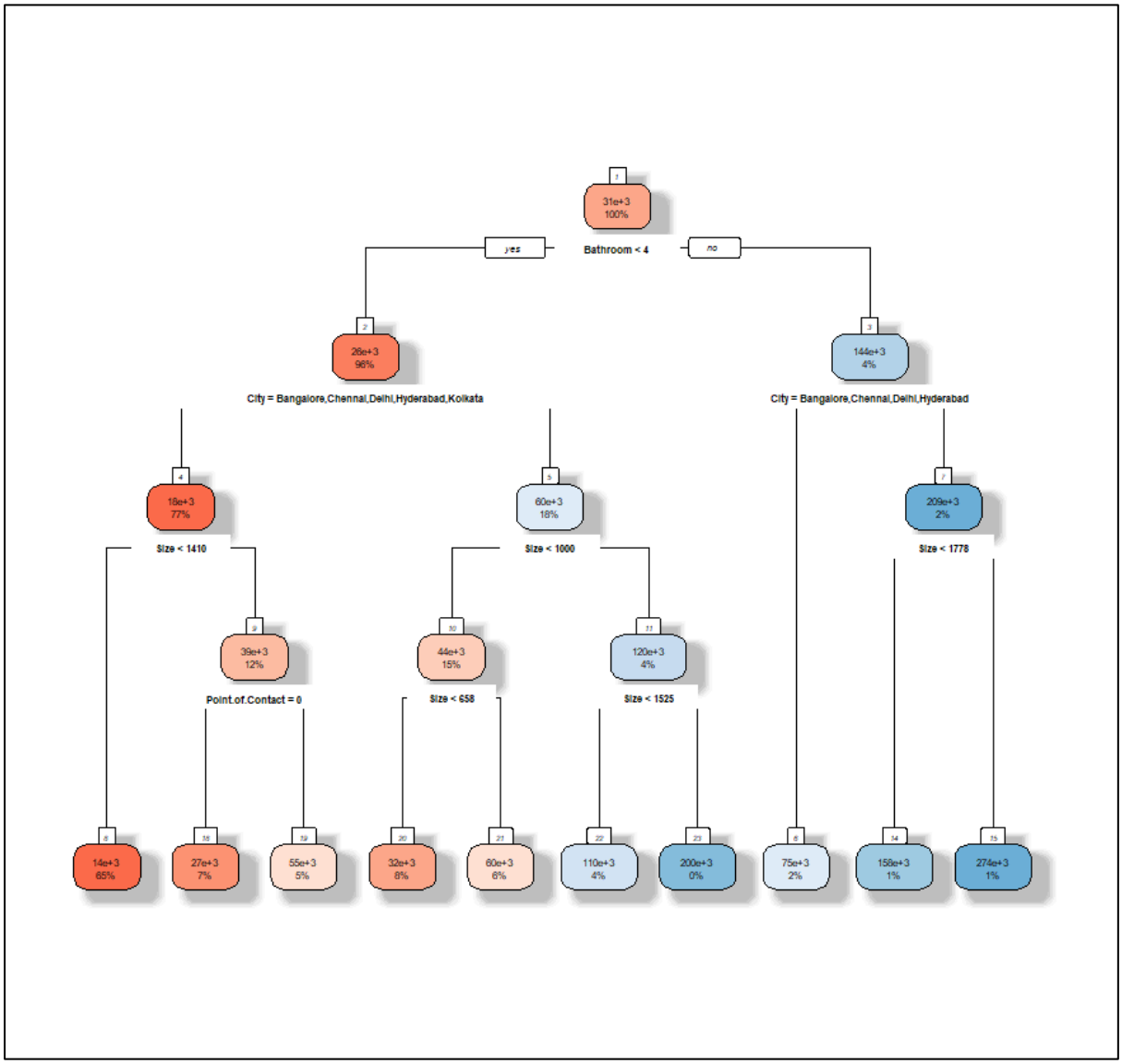
“Area.Type”, and “Tenant.Preferred” appear to have relatively lower relevance ratings. These features are considerably influencing the model's predictions.

4.3 Decision Tree Model

```
# Plot the decision tree
rpart.plot(tree_model, box.palette = "RdBu", shadow.col = "gray", nn = TRUE)

# Make predictions
tree_prediction <- predict(tree_model, newdata = test_data)

# Calculate Mean Absolute Error (MAE)
mae_tree <- mean(abs(tree_prediction - test_data$Rent))

# Calculate Mean Squared Error (MSE)
mse_tree <- mean((tree_prediction - test_data$Rent)^2)

# Calculate Root Mean Squared Error (RMSE)
rmse_tree <- sqrt(mse_tree)

# Calculate R-squared
r_squared_tree <- 1 - sum((tree_prediction - test_data$Rent)^2) / sum((test_data$Rent - mean(test_data$Rent))^2)


# Print evaluation metrics
cat("Decision Tree Model Evaluation:\n")
cat("Mean Absolute Error:", mae_tree, "\n")
cat("Mean Squared Error:", mse_tree, "\n")
cat("Root Mean Squared Error:", rmse_tree, "\n")
cat("R-squared:", r_squared_tree, "\n")
```

```
> cat("Decision Tree Model Evaluation:\n")
Decision Tree Model Evaluation:
> cat("Mean Absolute Error:", mae_tree, "\n")
Mean Absolute Error: 11799.06
> cat("Mean Squared Error:", mse_tree, "\n")
Mean Squared Error: 494753252
> cat("Root Mean Squared Error:", rmse_tree, "\n")
Root Mean Squared Error: 22243.05
> cat("R-squared:", r_squared_tree, "\n")
R-squared: 0.7822479
```

```
# Fitting a decision tree model to predict Rent based on other features
tree_model <- rpart(Rent ~ ., data = train_data, method = "anova")

# Plot the decision tree
rpart.plot(tree_model, box.palette = "RdBu", shadow.col = "gray", nn = TRUE)
```



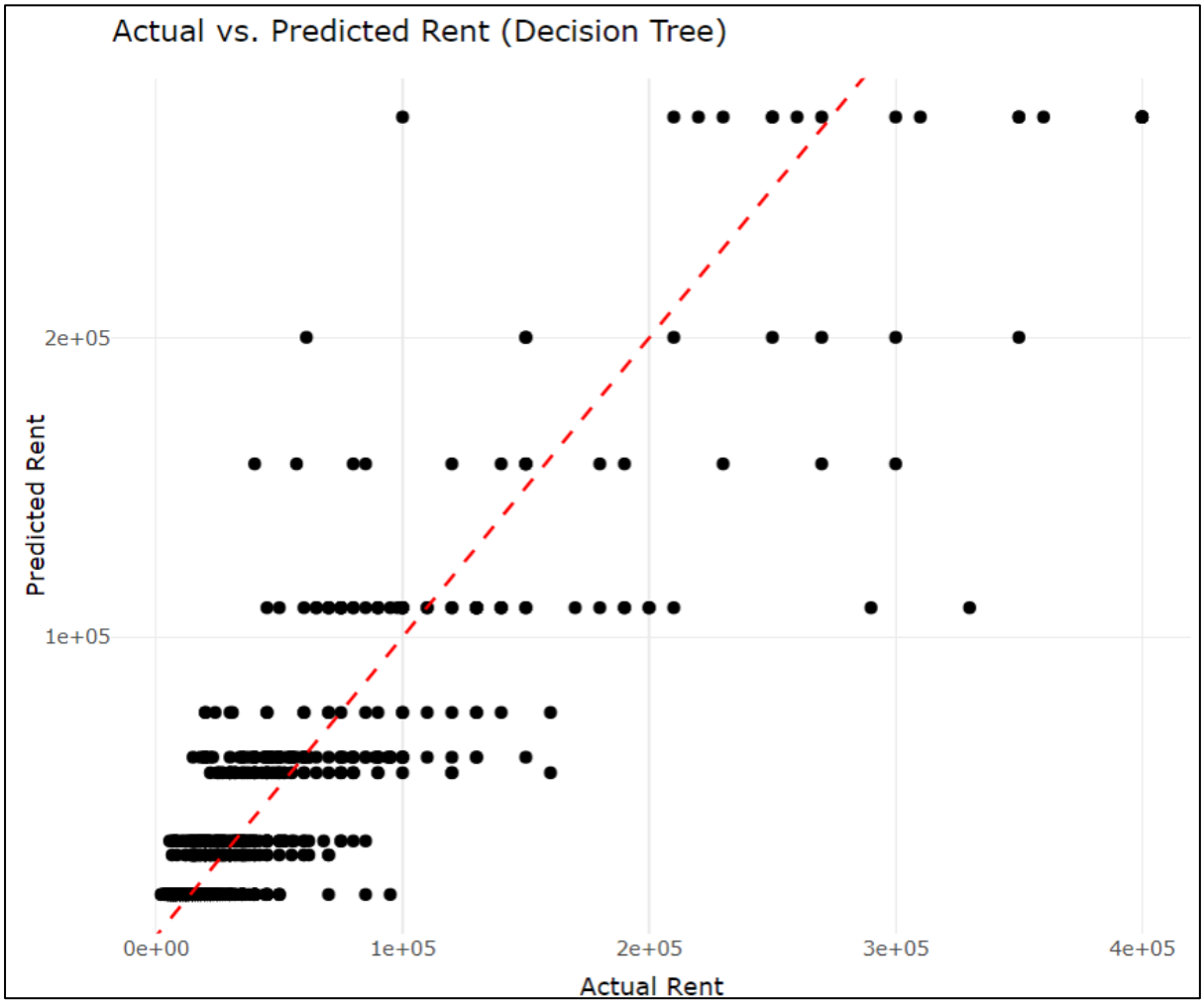
A decision tree is a non-parametric supervised learning algorithm used for regression and classification tasks. It has a tree structure with is organised hierarchically, and has a root node, branches, internal nodes, and leaf nodes (IBM, n.d.).

- In the above diagram, each node displays the feature used for the split.
- The colour of the node represents the average value of the target variable in that node, with darker shades of orange indicating higher values and darker shades of blue indicating lower values.

From the tree, the following interpretations can be made:

1. **Bathroom** is the most important feature for predicting rent, as it's the first feature the model splits on; **City** is the second most important feature.
2. **Size** is also significant, as it appears in the subsequent splits.
3. **Point.of.Contact** also plays a role in determining the rent but to a lesser extent.

```
# Create a data frame with actual and predicted values
actual_vs_predicted <- data.frame(Actual = test_data$Rent, Predicted = tree_prediction)

# Create a scatter plot
plot <- ggplot(actual_vs_predicted, aes(x = Actual, y = Predicted)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1, color = "red", linetype = "dashed") +  # Add a reference line
  labs(title = "Actual vs. Predicted Rent (Decision Tree)",
       x = "Actual Rent",
       y = "Predicted Rent") +
  theme_minimal()

p_dt <- ggplotly(plot)
p_dt
```



The graph compares the values of actual and predicted rent. Each point is a sample of data from the test set. While points further away from the diagonal line signify disparities between the actual and expected values, points closer to the diagonal line signify correct predictions.

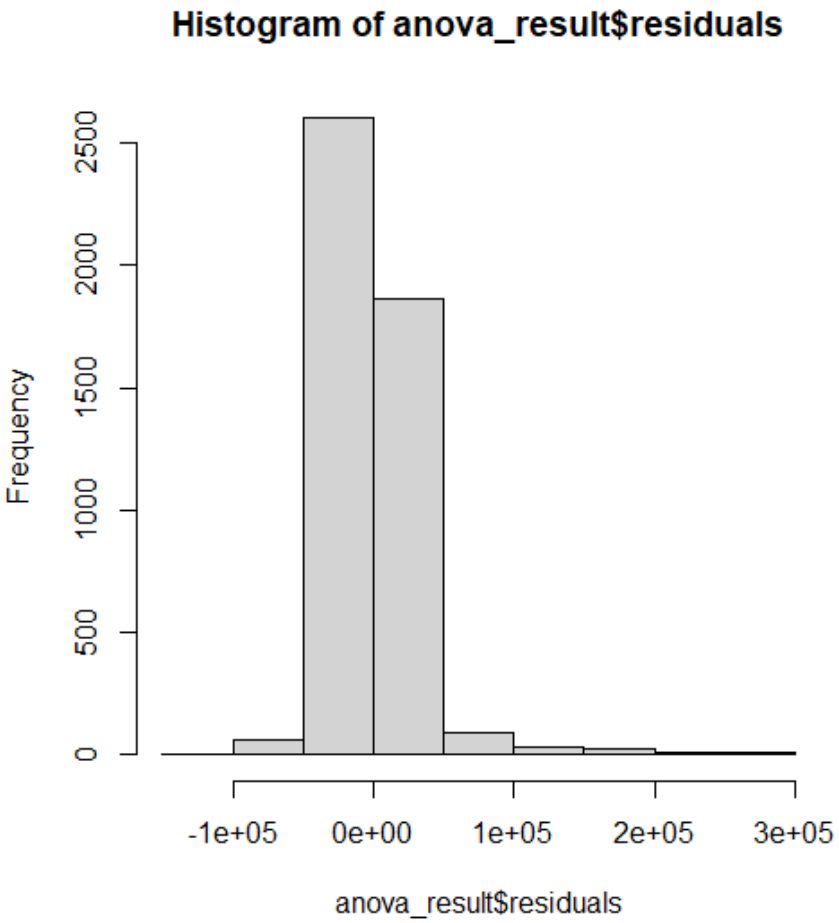
Additionally, there are numerous predictions located close to the diagonal, which suggests that the model is reasonably accurate. There are considerable differences, nevertheless, particularly for higher rent levels.

4.4 Anova testing for the whole dataset.

```
# Perform ANOVA testing
anova_result <- aov(Rent ~ ., data = HouseRent_Cleaned_Preprocessed)

# Summarize ANOVA results
summary(anova_result)

# Extract F-statistics and p-values for each predictor variable
f_statistics <- anova_result$"F value"
p_values <- anova_result$"Pr(>F)"

hist(anova_result$residuals)

qqPlot(anova_result$residuals,
       id = FALSE # id = FALSE to remove point identification
)
```

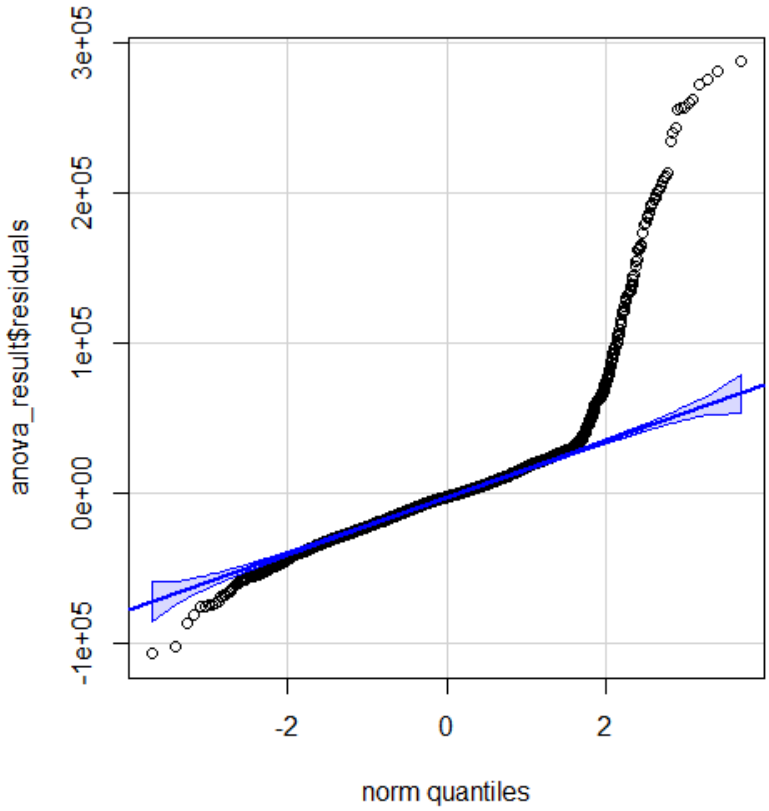
```
> summary(anova_result)
                    Df    Sum Sq   Mean Sq  F value Pr(>F)
BHK                  1 2.090e+12 2.090e+12 2180.354 <2e-16 ***
Size                 1 2.445e+11 2.445e+11  255.060 <2e-16 ***
Area.Type            1 5.962e+11 5.962e+11  621.852 <2e-16 ***
City                 1 5.325e+11 5.325e+11  555.462 <2e-16 ***
Furnishing.Status    1 1.328e+11 1.328e+11  138.480 <2e-16 ***
Tenant.Preferred     1 2.329e+08 2.329e+08    0.243  0.622
Bathroom             1 7.223e+11 7.223e+11  753.453 <2e-16 ***
Point.of.Contact     1 3.680e+11 3.680e+11  383.808 <2e-16 ***
Residuals         4681 4.488e+12 9.587e+08
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

All predictors—aside from "Tenant.Preferred"—have incredibly low p-values (0.05), showing that they are statistically significant in explaining the variation in the response variable. With a p-value of 0.622, which is higher than 0.05, the predictor "Tenant.Preferred" may not be meaningful in this model.



Observations of the above histogram:

- The distribution appears to be about centred around zero, meaning the model does not consistently anticipate rent prices higher or lower than they should be.
- Since the shape resembles a bell, it is likely that the residuals are generally regularly distributed.
- The residuals on the extreme left and right show a few significant negative and positive residuals, respectively. These might be anomalies or situations where the model did not match the data very well.



A "quantile-quantile" plot, or Q-Q plot, is used to determine if a collection of data comes from a theoretical distribution. This graph is typically used to examine if a collection of data reflects a normal distribution (Zach, 2021).

Additionally, the QQ-plots' points often follow a straight line, and many of them fall inside the confidence intervals, suggesting that residuals have a distribution that is like that of a normal distribution (Soetewey, 2020).

From the figure above most of the residuals fall inside the confidence intervals, which means that they are approximately normally distributed. However, there are some deviations in the tails, indicating potential outliers or heavier tails than a normal distribution.

5.0 Data Exploration

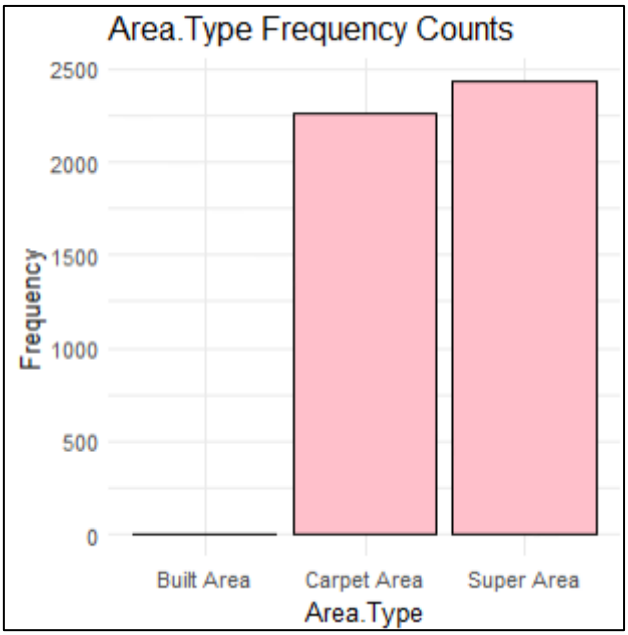
5.1 Data Exploration for Area Type

```
Area.Type_Count = as.data.frame(table(HouseRent_Cleaned$Area.Type))
names(Area.Type_Count)[1] = "Area.Type"
print(Area.Type_Count, row.names = FALSE)
```

```
  Area.Type Freq
 Built Area    2
Carpet Area 2258
 Super Area 2430
```

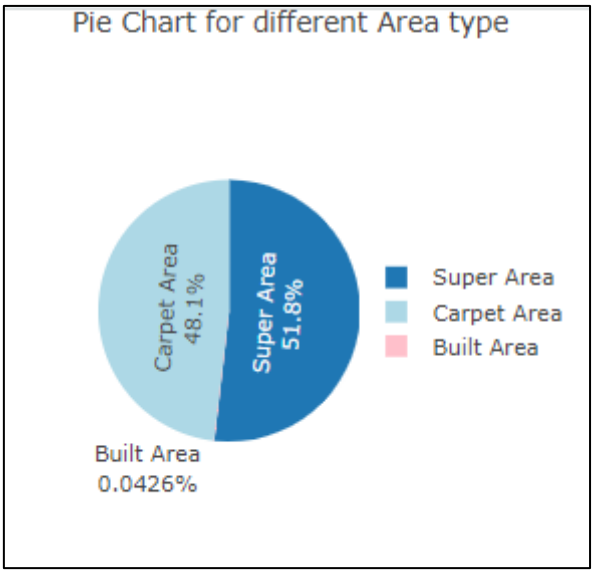
In the Area Type attribute, ‘Built Area’ has a very low share of the dataset; Meanwhile the other two variable, ‘Carpet Area’ and ‘Super Area’ have the similar row of value.

```
ggplot(Area.Type_Count, aes(x = Area.Type, y = Freq)) +
  geom_bar(stat = "identity", fill = "pink", color = "black") +
  labs(title = "Area.Type Frequency Counts",
       x = "Area.Type",
       y = "Frequency") +
  theme_minimal()
```



As the bar chart shown, ‘Carpet Area’ and ‘Super Area’ have the similar row of observation in the dataset, forms a clear difference compared to ‘Built Area’.

```
plot_ly(Area.Type_Count, labels =~Area.Type, values = ~Freq, type = "pie",
        marker = list(colors = c("pink", "lightblue")),
        textinfo = "label+percent",
        hoverinfo = 'text',
        text = ~paste('Area.Type = ',Area.Type, '\nFreq = ', Freq))%>%
  layout(showlegend = TRUE,
         legend = list(orientation = 'v', x = 1, y = 0.5),
         title = list(text = "Pie Chart for different Area type", font = list(size = 15)))
```



'Super Area' holds a 51.8% of data where 'Carpet Area' have a percentage 48.1%.

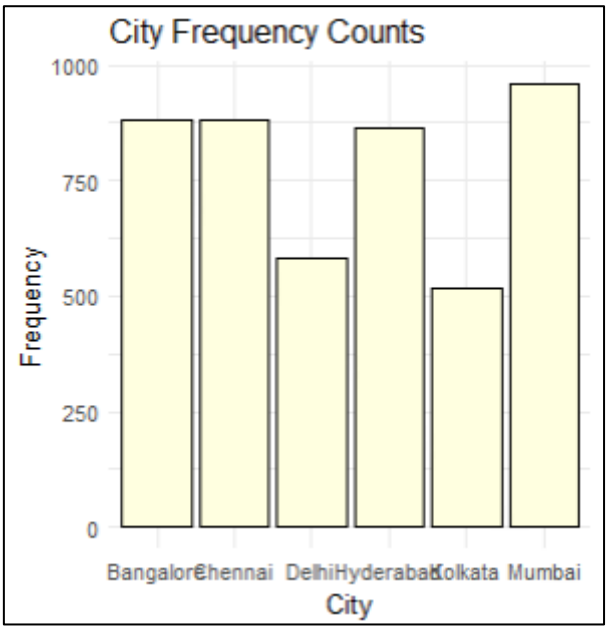
5.2 Data Exploration for City

```
City_Count = as.data.frame(table(HouseRent_Cleaned$City))
names(City_Count)[1] = "City"
print(City_Count, row.names = FALSE)
```

```
      City Freq
 Bangalore  882
   Chennai  883
     Delhi  584
 Hyderabad  863
   Kolkata  519
    Mumbai  959
```

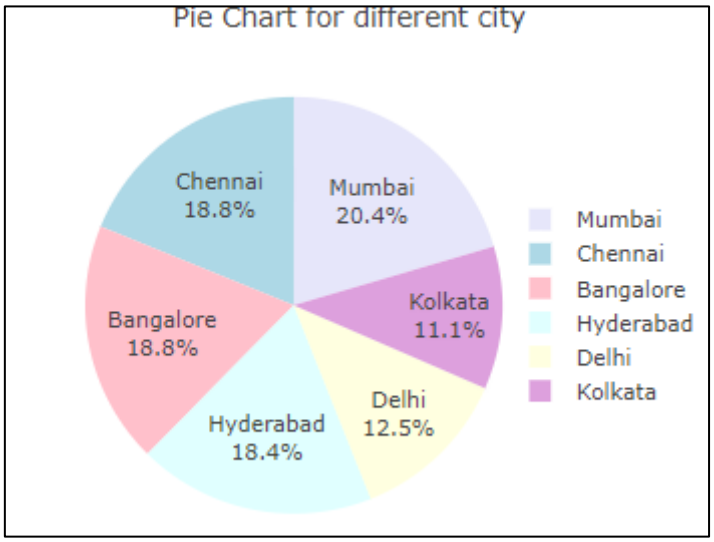
Based on the result, we can understand that all the cities have quite similar distribution, expect for Delhi and Kolkata which only have 500+ lines.

```
ggplot(City_Count, aes(x = City, y = Freq)) +
  geom_bar(stat = "identity", fill = "lightyellow", color = "black") +
  labs(title = "City Frequency Counts",
       x = "City",
       y = "Frequency") +
  theme_minimal()
```



The bar chart also shows that the Delhi and Kolkata have less frequency, while Mumbai have slightly higher than the others three do, which is Bangalore, Chennai, and Hyderabad.

```
plot_ly(City_Count, labels =~City, values = ~Freq, type = "pie",
        marker = list(colors = c("pink", "lightblue","lightyellow", "lightcyan", "#DDA0DD", "lavender")),
        textinfo = "label+percent",
        hoverinfo = 'text',
        text = ~paste('City = ',City, '\nFreq = ', Freq))%>%
  layout(showlegend = TRUE,
         legend = list(orientation = 'v', x = 1, y = 0.5),
         title = list(text = "Pie Chart for different city", font = list(size = 15)))
```



The pie chart shows that Kolkata and Delhi have only 11% to 12% data in the entire data set, meanwhile the other four city each owns 18% - 20% of the dataset.

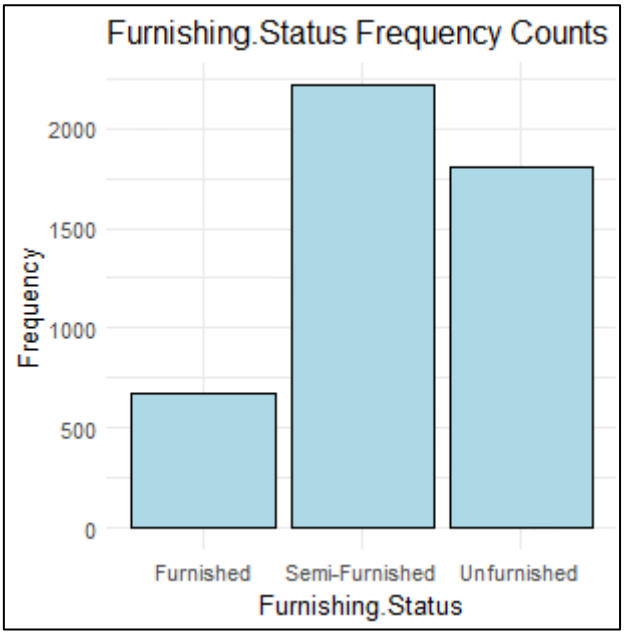
5.3 Data Exploration for Furnishing Status

```
Furnishing.Status_Count = as.data.frame(table(HouseRent_Cleaned$Furnishing.Status))
names(Furnishing.Status_Count)[1] = "Furnishing.Status"
print(Furnishing.Status_Count, row.names = FALSE)
```

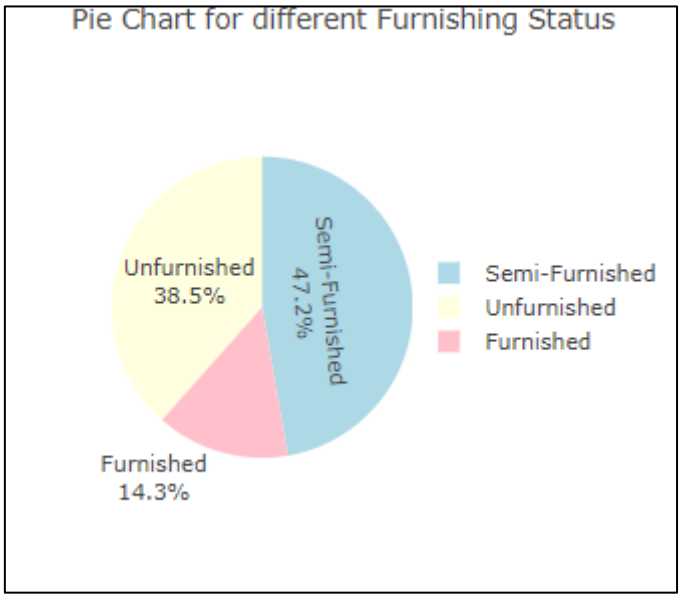
```
Furnishing.Status Freq
        Furnished  670
    Semi-Furnished 2216
      Unfurnished 1804
```

Based on the result, 'Furnished' house have only 670 rows of observation, while 'Semi-furnished' have the highest row, which is 2216 observations.

```
ggplot(Furnishing.Status_Count, aes(x = Furnishing.Status, y = Freq)) +
  geom_bar(stat = "identity", fill = "lightblue", color = "black") +
  labs(title = "Furnishing.Status Frequency Counts",
       x = "Furnishing.Status",
       y = "Frequency") +
  theme_minimal()
```



```
plot_ly(Furnishing.Status_Count, labels =~Furnishing.Status, values = ~Freq, type = "pie",
        marker = list(colors = c("pink", "lightblue","lightyellow")),
        textinfo = "label+percent",
        hoverinfo = 'text',
        text = ~paste('Furnishing.Stauts = ',Furnishing.Status, '\nFreq = ', Freq))%>%
  layout(showlegend = TRUE,
         legend = list(orientation = 'v', x = 1, y = 0.5),
         title = list(text = "Pie Chart for different Furnishing Status", font = list(size = 15)))
```



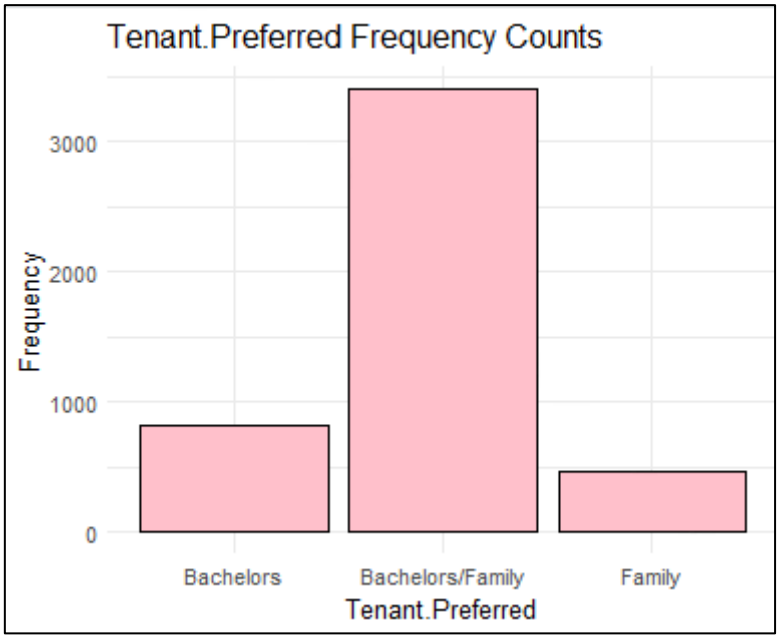
5.4 Data Exploration for Tenant Preferred

```
Tenant.Preferred_Count = as.data.frame(table(HouseRent_Cleaned$Tenant.Preferred))
names(Tenant.Preferred_Count)[1] = "Tenant.Preferred"
print(Tenant.Preferred_Count, row.names = FALSE)
```

```
 Tenant.Preferred Freq
        Bachelors  817
Bachelors/Family 3405
           Family  468
```

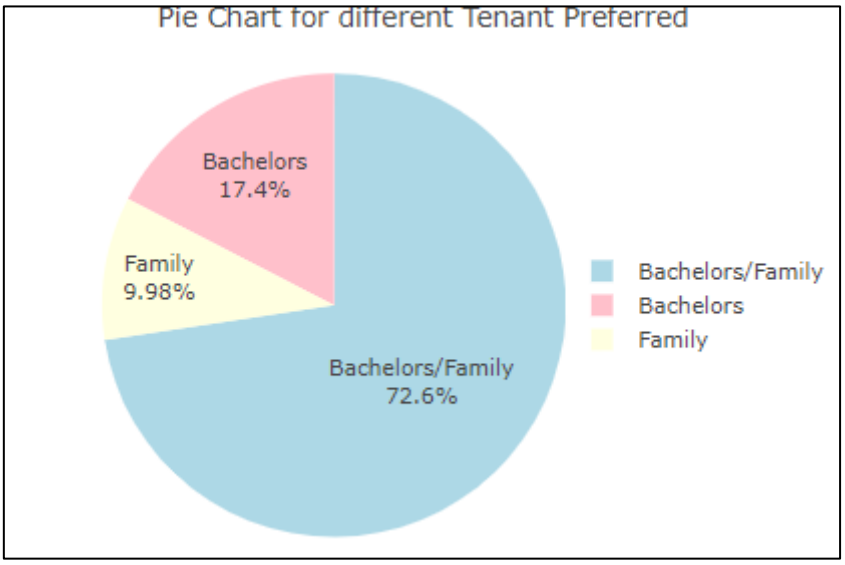
Based on the result, there is a significant difference between house that only specifies ‘Bachelors’ or ‘Family’, and the house that allow both tenants.

```
ggplot(Tenant.Preferred_Count, aes(x = Tenant.Preferred, y = Freq)) +
  geom_bar(stat = "identity", fill = "pink", color = "black") +
  labs(title = "Tenant.Preferred Frequency Counts",
       x = "Tenant.Preferred",
       y = "Frequency") +
  theme_minimal()
```



The bar chart also shows the major difference of data.

```
plot_ly(Tenant.Preferred_Count, labels =~Tenant.Preferred, values = ~Freq, type = "pie",
        marker = list(colors = c("pink", "lightblue","lightyellow")),
        textinfo = "label+percent",
        hoverinfo = 'text',
        text = ~paste('Tenant.Preferred = ',Tenant.Preferred, '\nFreq = ', Freq))%>%
  layout(showlegend = TRUE,
         legend = list(orientation = 'v', x = 1, y = 0.5),
         title = list(text = "Pie Chart for different Tenant Preferred", font = list(size = 15)))
```



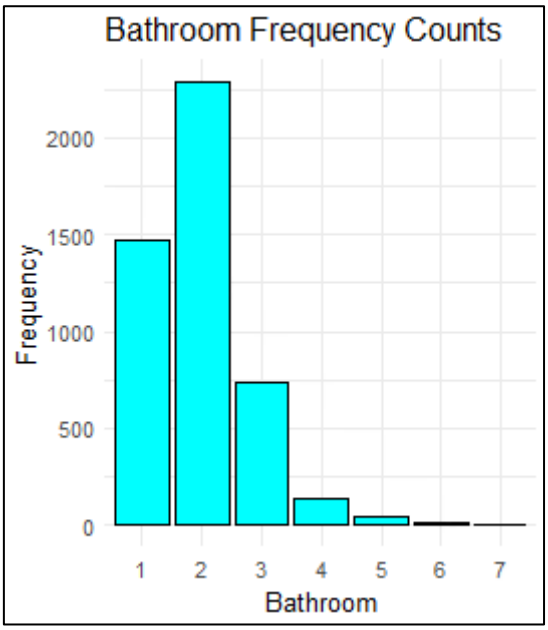
5.5 Data Exploration for Bathroom

```
Bathroom_Count = as.data.frame(table(HouseRent_Cleaned$Bathroom))
names(Bathroom_Count)[1] = "Bathroom"
print(Bathroom_Count, row.names = FALSE)
```

```
Bathroom Freq
       1 1474
       2 2287
       3  740
       4  132
       5   47
       6    8
       7    2
```

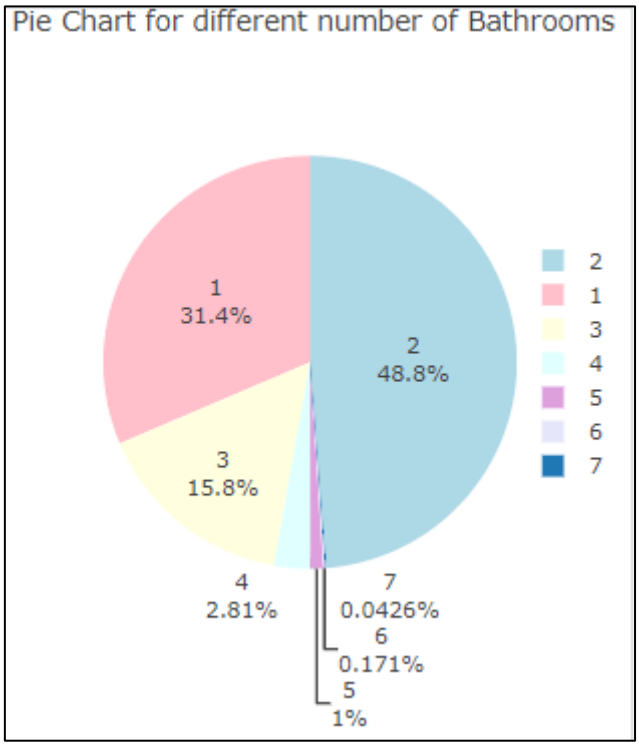
For bathroom, there are 2287 houses that have 2 bathrooms, considered as the most popular. Only 2 houses that own 7 bathrooms.

```
ggplot(Bathroom_Count, aes(x = Bathroom, y = Freq)) +
  geom_bar(stat = "identity", fill = "cyan", color = "black") +
  labs(title = "Bathroom Frequency Counts",
       x = "Bathroom",
       y = "Frequency") +
  theme_minimal()
```



The Bar chart also visualise the frequency of bathroom in each house.

```
plot_ly(Bathroom_Count, labels =~Bathroom, values = ~Freq, type = "pie",
        marker = list(colors = c("pink", "lightblue","lightyellow", "lightcyan", "#DDA0DD", "lavender")),
        textinfo = "label+percent",
        hoverinfo = 'text',
        text = ~paste('Bathroom = ',Bathroom, '\nFreq = ', Freq))%>%
  layout(showlegend = TRUE,
         legend = list(orientation = 'v', x = 1, y = 0.5),
         title = list(text = "Pie Chart for different number of Bathrooms", font = list(size = 15)))
```



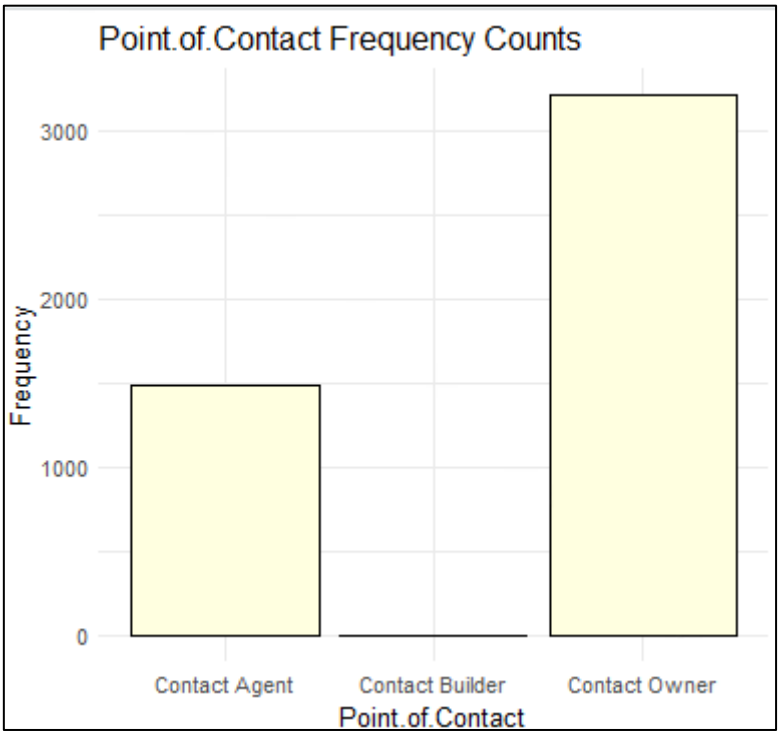
5.6 Data Exploration for Point of Contact

```
Point.of.Contact_Count = as.data.frame(table(HouseRent_Cleaned$Point.of.Contact))
names(Point.of.Contact_Count)[1] = "Point.of.Contact"
print(Point.of.Contact_Count, row.names = FALSE)
```

```
Point.of.Contact Freq
   Contact Agent 1486
 Contact Builder    1
   Contact Owner 3203
```

There is only one ‘Contact Builder’ in the whole data set, follow up with 1486 houses contact with ‘Contact Agent’, and most of the houses requires to contact with ‘Contact Owner’.

```
ggplot(Point.of.Contact_Count, aes(x = Point.of.Contact, y = Freq)) +
  geom_bar(stat = "identity", fill = "lightyellow", color = "black") +
  labs(title = "Point.of.Contact Frequency Counts",
       x = "Point.of.Contact",
       y = "Frequency") +
  theme_minimal()
```



```
plot_ly(Point.of.Contact_Count, labels =~Point.of.Contact, values = ~Freq, type = "pie",
        marker = list(colors = c("pink", "lightblue","lightyellow", "lightcyan")),
        textinfo = "label+percent",
        hoverinfo = 'text',
        text = ~paste('Point.of.Contact = ',Point.of.Contact, '\nFreq = ', Freq))%>%
  layout(showlegend = TRUE,
         legend = list(orientation = 'v', x = 1, y = 0.5),
         title = list(text = "Pie Chart for different point of contact", font = list(size = 15)))
```