Assumption:

Some major steps should be used as a guide and key for performing the data analysis. The first step is importing and loading the packages that will be used during the assignment, second is Data collection which is the data that should be worked with, A House Rent Dataset of more than 4700 rows was given as a database in the assignment. The third step is Data Cleaning which should clean the data set of any missing values, Next step is Data Pre-Processing where the data has to be fully prepared for Data Analysis. The last step is Data Analysis (Exploration, Visualization, and manipulation) where the process of making a decision about the future or the past depends on the analysis will be done.

Contents

1.0 Introduction:

R is a software environment and programming language for analyzing data and implementing graphics. R language comes in handy when it comes to processing a huge set of data that is hard for individuals to process. R language was founded in 1995 and has a decent number of libraries and functions that help for analyzing the data and make it one of the best environments for statistical computing and design. R studio and R language were the tools utilized for this assignment.

2.0 Data Import

2.1 Load Library

```
## load libraries
library(stringr)
library(ggplot2)
library(plotly)
library(lubridate)
library(tidyverse)
library(dplyr)
library(hrbrthemes)
library(viridis)
library(corrplot)
```

Figure 2 1

All the libraries needed in this assignment are imported, library is a group of functions built together to make the r environment much easier to work with. In this assignment, a few libraries are used to help us analyze the data and do pre-processing and visualization.

Github: aimancsv

2.2 Load Data

```
## Import the data
RentalDataset = read.csv("D:\\Desktop\\House_Rent_Dataset.csv"
                         , header = TRUE)
```

Figure 2 2

The above code is used to import the database with the header. The dataset provided in this assignment includes information about various house rents in India.

The function View() is used to show the entire dataset:

```
# View the Dataset in a table form
View(RentalDataset)
```

Figure 2 3

	Posted.On	BHK	Rent	Size	Floor	Area.Type	Area.Locality	City
1	5/18/2022	2	10000	1100	Ground out of 2	Super Area	Bandel	Kolkata
2	5/13/2022	2	20000	800	1 out of 3	Super Area	Phool Bagan, Kankurgachi	Kolkata
3	5/16/2022	2	17000	1000	1 out of 3	Super Area	Salt Lake City Sector 2	Kolkata
4	7/4/2022	2	10000	800	1 out of 2	Super Area	Dumdum Park	Kolkata
5	5/9/2022	2	7500	850	1 out of 2	Carpet Area	South Dum Dum	Kolkata
6	4/29/2022	2	7000	600	Ground out of 1	Super Area	Thakurpukur	Kolkata
7	6/21/2022	2	10000	700	Ground out of 4	Super Area	Malancha	Kolkata
8	6/21/2022	1	5000	250	1 out of 2	Super Area	Malancha	Kolkata

Figure 2 4

The dataset interfaces how tenants may choose the house based on their lifestyles and backgrounds such as the city, the number of bedrooms, and the price the tenants can afford. The dataset contains 4747 rows and 12 columns. New headers were assigned for the column, the original columns and the assigned columns were given below with the descriptions.

Original Column(s)	Assigned Column(s)	Descriptions
Posted On	Published_date	The date when the status was published
BHK	BHK	The number of bedrooms, Hall and kitchen

Rent	Price	The price of the house rented
Size	Square_Feet	The size of the house in Square Feet
Floor	Floor_Situated	Total Number of Floors and which floor is the house situated
Area Type	Area_Class	The area type is either Build or Carpet or Super Area
Area Locality	Location	The Location of the house
City	City	The City of the house
Furnishing Status	Furnishing_Types	The furnishing types whether Unfurnished or Semi Furnished or Furnished
Tenant Preferred	Renter_Preferred	The renter preferred whether Families or Bachelors
Bathroom	Bathroom	The number of bathrooms
Point of Contact	Contact_Type	The renter preferred whether contact with the Owner or Agent

3.0 Pre-processing, Cleaning, and Transformation

3.1 Assigning new columns header names:

```
# Assign header names
names(RentalDataset)=c("Published_date","BHK","Price",
                       "Square_Feet","Floor_Situated",
                       "Area_Class","Location","City",
                       "Furnishing_Types","Renter_Preferred",
                       "Bathroom","Contact_Type")
```

Figure 3. 1

The attached code is used to replace the original header names with new names without affecting the database. The function names() is used to assign new headers and show the header names for a dataset in the console.

```
> names(RentalDataset)
 [1] "Published_date"   "BHK"              "Price"
       "Square_Feet"      "Floor_Situated"
 [6] "Area_Class"       "Location"         "City"
       "Furnishing_Types" "Renter_Preferred"
[11] "Bathroom"         "Contact_Type"
```

Figure 3. 2

3.2 Split columns

	Published_date	BHK	Price	Square_Feet	Floor_Situated	Area_Class	Location
1	5/18/2022	2	10000	1100	Ground out of 2	Super Area	Bandel
2	5/13/2022	2	20000	800	1 out of 3	Super Area	Phool Bagan, Kankurgachi
3	5/16/2022	2	17000	1000	1 out of 3	Super Area	Salt Lake City Sector 2
4	7/4/2022	2	10000	800	1 out of 2	Super Area	Dumdum Park
5	5/9/2022	2	7500	850	1 out of 2	Carpet Area	South Dum Dum
6	4/29/2022	2	7000	600	Ground out of 1	Super Area	Thakurpukur
7	6/21/2022	2	10000	700	Ground out of 4	Super Area	Malancha
8	6/21/2022	1	5000	250	1 out of 2	Super Area	Malancha

Figure 3. 3

As shown in the figure above this is the new dataset after changing the headers, there is a column called Floor Situated which provides information about which flood does the renter live on and how many floors the house have.

```
> class(RentalDataset$Floor_Situated)
[1] "character"
```

Figure 3. 4

Using the function class() we can determine which data structure this value, the data in these columns are in a character class. To avoid logical errors and make it easier to work with, the data will be split into two separate columns which will be Floor_Number and Total_Floor.

```
# Split Floor_Situated column into Floor_Number and Total_Floor
RentalDataset[c('Floor_Number', 'Total_Floor')] <-
  str_split_fixed(RentalDataset$Floor, ' out of ', 2)
```

Figure 3. 5

Using the split function, the column was divided successfully, and the split word was “out of” which is the word between the floor number and the total floor.

Floor_Number	Total_Floor
Ground	2
1	3
1	3
1	2
1	2
Ground	1
Ground	4
1	2

Figure 3. 6

```
# Rearrange columns and remove original Floor_Situated column
RentalDataset <- RentalDataset[c('Published_date', 'BHK',
                                 'Price', 'Square_Feet',
                                 'Floor_Number', 'Total_Floor',
                                 'Area_Class', 'Location', 'City',
                                 'Furnishing_Types', 'Renter_Preferred',
                                 'Bathroom', 'Contact_Type')]

# Replace the value Ground with '0'
RentalDataset[RentalDataset == "Ground"] <- 0
```

Figure 3. 7

After dividing the columns, the attached code above is to rearrange the columns and remove the original columns, however, the Ground floor value should be converted to 0 to be able to change the column data structure from character to numeric class.

Floor_Number	Total_Floor
0	2
1	3
1	3
1	2
1	2
0	1
0	4
1	2

Figure 3. 8

The Floor_Number and Total_Floor columns after splitting the original columns and renaming the values Ground with 0.

3.3 Convert Data type

```
> str(RentalDataset)
'data.frame':   4746 obs.
 $ Published_date  : chr
 $ BHK             : int
 $ Price           : int
 $ Square_Feet     : int
 $ Floor_Number    : chr
 $ Total_Floor     : chr
 $ Area_Class      : chr
 $ Location        : chr
 $ City            : chr
 $ Furnishing_Types: chr
 $ Renter_Preferred: chr
 $ Bathroom        : int
 $ Contact_Type    : chr
```

Figure 3. 9

As it is shown in the above Image, using the function str() allows checking the data type for all the columns in a dataset.

	Published_date	Floor_Number	Total_Floor
1	5/18/2022	0	2
2	5/13/2022	1	3
3	5/16/2022	1	3
4	7/4/2022	1	2
5	5/9/2022	1	2
6	4/29/2022	0	1
7	6/21/2022	0	4

Figure 3. 10

Published_date, Floor_Number, and Total_Floor are the columns that need to be converted to make it easier and more efficient to analyze them.

3.3.1 Convert character to numeric data type

```
#Convert Floor_Number and Total_Floor Columns to Numeric format
RentalDataset$Floor_Number = as.numeric(as.character
                                        (RentalDataset$Floor_Number))
RentalDataset$Total_Floor = as.numeric(as.character
                                       (RentalDataset$Total_Floor))
```

Figure 3. 11

```
> str(v)
'data.frame':   4746
 $ Floor_Number: num
 $ Total_Floor : num
```

Figure 3. 12

Using the code above will allow converting Floor_Number and Total_Floor to numeric data type by using as. numeric() function.

Convert character to date data type

```
#Convert Published_date Column to Date/Time format
RentalDataset$Published_date <- as.POSIXct(
  RentalDataset$Published_date, format="%m/%d/%y" )
```

Figure 3. 13

```
> str(v)
'data.frame':   4746 obs. of  2 variables:
 $ Published_date: POSIXct, format: "2020-05-18"
```

Figure 3. 14

By using the code above, the data type has converted to Date/ Time format, It allows to a change of the data format as well, as it shows the format chosen in the code is “%m/%d%y”.

3.4 Dealing with Missing Values

Before starting to do any analysis, the dataset should not have any missing values that might affect the data analysis process.

```
# Checking missing values
sapply(RentalDataset, function(x) sum(is.na(x)))
```

Figure 3. 15

```
> sapply(RentalDataset, function(x) sum(is.na(x)))
  Published_date              BHK            Price      Square_Feet     Floor_Number
               0                0                0                0               34
     Total_Floor       Area_Class         Location             City Furnishing_Types
               4                0                0                0                0
Renter_Preferred         Bathroom     Contact_Type
               0                0                0
```

Figure 3. 16

As shown in the above figure, using the function sapply() will check all the missing values and show in which columns are the missing values and how the number of the missing values was found. In this dataset, there are 34 missing values in the Floor_Number column and 4 missing values in the Total_Floor columns. To deal with the missing values, there are a few techniques:

The first technique is by removing

```
# Drop rows with missing values
RentalDataset <- na.omit(RentalDataset)
```

Figure 3. 17

```
> sapply(RentalDataset, function(x) sum(is.na(x)))
  Published_date              BHK            Price      Square_Feet     Floor_Number
               0                0                0                0                0
     Total_Floor       Area_Class         Location             City Furnishing_Types
               0                0                0                0                0
Renter_Preferred         Bathroom     Contact_Type
               0                0                0
```

Figure 3. 18

To deal with the missing values, there are a few techniques:

The first technique that can be used is by using kNN() function located in VIM package, It examines the nearest neighbor and this technique will replace the missing values with the most accurate and similar one in the database.

The technique used in this code above is by dropping the entire row that has missing values by using the name. omit() function, this technique can only be used if the missing values are less than 20% of the dataset. The reason for using this technique is the missing data is floor number and total floor which cannot be replaced with other similar data.

3.5 Outlier Analysis

Outlier Analysis is used to check if the data or values are lying away from the dataset if some data have a huge distance from the other data, which may affect the accuracy of the data analysis, for this reason, these data should be processed before starting any analysis.

```
# Check Outlier (data that are lie away from the other points)
boxplot(RentalDataset$Price, main = "Price Outlier")
boxplot(RentalDataset$Square_Feet, main = "Square Feet Outlier")
boxplot(RentalDataset$Floor_Number, main = "Floor Number Outlier")
boxplot(RentalDataset$Total_Floor, main = "Total Floor Outlier")
boxplot(RentalDataset$BHK, main = "BHK Outlier")
```

Figure 3. 19

The first step is to check the Outlier data, the above code is to check if any data points lie away from others using the plot() function to visualize the points and get a better vision of the Outlier data.

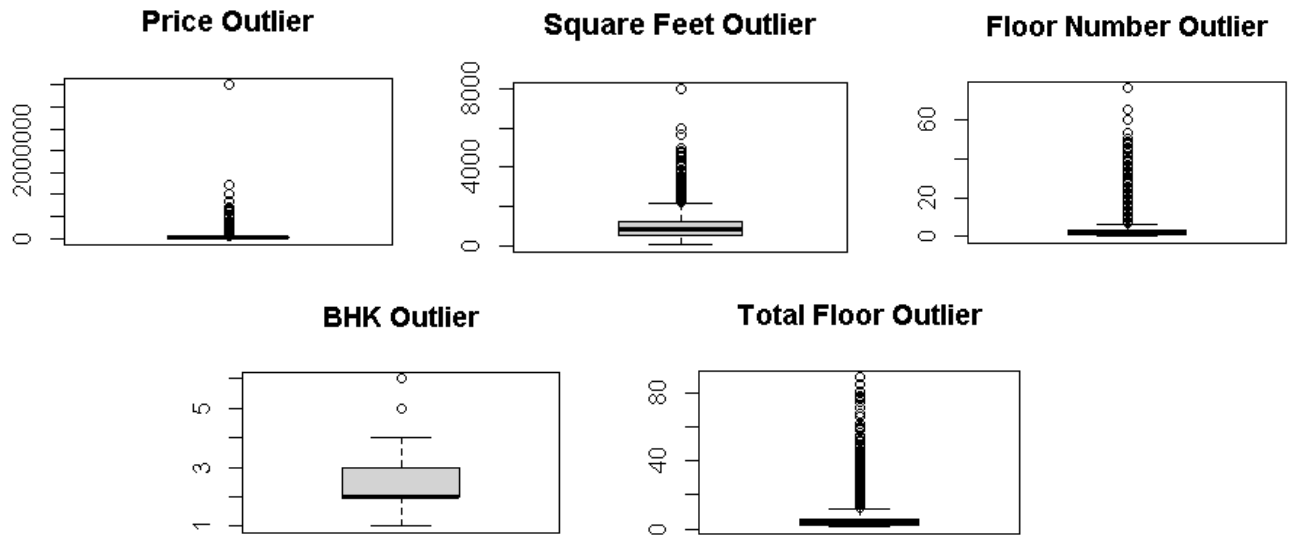

Figure 3. 20

As shown in the images above, there is a clear vision where the points lie away from the other points in Price, Square Feet, and Floor Number Outliers.

```
# Outlier Implementation
# Price Outlier
lowwerPrice <- quantile(RentalDataset$Price, 0.010)
upperPrice <- quantile(RentalDataset$Price, 0.990)
RentalDataset <- subset(RentalDataset, RentalDataset$Price >
                        lowwerPrice & RentalDataset$Price < upperPrice)

# Square_Feet Outlier
lowwerSquare_Feet <- quantile(RentalDataset$Square_Feet, 0.010)
upperSquare_Feet <- quantile(RentalDataset$Square_Feet, 0.990)
RentalDataset <- subset(RentalDataset, RentalDataset$
                        Square_Feet > lowwerSquare_Feet &
                        RentalDataset$Square_Feet < upperSquare_Feet)

# Floor_Number Outlier
lowwerFloor_Number <- quantile(RentalDataset$Floor_Number, 1)
UpperFloor_Number <- quantile(RentalDataset$Floor_Number, 0.990)
RentalDataset <- subset(RentalDataset, RentalDataset$Floor_Number <
                        lowwerFloor_Number & RentalDataset$Floor_Number <
                        UpperFloor_Number)
```

Figure 3. 21

As shown in the figure above the implementations for Outlier are coded using quantile() and subset() functions, quantile() function is applied to generate probability within a range of [0 to 1]. The range used in the implementations was specified based on the columns' data. subset() function is used to choose data from the dataset that fits a specific need.

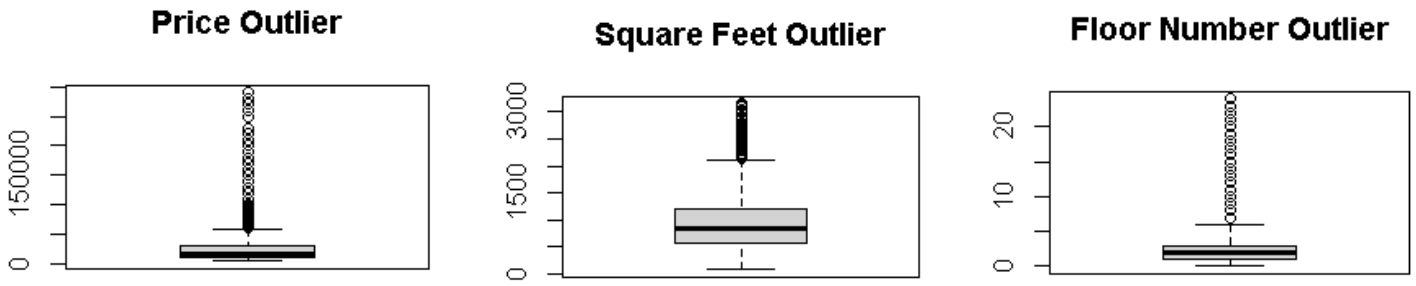

Figure 3. 22

After the implementations for Outlier are done, There are no more data points that lie away from other points, which will make the dataset analysis more accurate and efficient,, the dataset is ready for data Exploration and Visualization.

3.6 Data Correlation

```
cor_matrix<-cor(RentalDataset[,c(2,3,4,12,10)])
str(RentalDataset)
cor_matrix
```

Figure 3. 23

Data Correlation is used to display the correlation and relationship between vectors.

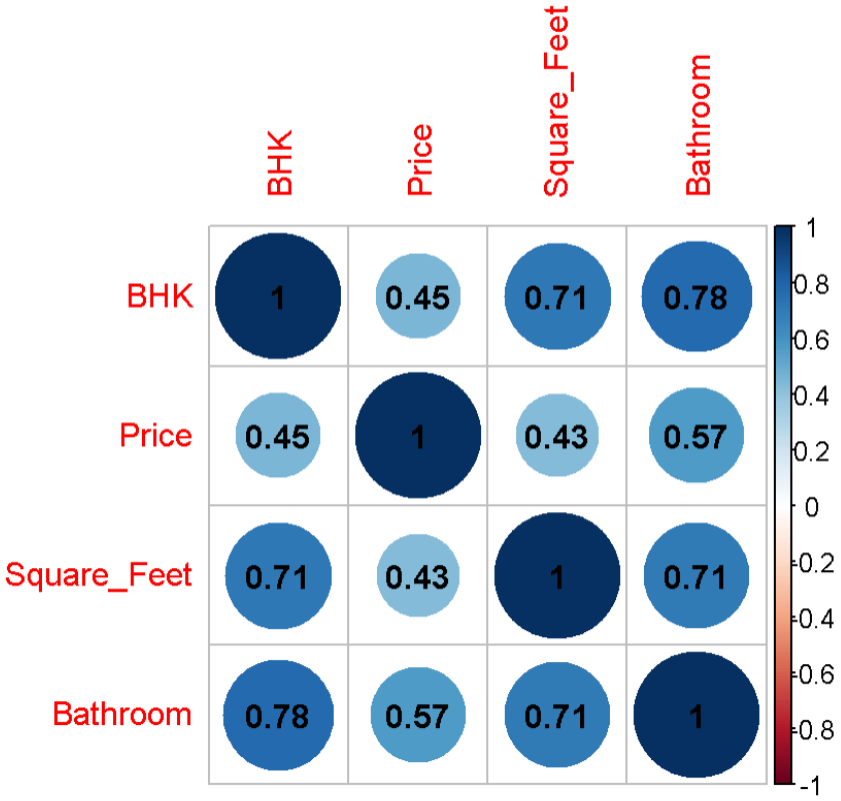

Figure 3. 24

4.0 Question 1 Renting houses on monthly basis

4.1 Analysis: Number of Bachelors/Families rented monthly

```
ggplot(aes(x= factor(Published_date, levels = c("Apr","May", "Jun","Jul")),
           fill = Renter_Preferred))+
geom_bar(position = "dodge", alpha = 0.5)+
theme_bw()+
theme(panel.grid.major = element_blank(),
      panel.grid.minor = element_blank())+
scale_fill_manual(values=c('blue', "green","red"))+
labs(title = "Number of Bachelors/Family rented monthly",
     x = "Month",
     y = "Number of houses rented",
     fill = "Renter type",
     )
```

Figure 4. 1

The above code is to calculate the number of Bachelors and Families rented in Apr, May, Jun, and Jul. Starting by assigning the needed columns in a different variable so it doesn't affect the rest of the data. the columns Published_date and Renter_Preferred are subsetted. geom_bar is used from ggplot to visualize the data.

Output:

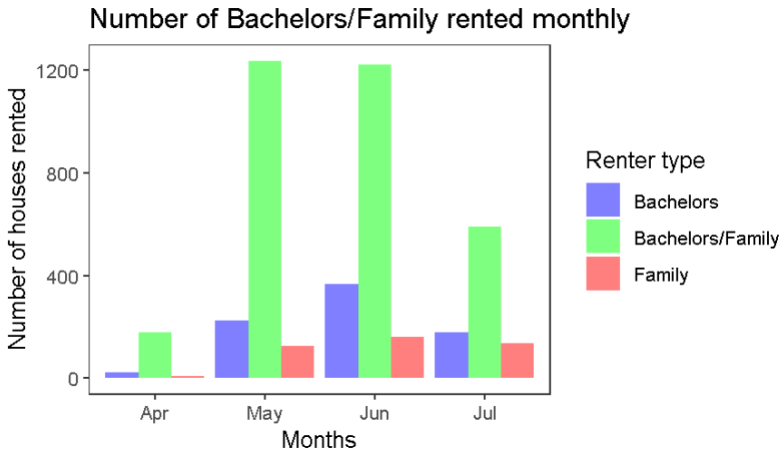

Figure 4. 2

Finding:

- Most of the renter types preferred both bachelors and families
- Bachelors type are more likely to rent houses than families
- There is no direct relationship between the date and the increase in the number of house rents

4.2 Analysis: Number of all families rented monthly.

```
mutate(Renter_Preferred = recode(Renter_Preferred,
                                 "Bachelors/Family" = "Family"))%>%
filter(Renter_Preferred %in% ("Family"))%>%
```

Figure 4. 3

Almost the same concept of analysis 1.1 is used here as well, reorder() function is used to get all the bachelor data by converting all the Bachelors/ family data to family, Followed by the function filter() to filter all the values in the dataset that have Bachelors/ family.

Output:

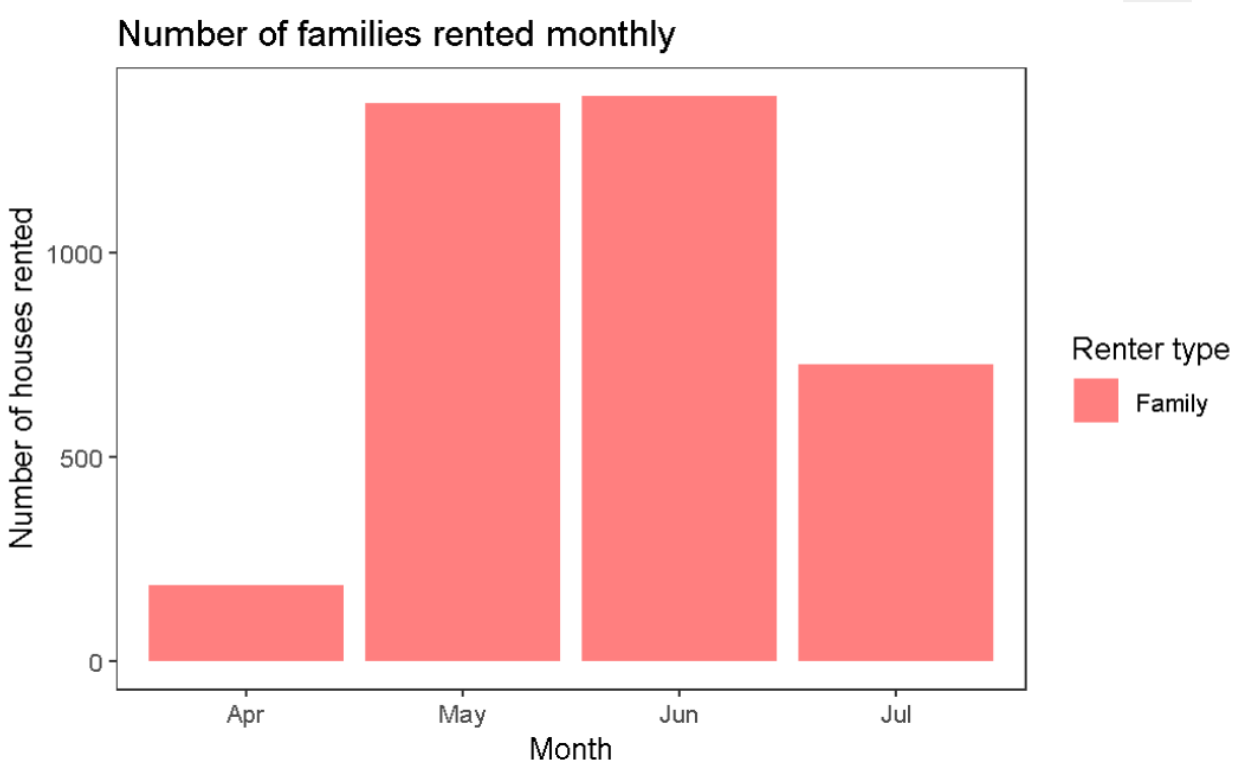

Figure 4. 4

Finding:

- Most of the rented houses by families are recorded in June
- The least number of rented houses by families are recorded in April
- May and June are the most periods when families rent houses

4.3 Analysis: Number of all bachelors rented monthly.

```
mutate(Renter_Preferred = recode(Renter_Preferred,
                                 "Bachelors/Family" = "Bachelors"))%>%
filter(Renter_Preferred %in% ("Bachelors"))%>%
```

Figure 4. 5

The function recode() and filter() are used here as well to get all the bachelors values from the dataset.

Output:

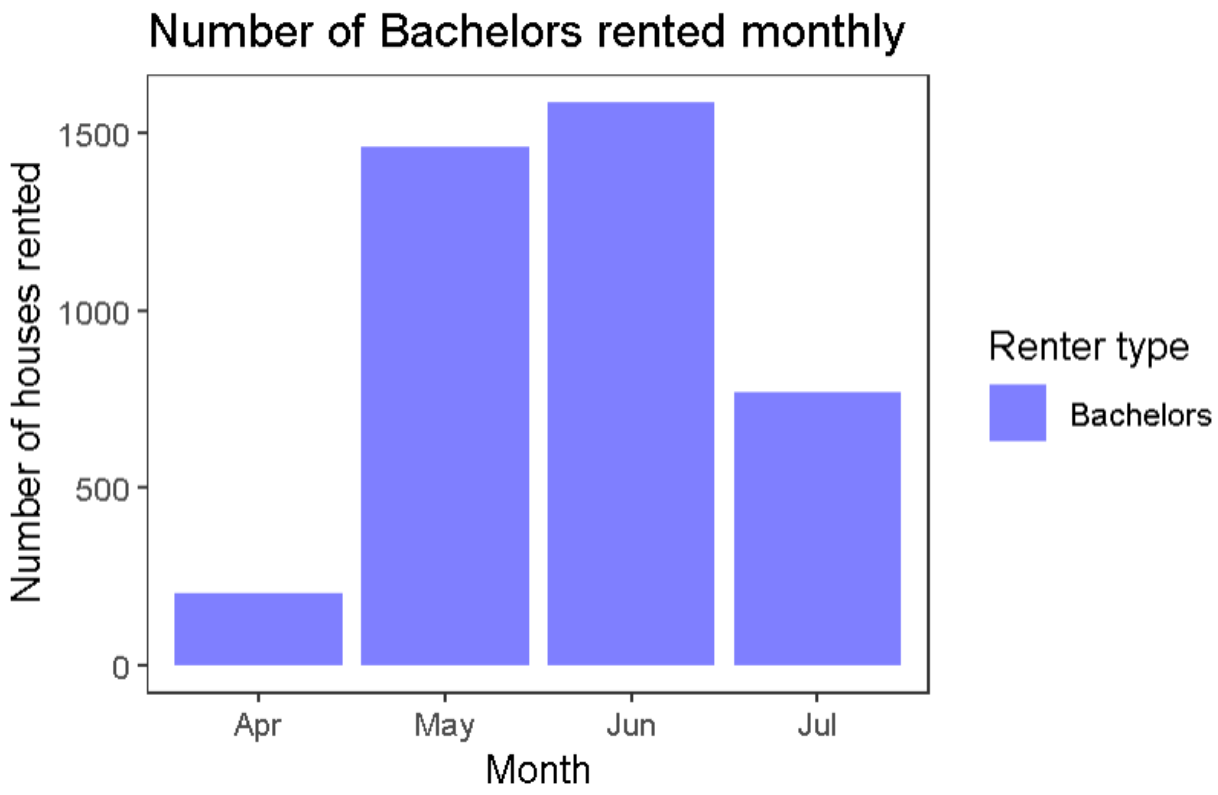

Figure 4. 6

Finding:

- Most of the rented houses by bachelors are recorded in June
- The least number of rented houses by bachelors is recorded in April
- May and June are the most periods when bachelors rent houses

4.4 Analysis: Average price monthly

```
anal1_4 = subset(RentalDataset,
                 select = c("Published_date","Price" ))
  subset(RentalDataset, select = c("Published_date","Price" ))%>%
  group_by(Published_date = month(as.POSIXlt(Published_date,
  summarise(Count=mean(Price))%>%
  tail(4) %>%
  ggplot(aes(x= Published_date , y = Count))+
  geom_line(color="#69b3a2", size=2, alpha=0.9) +
  geom_point(shape=21, color="red", fill="#69b3a2", size=7) +
  theme_ipsum() +
    labs(title = "Average price monthly",
         x = "Month",
         y = "Average")
```

Figure 4. 7

The above code is to calculate the mean of the price monthly, starting by subsetting the needed columns which are the Published_date and Price, the plot used in the analysis is geom_line to show a clear visualization of the data.

Output:

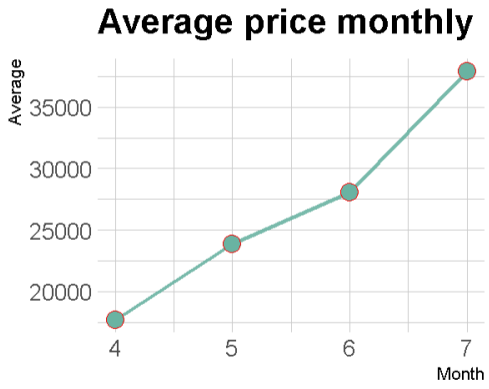

Figure 4. 8

Findings:

- The average price is increasing monthly
- There is a direct relationship between the average price and the months
- The cheapest rental period was in the month of 4
- The most expensive period rental period was in the month of 7

4.5 Analysis: Average rent by furnishing types

```
anal1_5<-RentalDataset%>%
  group_by(Furnishing_Types)%>%
  summarise(Avg_rent = mean(Price))%>%
  ggplot(aes(Furnishing_Types, Avg_rent, fill = Avg_rent))+
  geom_bar(stat='Identity') +
  geom_text(aes(label = signif(Avg_rent)),vjust = 1.5, color = "white")+
  labs(title='Average Rent by Furnishing Types', y = ' Average Rent')+
  theme_bw()
```

Figure 4. 9

The code above calculates the average rent by furnishing type whether furnished, semi-furnished, or unfurnished, starting by choosing the column needed which are Furnished_Type and Price, and assigning them to a variable, the plot used is geom_bar and the function signif() is used to display the values on top of the bars.

Output:

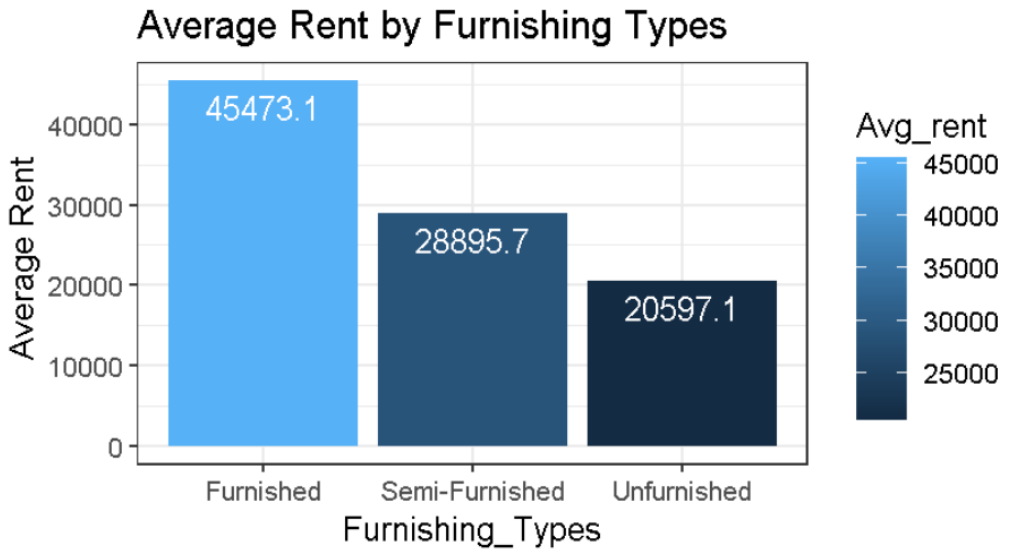

Figure 4. 10

Findings:

- The average rent for the Furnished type is the highest average of 45473.
- The semi-furnished has the middle average price of 28895.
- The lowest average rent price is for the Unfurnished of 20597.

4.6 Analysis: Number of houses rented monthly by furnishing types

```
anal1_6 = subset(RentalDataset, select = c("Published_date","Furnishing_Types" ))
anal1_6$Published_date <- format(as.POSIXct.Date(anal1_6$Published_date,
                                                 format="%m/%d/%Y"),"%b")
anal1_6 %>%
  ggplot(aes(x= factor(Published_date, levels = c("Apr","May", "Jun","Jul")),
             fill = Furnishing_Types))+
  geom_bar(position = "dodge", alpha = 0.5)+
  coord_flip()+
  theme_bw()+
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())+

  scale_fill_manual(values=c('brown', "gray","black"))+
  labs(title = "Number of houses rented monthly by furnishing types",
       x = "Months",
       y = "Count",
       fill = "Furnishing Types")
```

Figure 4. 11

The code above is to display the number of houses rented by the type of house, Starting with subsetting the needed columns: Published_date and Furnishing_Types. The format() function is used to get only the month from Publishid_date, the reason behind that is when it comes to displaying the values in the plot panel, it will be more clear and easy to see the differences. In this plot, the coord_flip() function is used to flip the analysis and give a new look to the data.

Output:

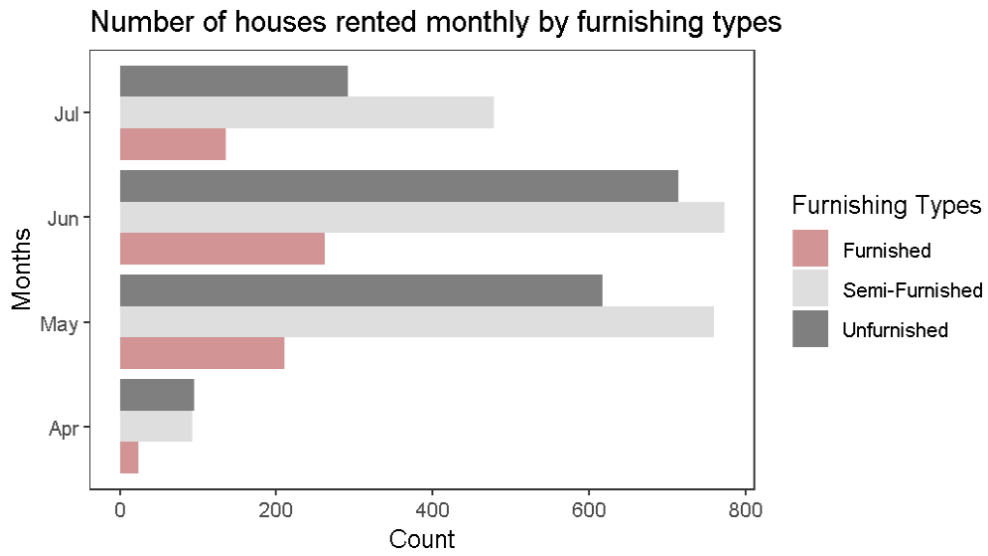

Figure 4. 12

Findings:

- Tenants mostly prefer to rent Semi furnished houses
- The most rented house is recorded in Jun
- Most of the tenants don't prefer to rent furnished houses

4.7 Analysis: When renters usually rent from agemts

```
anal1_7 %>%
  group_by(Published_date = format(Published_date, "%B"),Contact_Type )%>%
  summarise(Count = length(Contact_Type))%>%
  subset( Contact_Type != "Contact Builder")%>%
  view()%>%
  ggplot(aes( x= factor(Published_date, levels = c("April","May", "June","July")),
              y=Count, fill=Contact_Type)) +
  geom_bar(position="dodge", stat="identity") +
  scale_fill_viridis(discrete = T) +
  ggtitle("Renters Contact Type") +
  theme_ipsum() +
  xlab("Month")
```

Figure 4. 13

The code above is to display when renters are usually renting from agents and owners, the geaom_bar is used here to display the data in bars format with dodge position to make the bars next to each other.

Output:

	Published_date	Contact_Type	Count
1	April	Contact Agent	1
2	April	Contact Owner	210
3	July	Contact Agent	458
4	July	Contact Owner	447
5	June	Contact Agent	573
6	June	Contact Owner	1175
7	May	Contact Agent	343
8	May	Contact Owner	1245

Figure 4. 14

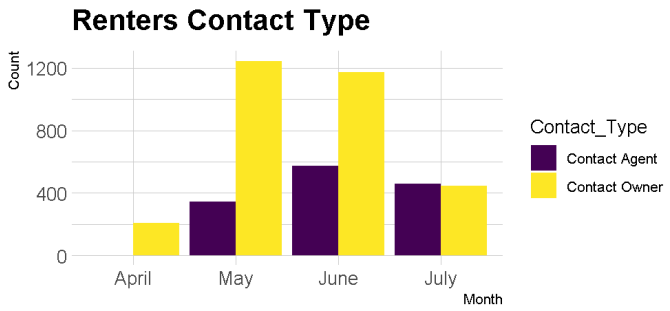

Figure 4. 15

Findings:

- Most renters contact agents in July.
- In April, renters prefer contacting Owners.
- In May and June, most renters are more likely to contact owners

4.8 Analysis: Which month has the most BHK in the carpet area

```
anal1_8 %>%
  subset(Area_Class == "Carpet Area")%>%
  group_by(Published_date = format(Published_date, "%B"),BHK, Area_Class)%>%
  summarise(Count = length(Contact_Type))%>%
  ggplot(aes( x= BHK,
              y=Count,fill = Area_Class)) +
  geom_bar(position="stack", stat="identity") +
  facet_wrap(~factor(Published_date, levels = c("April","May", "June","July"))) +
  ggtitle("Number of BHK in Carpet Area") +
  theme_ipsum() +
  scale_fill_viridis(discrete = T) +
  xlim(1:6,main=' ')+
  xlab("BHK")
```

Figure 4. 16

The analysis above is to display which month has the most BHK in Carpet Area, the function subset() is used to filter the data and get only the rows that have the Carpet area. geom_bar is used to display the analysis in a bar style.

Output:

	Published_date	BHK	Area_Class	Count
4	April	4	Carpet Area	2
5	July	1	Carpet Area	101
6	July	2	Carpet Area	233
7	July	3	Carpet Area	200
8	July	4	Carpet Area	32
9	June	1	Carpet Area	218
10	June	2	Carpet Area	403

Figure 4. 17

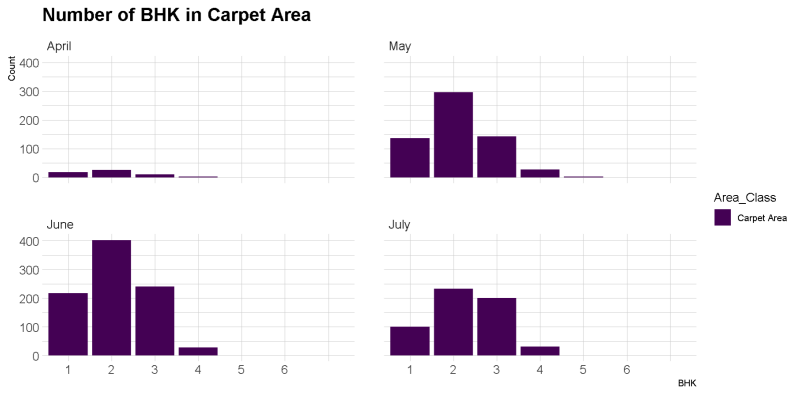

Figure 4. 18

Findings:

- The most number of BHKs in the Carpet Area is found in June
- The less number of BHK in the Carpet Area is found in April
- Most of the houses have two BHKs in the Carpet Area

4.9 Analysis: Which month has the cheapest price and four BHKs in the Super Area

```
anal1_9 %>%
  subset(Area_Class == "Super Area")%>%
  group_by(Published_date = format(Published_date, "%B"),BHK, Area_Class)%>%
  summarise(Avg_Price = mean(Price))%>%
  view()%>%
  ggplot(aes( x= BHK,
              y=Avg_Price,fill = Area_Class)) +
  geom_point(color = "white", size = 3) +
  facet_wrap(~factor(Published_date, levels = c("April","May", "June","July"))) +
  ggtitle("Avarege price and most BHK in the Super Area") +
  theme_dark() +
  scale_fill_viridis(discrete = T) +
  xlim(1:6,main='')+
  xlab("BHK")
```

Figure 4. 19

The code above checks the BHKs and the super area based on the price, geom_pont() is used here to display the result in points style

Output:

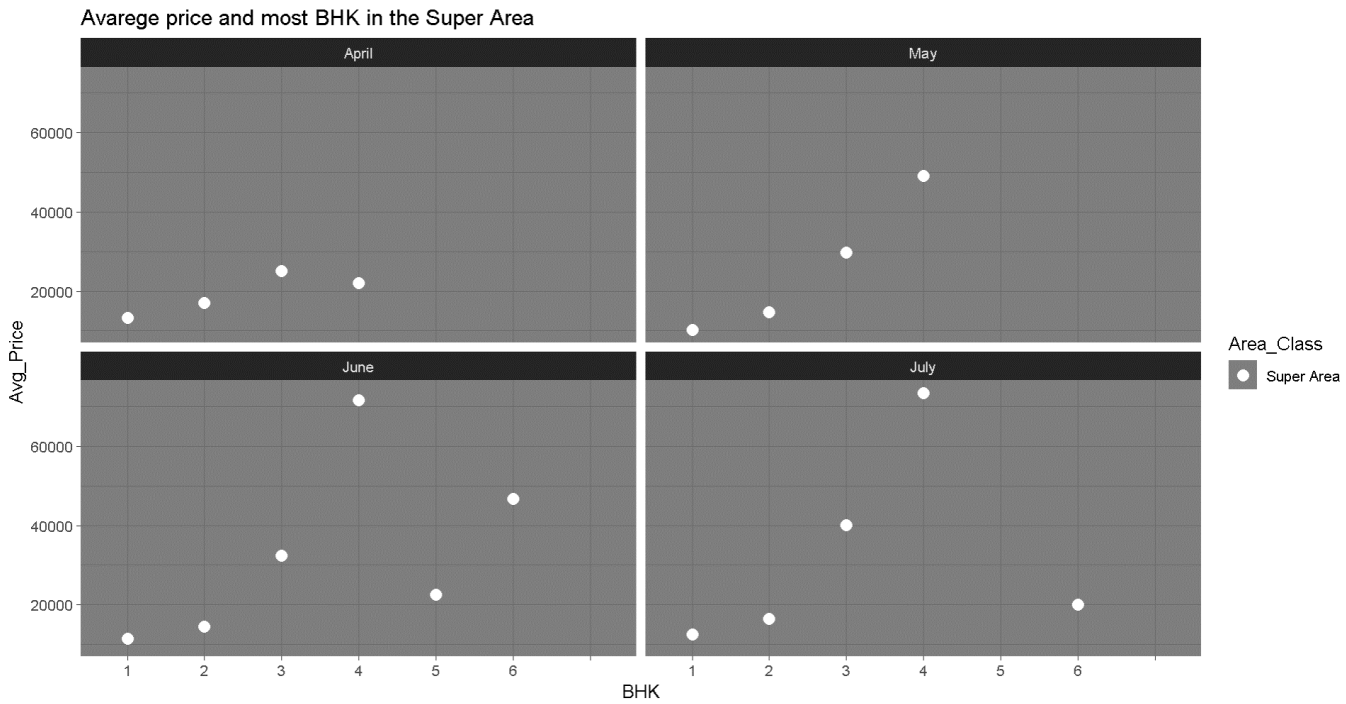

Figure 4. 20

Findings:

- April is the cheapest month to rent a house with four BHKs with a super area
- June is the most expensive month to rent a house with four BHKs with a super area

4.10 Analysis When do cities have the cheapest price for Carpet areas and have three bathrooms?

```
anal1_10 %>%
  subset(Area_Class == "Carpet Area")%>%
  subset(Bathroom >= 3)%>%
  group_by(Published_date = format(Published_date, "%B"),
           Bathroom, Area_Class, City)%>%
  summarise(Avg_Price = mean(Price))%>%
  view()%>%
  ggplot(aes( x=Bathroom ,
              y=Avg_Price, fill = City)) +
  geom_bar(position="dodge", stat="identity") +
  facet_wrap(~factor(Published_date, levels = c("April","May", "June","July"))) +
  ggtitle("Avarege price and more than three bathrooms in Carpet Areas each month") +
  scale_fill_viridis(discrete = T) +
  theme_bw() +
  xlab("Bathroom")
```

Figure 4. 21

The code above is to display the cities that have three bathrooms at least and they have to be in a Carpet area. The function subset() is used to filter the Area class data and the Bathroom data from any unwanted data. geom_bar is used to display the data in a bar shape.

Output:

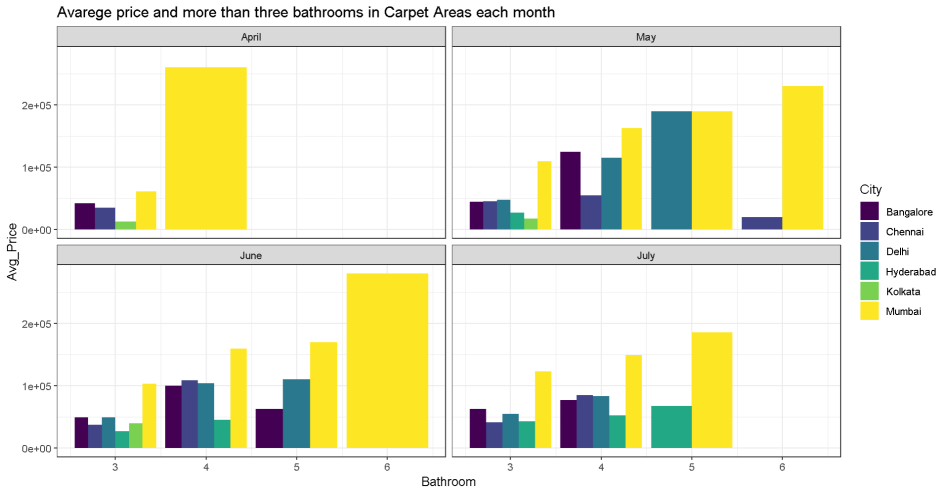

Figure 4. 22

Findings:

- The cheapest city and month in Carpet Area to find a house with more than three bathrooms is Hyderabad in April
- The most expensive city and month in Carpet Area to find a house with more than three bathrooms is Mumbai in June
- Most of the houses in the Carpet Area with a good have three bathrooms

4.11 Analysis (conclusion)

```
anal1_11 = RentalDataset %>%
  group_by(Published_date = format(Published_date, "%B"),BHK,Price,Area_Class,
                                   Furnishing_Types)%>%
  summarise(Avg_Price = mean(Price))%>%
  ggplot (anal1_11,aes(BHK,Avg_Price,size =Area_Class, color =Furnishing_Types ))+
  geom_point()+
  facet_wrap (~factor(Published_date, levels = c("April","May", "June","July")))+
  labs(title = "Month described by Price and other factors",
       x = "BHK",
       y = "Avarage Price")
```

Figure 4. 23

Output:

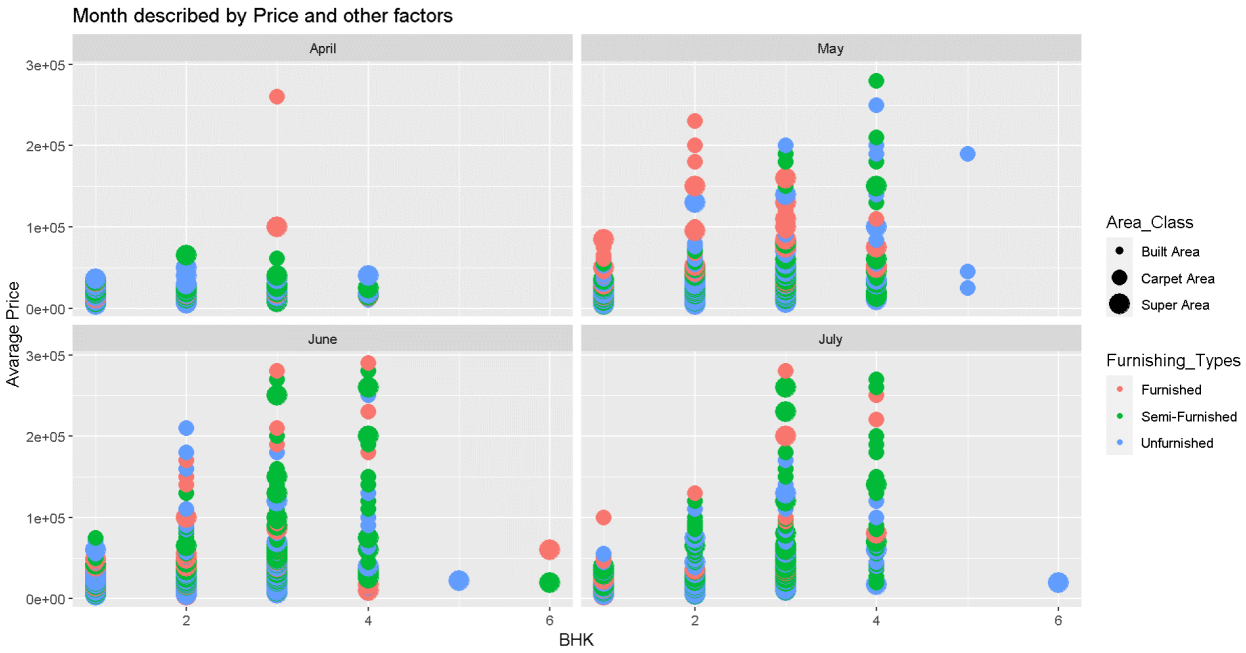

Figure 4. 24

Conclusion:

is increasing every month so the cheapest rent price was found in April, as expected furnished houses are the most expensive type but tenants are mostly renting semi-furnished houses. Most renters contact agents in July. and in April, renters prefer contacting Owners. For renters who prefer renting houses with more BHKs, • the most number of BHKs in the Carpet Area is found in June, and the less number of BHKs was found in April, but April is the cheapest month to rent a house with four BHKs in a super area while June was the most expensive month with four BHKs, to be more specific, the cheapest city and month in Carpet Area to find a house with more than three bathrooms is Hyderabad in April while the most expensive is in Mumbai in June.

5.0. Question 2: Distribution of what Bachelors/Families prefer:

5.1 Analysis: What Area Class do Bachelors and families prefer?

```
anal2_1 = RentalDataset %>%
  group_by(Area_Class)%>%
  filter(Renter_Preferred %in% c("Bachelors"))%>%
  count()%>%
  ungroup()%>%
  mutate(Percentage = `n`/sum(`n`))%>%
  arrange(desc(Area_Class))

lab = scales::percent(anal2_1$Percentage)
ggplot(data = anal2_1)+
  geom_bar(aes("",Percentage,fill= Area_Class),stat = "identity")+
  coord_polar("y",start = 0)+
  theme_void()+
  geom_text(aes(x=1,y=cumsum(Percentage)-Percentage/2,label = lab))+
  labs(title = "What Area Class do Bachelors prefer?",
       fill = "Area Class")
```

Figure. 5. 1

The above code is to analyze which type of area bachelors and families prefer whether is Super Area or Carpet Area using pie chart.

Output:

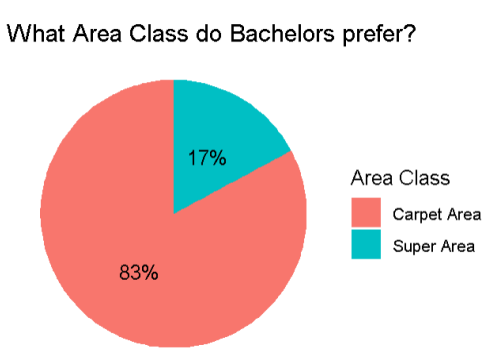

Figure. 5. 2

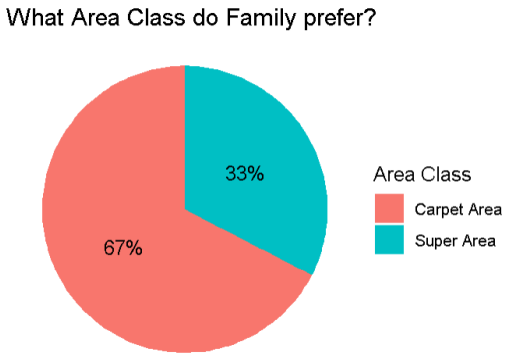

Figure. 5. 3

Findings:

- 83% of bachelors prefer Carpet Area.
- 17% of bachelors prefer Super Area.
- 67% of families prefer Carpet Area.
- 33% of families prefer Super Area.

5.2 Analysis: What City do Bachelors/Families prefer?

```
anal2_2 = RentalDataset %>%
  group_by(City)%>%
  filter(Renter_Preferred %in% c("Bachelors"))%>%
  count()%>%
  ungroup()%>%
  mutate(Percentage = `n`/sum(`n`))%>%
  arrange(desc(City))

lab = scales::percent(anal2_2$Percentage)
ggplot(data = anal2_2)+
  geom_bar(aes("",Percentage,fill= City),stat = "identity")+
  coord_polar("y",start = 0)+
  theme_void()+
  geom_text(aes(x=1,y=cumsum(Percentage)-Percentage/2,label = lab))+
  labs(title = "What City do Bachelors prefer?",
       fill = "City")+
  scale_fill_manual(values=c("#fff5ef", "#c1d0d7", "#a9ba39",
                             "#63985c",  "#3b6f6c", "#499fa5"))
```

Figure. 5. 4

The attached code is to analyze which city is preferred the most by bachelors and families, the function scale_fill_manual() is used to set the color for the pie manually.

Output:

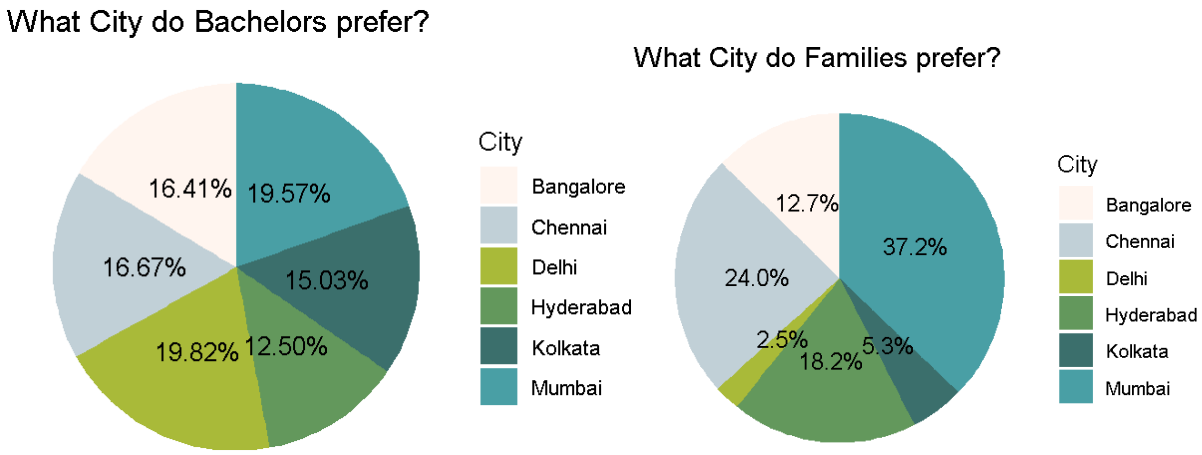

Figure. 5. 5

Figure. 5. 6

Findings:

- The percentage of houses in every city
- Bachelors and Families are most rented in Mumbai
- The least number of houses rented by families are in Delhi
- The number of houses rented by bachelors is approximately equal in every city.

5.3 Analysis: What Contact Types do Bachelors/Families prefer?

```
anal2_3 = RentalDataset %>%
  group_by(Contact_Type)%>%
  filter(Renter_Preferred %in% c("Bachelors"))%>%
  count()%>%
  ungroup()%>%
  mutate(Percentage = `n`/sum(`n`))%>%
  arrange(desc(Contact_Type))
lab = scales::percent(anal2_3$Percentage)
ggplot(data = anal2_3)+
  geom_bar(aes("",Percentage,fill= Contact_Type),stat = "identity")+
  coord_polar("y",start = 0)+
  theme_void()+
  geom_text(aes(x=1,y=cumsum(Percentage)-Percentage/2,label = lab))+
  labs(title = "What Contact Type do Bachelors prefer?",
       fill = "Contact Type")+
  scale_fill_manual(values=c("#fff5ef", "#c1d0d7"))
```

Figure. 5. 7

The above code is to display which contact type bachelors and families prefer, either contact the owner or contact the agent, the function mutate() is used to calculate the percentage and assign it to the label to display it in the pie chart.

Output:

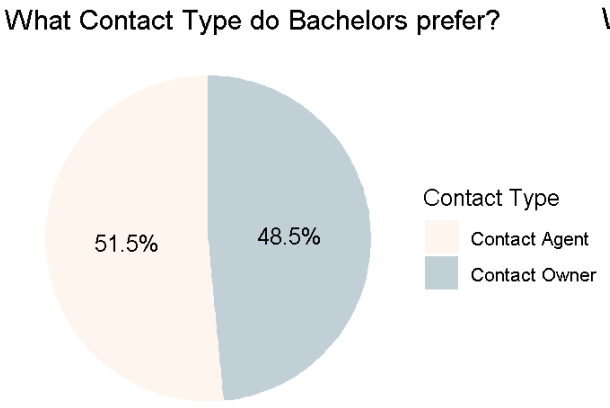

Figure. 5. 8

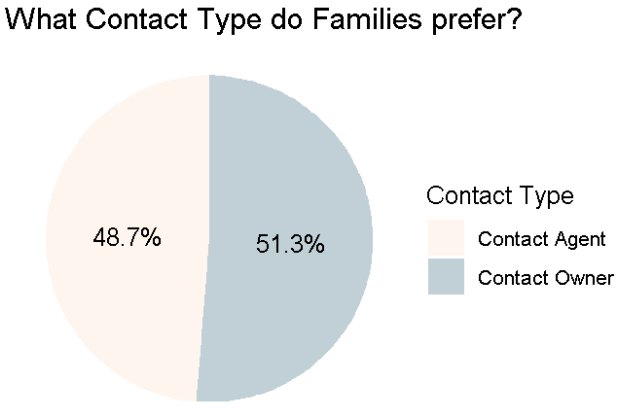

Figure. 5. 9

Findings:

- Bachelors prefer to contact agents more than families.
- Families prefer to contact owners more than Bachelors.

5.4 Analysis: What Furnishing Status do renters prefer if it was Super Area?

```
anal2_4 = RentalDataset %>%
  subset(Area_Class %in% c("Super Area"))%>%
  group_by(Furnishing_Types,Area_Class)%>%
  count( name= "Count" )%>%
  view()%>%
  ungroup()%>%
  mutate(Percentage = `Count`/sum(`Count`))%>%
  arrange(desc(Furnishing_Types))
lab = scales::percent(anal2_4$Percentage)
ggplot(data = anal2_4)+
  geom_bar(aes("",Percentage,fill= Furnishing_Types),stat = "identity")+
  coord_polar("y",start = 0)+
  theme_void()+
  geom_text(aes(x=1,y=cumsum(Percentage)-Percentage/2,label = lab))+
  labs(title = "Super Area ",
       fill = "Furnishing Status")+
  scale_fill_manual(values=c("#fff5ef", "#c1d0d7", "blue"))
```

Figure. 5. 10

The above code is to display what furnishing status renters like it was the super area, the function coord_polar is used to make the shape circle.

Output:

	Furnishing_Types	Area_Class	Count
1	Furnished	Super Area	295
2	Semi-Furnished	Super Area	1112
3	Unfurnished	Super Area	925

Figure. 5. 11

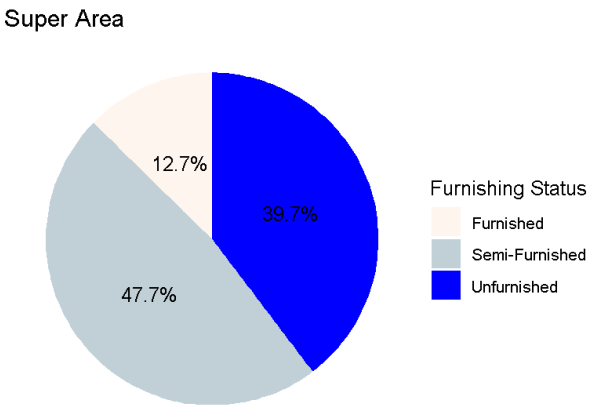

Figure. 5. 12

Findings:

- 12,7% of renters prefer Furnished type if its Super Area.
- 39,7% of renters prefer Ungurnished type if its Super Area.
- 47,7% of renters prefer Semi-Furnished type if its Super Area.

5.5 Analysis: What City tenants prefer if they want a Carpet area?

```
anal2_5 = RentalDataset %>%
  subset(Area_Class %in% c("Carpet Area"))%>%
  group_by(City ,Area_Class)%>%
  count( name= "Count" )%>%
  view()%>%
  ungroup()%>%
  mutate(Percentage = `Count`/sum(`Count`))%>%
  arrange(desc(City))
lab = scales::percent(anal2_5$Percentage)
ggplot(data = anal2_5)+
  geom_bar(aes("",Percentage,fill= City),stat = "identity")+
  theme_void()+
  geom_text(aes(x=1,y=cumsum(Percentage)-Percentage/2,label = lab))+
  labs(title = "Best city for renters for Carpet Area ",
       fill = "City")+
  scale_fill_manual(values=c("#49678D", "#ED30C7", "#4A192C",
                             "#9D9101", "#F3A505", "#FF2301"))
```

Figure. 5. 13

The code above is to analyze what city renters are more likely prefer if it's a carpet area, the function scale() is used to calculate the number to percentage form.

Output:

	City	Area_Class	Count
1	Bangalore	Carpet Area	320
2	Chennai	Carpet Area	304
3	Delhi	Carpet Area	302
4	Hyderabad	Carpet Area	224
5	Kolkata	Carpet Area	257
6	Mumbai	Carpet Area	712

Figure. 5. 14

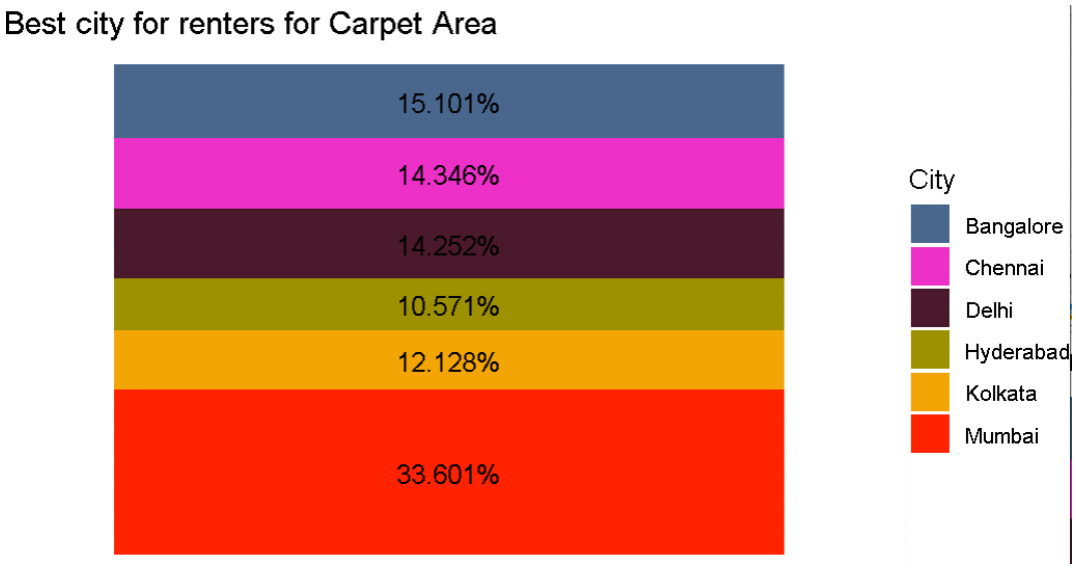

Figure. 5. 15

Findings

- Most renters prefer Mumbai city if it a Carpet Area.
- Kolkata city is the last option to choose if renters are looking for a carpet area.

5.6 Analysis: What type of Point of Contact do renters prefer if the house is furnished

```
anal2_6 = RentalDataset %>%
  subset(Furnishing_Types %in% c("Furnished"))%>%
  group_by(Furnishing_Types ,Contact_Type)%>%
  count( name= "Count" )%>%
  view()%>%
  ungroup()%>%
  mutate(Percentage = `Count`/sum(`Count`))%>%
  arrange(desc(Contact_Type))
lab = scales::percent(anal2_6$Percentage)
ggplot(data = anal2_6)+
  geom_bar(aes("",Percentage,fill= Contact_Type),stat = "identity")+
  theme_void()+
  geom_text(aes(x=1,y=cumsum(Percentage)-Percentage/2,label = lab))+
  labs(title = "Point of Contact if the house is furnished ",
       fill = "Furnished Type")+
  scale_fill_manual(values=c("#2E0182", "#7AAFC3", "#63935A"))
```

Figure. 5. 16

- The code above is to check which point of contact renters prefer if the house is furnished, the function geom_text(0) is used to display the percentage that was gotten from scale() function.

Output:

	Furnishing_Types	Contact_Type	Count
1	Furnished	Contact Agent	263
2	Furnished	Contact Owner	369

Figure. 5. 17

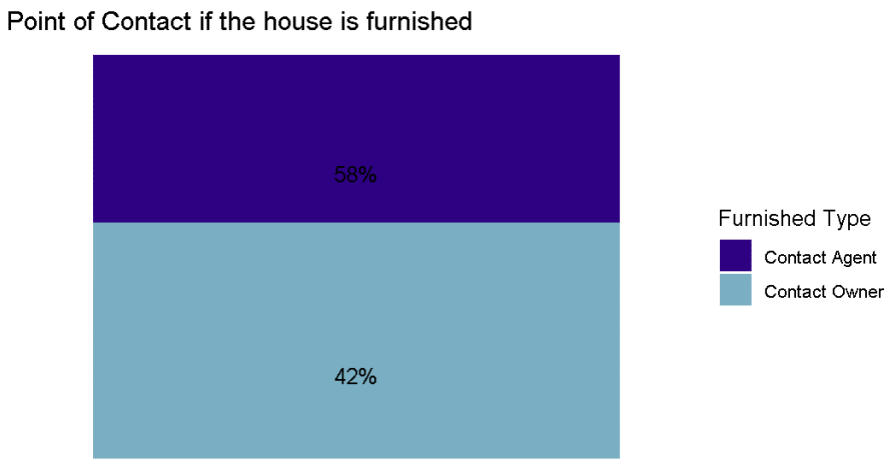

Figure. 5. 18

Findings:

- 58% of renters prefer contacting an agent if the house is furnished.
- 42% of renters prefer contacting an owners if the house is furnished.

Github: aimancsv

5.7 Analysis: What city do renters prefer if the house has 3 BHK and fully furnished

```
anal2_6 = RentalDataset %>%
  subset(Furnishing_Types %in% c("Furnished"))%>%
  subset(BHK == 3)%>%
  group_by(Furnishing_Types ,BHK,City)%>%
  count( name= "Count" )%>%
  view()%>%
  ungroup()%>%
  mutate(Percentage = `Count`/sum(`Count`))%>%
  arrange(desc(City))
lab = scales::percent(anal2_6$Percentage)
ggplot(data = anal2_6)+
  geom_bar(aes("",Percentage,fill= City),stat = "identity")+
  theme_void()+
  geom_text(aes(x=1,y=cumsum(Percentage)-Percentage/2,label = lab))+
  labs(title = "Houses have three BHK and furnished ",
       fill = "City")
```

Figure. 5. 19

The code above is to check if the houses available in all the cities in the database have three BHK and are furnished, a subset(0 functions is used here to filter the data from other unwanted values.

Output:

	Furnishing_Types	BHK	City	Count
1	Furnished	3	Bangalore	21
2	Furnished	3	Chennai	29
3	Furnished	3	Delhi	20
4	Furnished	3	Hyderabad	35
5	Furnished	3	Kolkata	8
6	Furnished	3	Mumbai	59

Figure. 5. 20

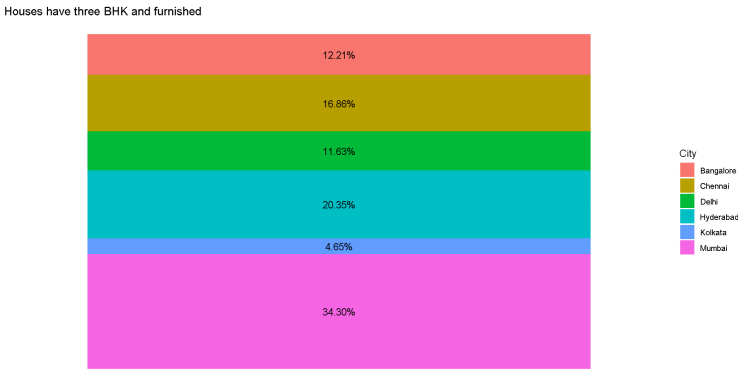

Figure. 5. 21

Fiindings:

- Most of the houses that furnished and have threeBHKs are in Mumbai
- The lease amount of houses that have three BHKs and dully furnished are in Kolkata

5.8 Analysis: What type of renters prefer contacting to owners if the is unfurnished

```
anal2_7 = RentalDataset %>%
  subset(Furnishing_Types %in% c("Unfurnished"))%>%
  subset(Contact_Type %in% c("Contact Owner"))%>%
  group_by(Furnishing_Types ,Contact_Type,Renter_Preferred)%>%
  count( name= "Count" )%>%
  view()%>%
  ungroup()%>%
  mutate(Percentage = `Count`/sum(`Count`))%>%
  arrange(desc(Renter_Preferred))
lab = scales::percent(anal2_7$Percentage)
ggplot(data = anal2_7)+
  geom_bar(aes("",Percentage,fill= Renter_Preferred),stat = "identity")+
  theme_void()+
  coord_polar("y",start = 0)+
  geom_text(aes(x=1,y=cumsum(Percentage)-Percentage/2,label = lab))+
  labs(title = "Unfurnished houses by Owners",
       fill = "Renter Preferred")+
  scale_fill_manual(values=c("#fff5ef", "#c1d0d7", "blue"))
```

Figure. 5. 22

The code above is to display the houses that are rented by owners and its unfirnished based on the rental type, group_by() function is used to get the needed colomn for the analysis.

Output:

	Furnishing_Types	Contact_Type	Renter_Preferred	Count
1	Unfurnished	Contact Owner	Bachelors	235
2	Unfurnished	Contact Owner	Bachelors/Family	973
3	Unfurnished	Contact Owner	Family	74

Figure. 5. 23

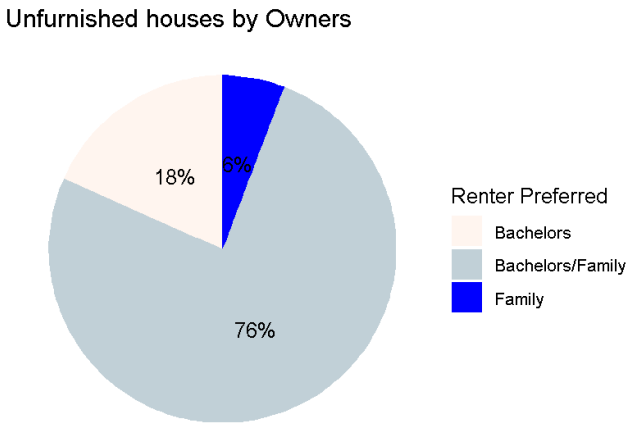

Figure. 5. 24

Findings:

- 6% of families prefer rented unfurnished houses by owners
- 18% of bachelors prefer rented unfurnished houses by owners
- Most of the unfurnished houses by owners are rented from both types

5.9 Analysis: What cities have houses for bachelors with a Super area and 2 BHKs and 1 bathroom

```
anal2_8 = RentalDataset %>%
  subset(Area_Class %in% c("Super Area"))%>%
  subset(BHK == 2)%>%
  subset(Bathroom == 1)%>%
  group_by(Area_Class ,BHK,Bathroom,City)%>%
  count( name= "Count" )%>%
  view()%>%
  ungroup()%>%
  mutate(Percentage = `Count`/sum(`Count`))%>%
  arrange(desc(City))
lab = scales::percent(anal2_8$Percentage)
ggplot(data = anal2_8)+
  geom_bar(aes("",Percentage,fill= City),stat = "identity")+
  theme_void()+
  coord_polar("y",start = 0)+
  geom_text(aes(x=1,y=cumsum(Percentage)-Percentage/2,label = lab))+
  labs(title = "Super Area Houses with 2 BHKs and 1 bathroom",
       fill = "City")+
  scale_fill_manual(values=c("#fff5ef", "#c1d0d7", "#a9ba39",
                             "#63985c",  "#3b6f6c", "#499fa5"))
```

Figure. 5. 25

The code above is to display the houses that have one bathroom and two BHks in the super area by cities, Count() function is used to count the value found based on the conditions above, this code will present a pie-style.

Output:

	Area_Class	BHK	Bathroom	City	Count
1	Super Area	2	1	Bangalore	56
2	Super Area	2	1	Chennai	37
3	Super Area	2	1	Delhi	41
4	Super Area	2	1	Hyderabad	21
5	Super Area	2	1	Kolkata	81
6	Super Area	2	1	Mumbai	2

Figure. 5. 26

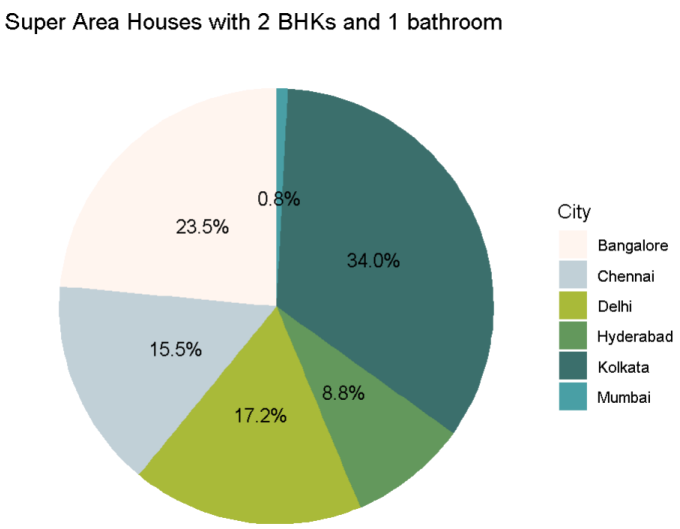

Figure. 5. 27

Findings:

- There are a lot of houses with two BHKs and 1 bathroom in the super area in Kolkata with 34%.
- It's hard to find a house with two BHKs and 1 bathroom in the super area in Mumbai with 0.8%

5.10 Analysis: Scenario for finding a house

Assuming there is a Scenario of someone trying to find a house with specific features:

- The house must have four BHKs
- The house must have two bathrooms
- The house must be in a Super area.
- The point of contact must be through the owner

The question is what cities have these features?

Code:

```
anal2_9 = RentalDataset %>%
  subset(BHK == 4)%>%
  subset(Bathroom == 2)%>%
  subset(Area_Class == c("Super Area"))%>%
  subset(Contact_Type == "Contact Owner")%>%
  group_by(Area_Class ,BHK,Bathroom,City,Contact_Type)%>%
  count( name= "Count" )%>%
  view()%>%
  ungroup()%>%
  mutate(Percentage = `Count`/sum(`Count`))%>%
  arrange(desc(City))
lab = scales::percent(anal2_9$Percentage)
```

Figure. 5. 28

The scenario is achieved by applying the above cove, subset(0) functions are used to filter the values needed in the scenario, with help of the group_by()function.

Output:

	Area_Class	BHK	Bathroom	City	Contact_Type	Count
1	Super Area	4	2	Bangalore	Contact Owner	1
2	Super Area	4	2	Delhi	Contact Owner	2
3	Super Area	4	2	Hyderabad	Contact Owner	3
4	Super Area	4	2	Kolkata	Contact Owner	2

Figure. 5. 29

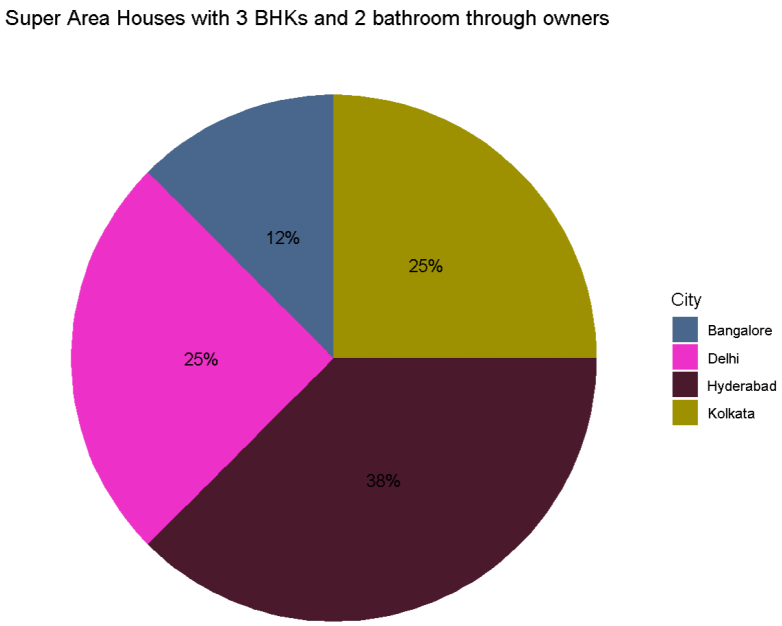

Figure. 5. 30

Findings:

- There are only four cities that have specific features in the scenario.
- Hyderabad is the most city that has houses with the scenario features.
- Bangalore is the least city that has houses with scenario features.

5.11 Analysis (conclusion)

```
# Analysis 2.10:
anal1_10 = RentalDataset %>%
  group_by(Renter_Preferred,BHK,Area_Class,Contact_Type,
           Furnishing_Types)%>%
  summarise(Count = length(Renter_Preferred))%>%
  view()
  ggplot (anal1_10,aes(BHK,Count,size =Area_Class, color =Furnishing_Types))+
  geom_point()+
  facet_wrap (~Renter_Preferred)+
  labs(title = "Type of renter described by BHK, Area Class and Furnishing_Types",
       x = "BHK",
       y = "Count")
```

Figure. 5. 31

The code above is to conclude this question of which it is Bachelors and families prefer, the code has grouped all the columns used in this question and found the count of all of them.

Output:

	Renter_Preferred	BHK	Area_Class	Contact_Type	Bathroom	Furnishing_Types	Count
1	Bachelors	1	Carpet Area	Contact Agent	1	Furnished	7
2	Bachelors	1	Carpet Area	Contact Agent	1	Semi-Furnished	27
3	Bachelors	1	Carpet Area	Contact Agent	1	Unfurnished	27
4	Bachelors	1	Carpet Area	Contact Agent	2	Furnished	2
5	Bachelors	1	Carpet Area	Contact Agent	2	Semi-Furnished	6
6	Bachelors	1	Carpet Area	Contact Agent	2	Unfurnished	13
7	Bachelors	1	Carpet Area	Contact Owner	1	Furnished	6
8	Bachelors	1	Carpet Area	Contact Owner	1	Semi-Furnished	17
9	Bachelors	1	Carpet Area	Contact Owner	1	Unfurnished	35
10	Bachelors	1	Carpet Area	Contact Owner	2	Furnished	1

Figure. 5. 32

conclusion:

According to the question(what Bachelors/Families prefer), based on the analysis above, Bachelors prefer their area class to be Carpet Area, and most of the bachelors prefer Delhi and Mumbai city and Bachelors contact preferred is with agents. On the other hand, families prefer Carpet areas as well and their preferred city is Mumbai, families prefer contacting the owner directly unlike bachelors. There also other factors were checked and given scenarios to analyze deeply what they prefer.

6.0. Question 3: Distribution of how much will the tenant pay according to the size?

6.1 Analysis: Price described by size and multi factors.

```
anal3_1 = RentalDataset %>%
  ggplot (aes(Square_Feet,Price,size =BHK, color =Area_Class ))+
  geom_point()+
  facet_wrap (~Renter_Preferred)+
  labs(title = "Price described by size and multi factors",
       x = "Square Feet",
       y = "Price")
```

Figure 6. 1

The code above is used to display multi factors to get an idea about the data using Scatterplot, scatter can be used by applying geom_point from ggplot2, in this analysis, we will display the relationships between Size, Price, BHK, Area Class, and Renter Preferred.

Output:

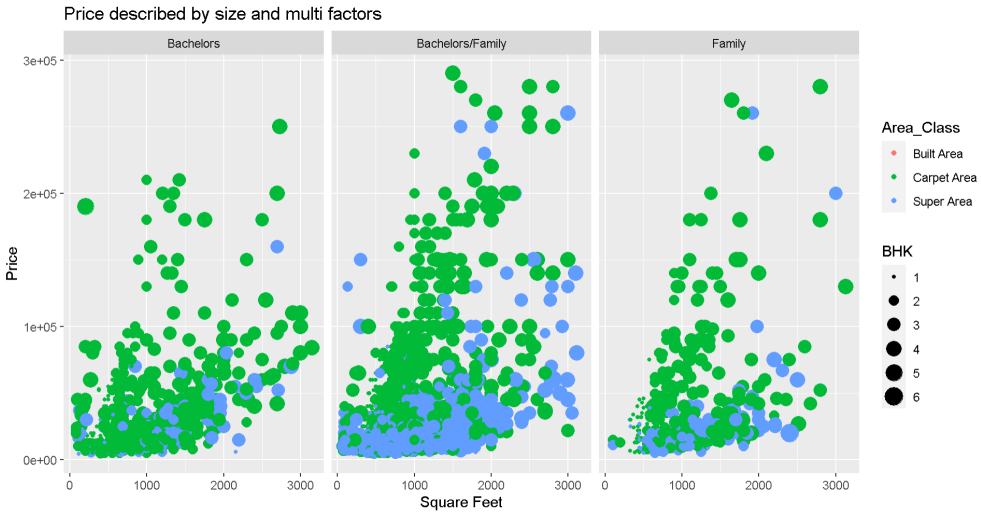

Figure 6. 2

Findings:

- The relationship between Size, Price, Area class, BHK, and Renter Preferred.
- The larger the house the more expensive.
- Carpet areas are more than Super areas.
- Bachelors and families have the most value.
- The more BHK the house has the bigger the house and the more expensive.

6.2 Analysis: The number of houses by size

```
# Analysis 3.2: The number of houses by size
anal3_2<-RentalDataset%>%
   group_by(Square_Feet)%>%
   summarise(Count = length(Square_Feet))%>%
  top_n(10)%>%
  ggplot(aes(Square_Feet, Count))+
  geom_point(size = 3, alpha = 0.8)+
  geom_smooth(color = "red")+
  theme_bw()+
  labs(title = "The number of houses by size",
       x = "Square Feet",
       y = "Count")

anal3_2
```

Figure 6. 3

The code above is to calculate the number of houses rented by size, top_10() function is used to get the top 10 values of the data, the geom_point is used in this analysis.

Output:

	Square_Feet	Count
1	400	131
2	450	138
3	500	184
4	600	222
5	700	182
6	800	217
7	900	164
8	1000	237
9	1100	172
10	1200	191

Figure 6. 4

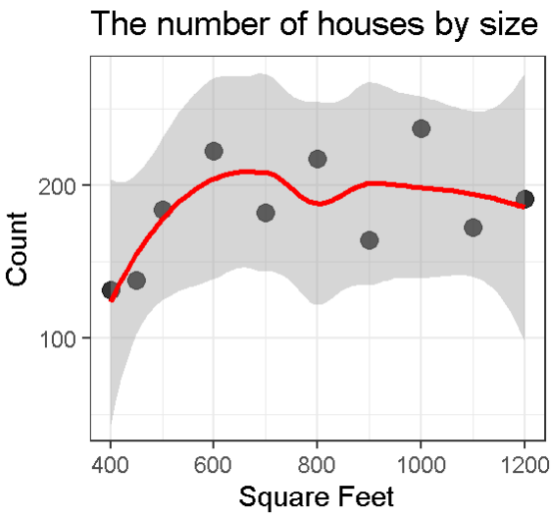

Figure 6. 5

Findings:

- Houses with 1,000 square feet were recorded as the most houses rented by size.
- Houses in the range between 400 and 1200 square feet are the most rented house.
- The red line doesn't specify any relationship between the number of the houses rented and the size.6.3 Analysis: The average size by BHK

6.3 Analysis: The average size by BHK

```
anal3_3<-RentalDataset%>%
  group_by(BHK)%>%
  summarise(Avg_size = mean(Square_Feet))%>%
  ggplot(aes(BHK, Avg_size))+
  geom_point(size = 3, alpha = 0.8,color = "green")+
  geom_smooth(color = "white")+
  theme_dark()+
  labs(title = "The average size by BHK",
       x = "BHK",
       y = "Square Feet")
```

Figure 6. 6

The analysis above is to calculate the average size by BHK, the plot used in this analysis us geom_point from ggplot, and geaom_smooth is used to draw a line for the points

Output:

	BHK	Avg_size
1	1	489.7965
2	2	860.7806
3	3	1412.0642
4	4	1969.8320
5	5	1335.0000
6	6	2000.0000

Figure 6. 7

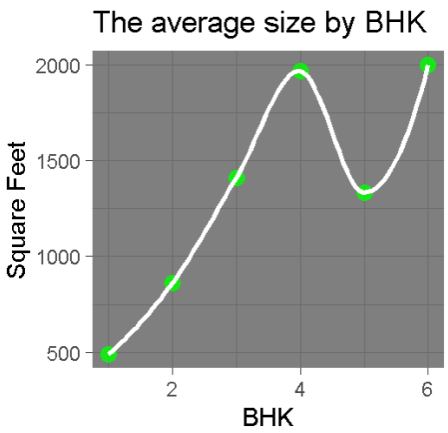

Figure 6. 8

Findings:

- The Houses with one BHK has the lowest average size
- The Houses with five BHK has the largest average size
- Mostly the more BHK the house has the bigger

6.4 Analysis: The average rent by BHK

```
anal3_4<-RentalDataset%>%
  group_by(BHK)%>%
  summarise(Avg_price = mean(Price))
theme_black = function() {
  theme(
    # Specify panel options
    panel.background = element_rect(fill = "black", color  =  NA),
    panel.border = element_rect(fill = NA, color = "white"),
    panel.grid.major = element_line(color = "grey35"),
    panel.grid.minor = element_line(color = "grey20"),
    panel.margin = unit(0.5, "lines"))
}
  ggplot(anal3_4,aes(BHK, Avg_price))+
  geom_point(size = 3, alpha = 0.8,color = "white")+
  geom_smooth(method = "lm", se = FALSE, color = "white")+
  theme_black()+
  labs(title = "The average Price by BHK",
       x = "BHK",
       y = "Price")
```

Figure 6. 9

The above code is to calculate the average price by the number of BHK, a function in the analysis which is theme_black, to apply a black theme in the ggplot. lm method is applied to get a straight line to display if the price is increasing or decreasing by the number of BHK.

Output:

	BHK	Avg_price
1	1	14436.35
2	2	21457.25
3	3	46821.15
4	4	106443.21
5	5	70625.00
6	6	40000.00

Figure 6. 10

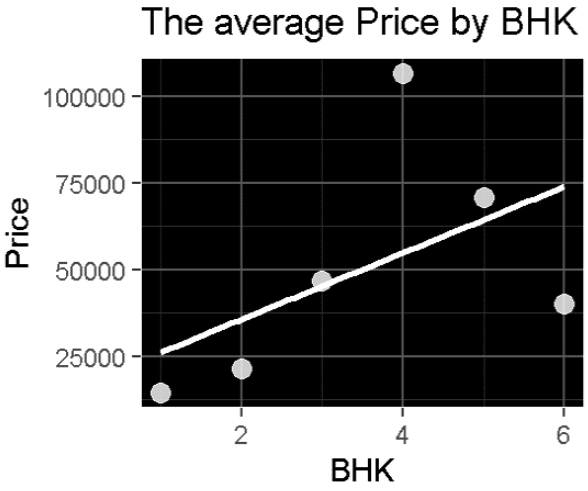

Figure 6. 11

Findings:

- The houses with one BHK have the cheapest average price
- The houses with four BHK have the most expensive average price
- The line shows mostly the more BHK the house has the most expincive

6.5 Analysis: The number of houses by BHK

```
anal3_5<-RentalDataset%>%
  group_by(BHK)%>%
  summarise(Count = length(BHK))%>%
  top_n(10)
  ggplot(anal3_5, aes(BHK, Count))+
  geom_point(size = 3, alpha = 0.8)+
  geom_smooth(method = 'glm')+
  theme_bw()+
  labs(title = "Number of houses by BHK",
       x = "BHK",
       y = "Count")
anal3_5
```

Figure 6. 12

The analysis above is to check how many houses there are by the number of BHK, the method glm is used to display a shadow to show the points that lie a bit away from the other data.

Output:

	BHK	Count
1	1	1086
2	2	2206
3	3	1028
4	4	125
5	5	4
6	6	4

Figure 6. 13

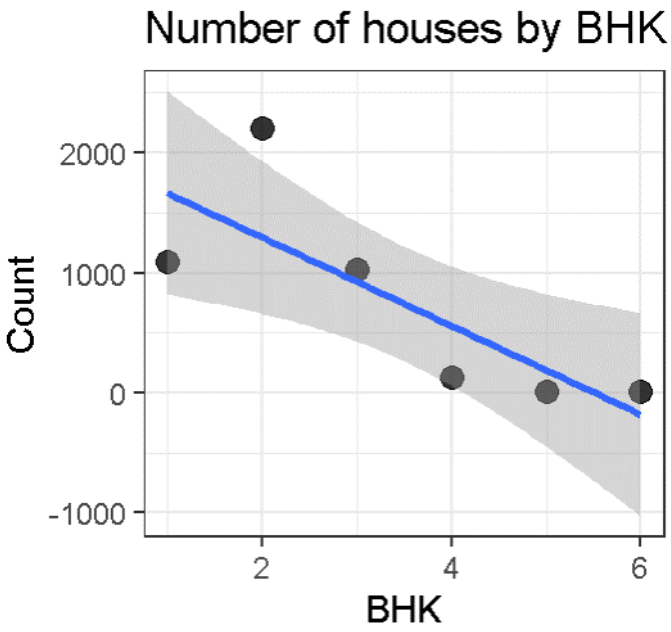

Figure 6. 14

Findings:

- There are only 4 houses with five and six BHK
- Houses with two of BHKs are recorded as the most rented house
- The less BHK the house has the more renters

6.6 Analysis The average size by Area Type

```
anal3_6<-RentalDataset%>%
  group_by(Area_Class)%>%
  summarise(Avg_size = mean(Square_Feet))%>%
  view()
```

Figure 6. 15

The code above is to calculate the average by area type using mean() function.

Output:

	Area_Class	Avg_size
1	Built Area	750.0000
2	Carpet Area	943.3870
3	Super Area	918.2826

Figure 6. 16

Findings:

The average size for Carpet Area is 943 Square feet.

The average size for Built Area is 750 Square feet.

The average size for Super Area is 918 Square feet.

6.7 Analysis The average price by Area Type

```
anal3_7<-RentalDataset%>%
  group_by(Area_Class)%>%
  summarise(Avg_Price = mean(Price))%>%
  view()
```

Figure 6. 17

The above code is to find the average price for the area type, summarise function is used to filter the data according to the mean of the price.

Output:

	Area_Class	Avg_Price
1	Built Area	10500.00
2	Carpet Area	39184.23
3	Super Area	17941.78

Findings:

- The average price for Carpet Area is 39.184 Per year.
- The average price for Built Area is 39.184 Per year.
- The average price for Super Area is 17.941 Per year.

6.8 Analysis The average price of every city.

```
anal3_8<-RentalDataset%>%
  group_by(City)%>%
  summarise(Avg_Price = mean(Price))%>%
  view()
ggplot(anal3_8, aes(x=City, y=Avg_Price)) +
  geom_bar(stat = "identity") +
  coord_flip()+
  labs(title = "Average size in every city",
       x = "City",
       y = "Avg_Price")
```

Figure 6. 18

The code above is to find the average in every city, and geom_bar is used to display the data.

Output:

	City	Avg_Price
1	Bangalore	19134.34
2	Chennai	19617.96
3	Delhi	28148.99
4	Hyderabad	17943.17
5	Kolkata	11898.73
6	Mumbai	64373.38

Figure 6. 19

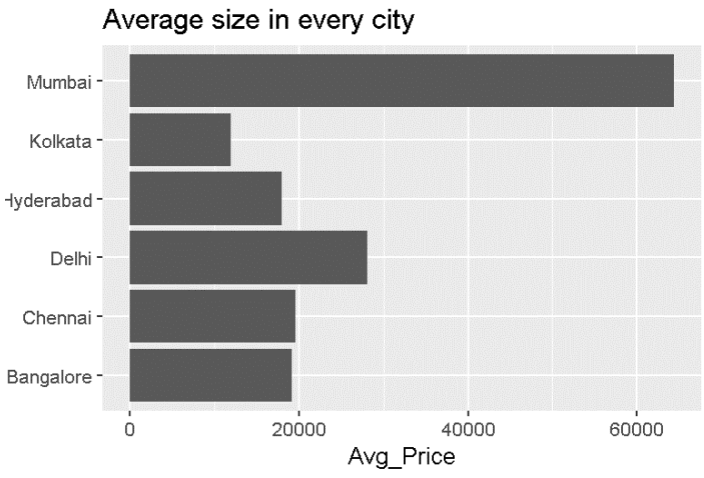

Figure 6. 20

Findings:

- The average price in Mumbai was the highest with 64.000
- The average price in Kolkata is the lowest with 11.000

6.9 Analysis: Average Price of every city by area and Furnished type.

```
# Analysis 3.9:  verage Price of every city by area and Furnished type .
anal3_9<-RentalDataset%>%
  group_by(City, Area_Class,Furnishing_Types)%>%
  summarise(Avg_Price = mean(Price))%>%
  view()
ggplot(anal3_9, aes(x=Furnishing_Types, y=Avg_Price,  fill = City)) +
  geom_bar(stat = "identity") +
  facet_wrap (~Area_Class)+
  labs(title = "Average Price of every city by area and Furnished type",
       x = "Furnishing Types",
       y = "Avg_Price")
```

Figure 6. 21

The code above is to check the average price of every city by area and furnished type. The code fecet_wrap() is used to add new column data to the analysis.

Output:

	City	Area_Class	Furnishing_Types	Avg_Price
1	Bangalore	Carpet Area	Furnished	44654.29
2	Bangalore	Carpet Area	Semi-Furnished	22644.39
3	Bangalore	Carpet Area	Unfurnished	27095.16
4	Bangalore	Super Area	Furnished	18632.08
5	Bangalore	Super Area	Semi-Furnished	16234.10
6	Bangalore	Super Area	Unfurnished	11166.43
7	Chennai	Built Area	Furnished	15000.00
8	Chennai	Carpet Area	Furnished	32321.36
9	Chennai	Carpet Area	Semi-Furnished	26198.09
10	Chennai	Carpet Area	Unfurnished	20031.03

Figure 6. 21

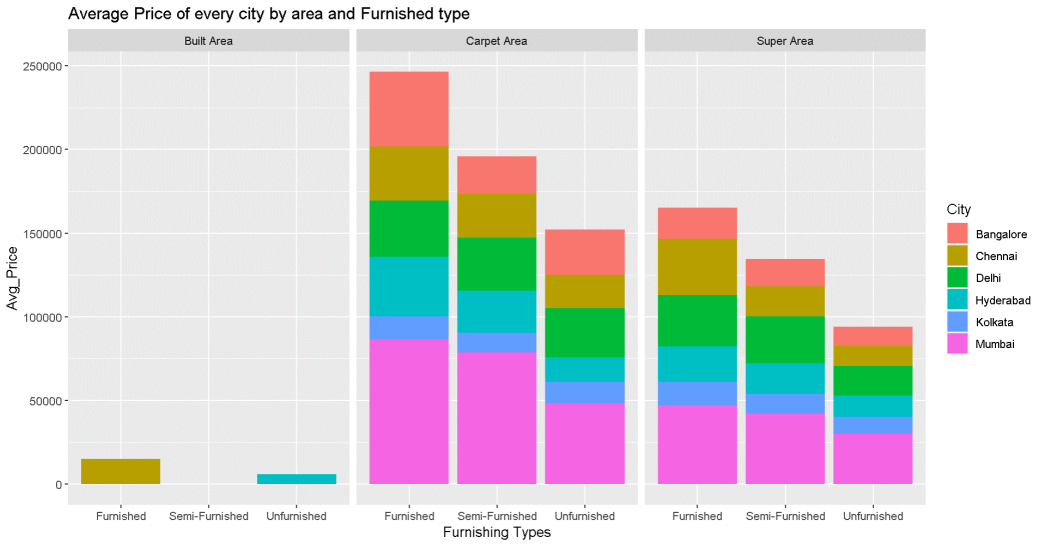

Figure 6. 22

Findings:

- Furnished type in the carpet area has the most average price.
- Unfurnished type in the super area has the least average price

6.10 Analysis (conclusion)

	BHK	Avg_size	Avg_price	Count
1	1	489.7965	14436.35	1086
2	2	860.7806	21457.25	2206
3	3	1412.0642	46821.15	1028
4	4	1969.8320	106443.21	125
5	5	1335.0000	70625.00	4
6	6	2000.0000	40000.00	4

Figure 6. 23

From all the analyses above, the average size, average price, and the number of houses based on size and price are recorded.

```
anal3_6 = RentalDataset
anal3_6 = merge(x = anal3_3, y = anal3_4,  by = "BHK")
anal3_6 = merge(x = anal3_6,y = anal3_5,  by = "BHK")
anal3_6$BHK<-as.character(anal3_6$BHK)
  ggplot (anal3_6,aes(Avg_size,Avg_price,size =Count,color = BHK ))+
  geom_point()+
  theme_bw()+
  geom_smooth(size = 7)+
  labs(title = "BHK described by the price and size",
       x = "Avarage Size",
       y = "Avarage Price")
```

Figure 6. 24

The code above is to describe the BHK by the price and size, starting with making a new table that has the column needed, the function merge() is used to merge the data from different tables,

Geom_point() is used in this final conclusion.

Output:

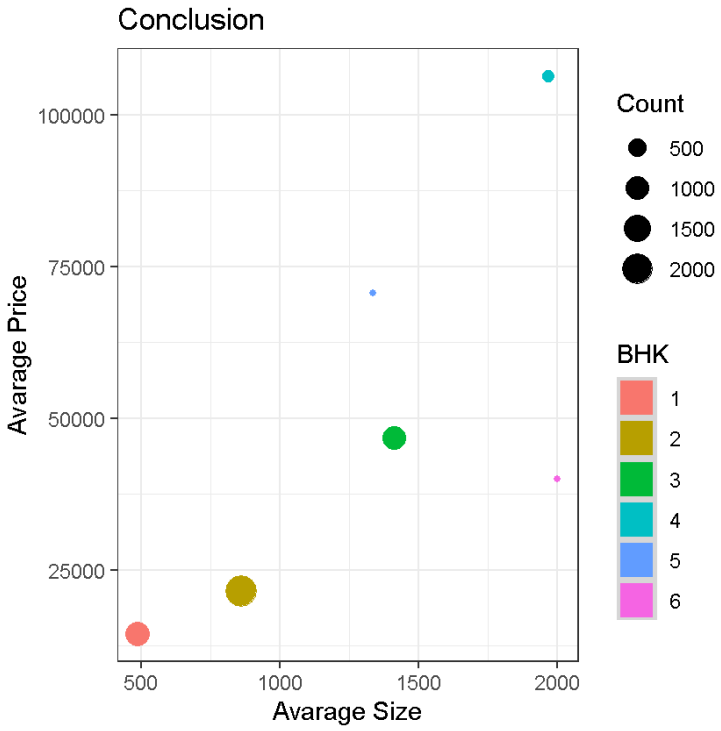

Figure 6. 25

Findings (Conclusion):

- For houses of one BHK, the average house is the least which is 490 square feet, houses have the smallest size, and the tenant will pay approximately 14,000 per year which is the cheapest compared to other houses with more BHKs, 1086 Houses have one BHK.
- For houses of two BHKs, the average house will be 860 square feet and the tenant will pay approximately 21,00 per year, There are 2206 Houses that have two BHKs. Houses with two BHKs have the most renters.
- For houses of three BHKs, the average house is 1,400 square feet and the tenant will pay approximately 47,000 per year, 1028 Houses have three BHKs.
- For houses of four BHKs, the average house is 1,970 square feet and the tenant will pay approximately 106,000 per year, 125 Houses have four BHKs.
- For houses of five BHKs, the average house is 1,300 square feet and the tenant will pay approximately 71,000 per year, 4 Houses have five BHKs.
- For houses of six BHKs, the average house is 2,000 square feet, houses with six BHKs have to biggest size, the tenant will pay approximately 40,000 per year, 4 Houses have five BHKs.

7.0 Question 4: Distribution of which city do tenants prefer to live in and why

7.1 Analysis: Number of houses by the city

```
anal4_1 = RentalDataset
anal4_1<-RentalDataset%>%
  group_by(City)%>%
  summarise(Count = length(City))
ggplot(anal4_1,aes(x=City, y=Count, colour = City)) +
  geom_segment( aes(x=City, xend=City, y=1, yend=Count)) +
  geom_point(  size=4, alpha=0.6) +
  stat_summary(fun = mean, geom = "point", size = 5)+
  geom_hline(aes (yintercept = mean (Count)),colour = "gray70",
             size = 0.9)+
  geom_segment(aes(x = City, xend = City,
                   y = 1, yend = Count), size = 2)+
  theme_light() +
  coord_flip() +
  theme (legend.position = "none") +
  theme(
    panel.grid.major.y = element_blank(),
    panel.border = element_blank(),
    axis.ticks.y = element_blank()
  )+
  labs (title = "Number of houses by the city")
```

Figure 7. 1

The analysis above is to count how many houses in every city, the plot used in this analysis is Lollipop plot from ggplot2, a gray line is applied to find the mean of the houses number

Output:

	City	Count
1	Bangalore	862
2	Chennai	866
3	Delhi	547
4	Hyderabad	822
5	Kolkata	496
6	Mumbai	860

Figure 7. 2

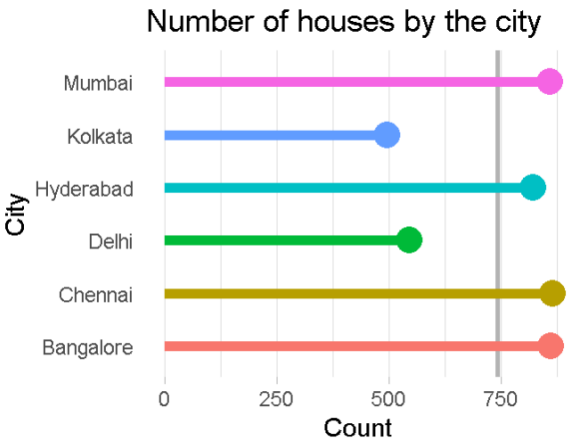

Figure 7. 3

Findings:

- The most number of houses are found in Chennai
- The least number of houses are found in Delhi
- The mean of the houses by cities is 750.

7.2 Analysis: The most rented location in every city

```
anal4_2 = RentalDataset
anal4_2<-RentalDataset%>%
  group_by(City, Location)%>%
  summarise(Count = length(City))%>%
  top_n(1)
ggplot(anal4_2,aes(x=Location, y=Count, colour = City)) +
  geom_segment( aes(x=Location, xend=Location, y=1, yend=Count)) +
  geom_point(  size=4, alpha=0.6) +
  stat_summary(fun = mean, geom = "point", size = 5)+
  geom_hline(aes (yintercept = mean (Count)),colour = "gray70",
             size = 0.9)+
  geom_segment(aes(x = Location, xend = Location,
                   y = 1, yend = Count), size = 2)+
  theme_light() +
  coord_flip() +
  theme(
    panel.grid.major.y = element_blank(),
    panel.border = element_blank(),
    axis.ticks.y = element_blank()
  )+
  labs (title = "The most rented location in every city")
```

Figure 7. 4

The code above to check which location has rented most of the houses by city, top() function is used to get the most rented location in every city.

Output:

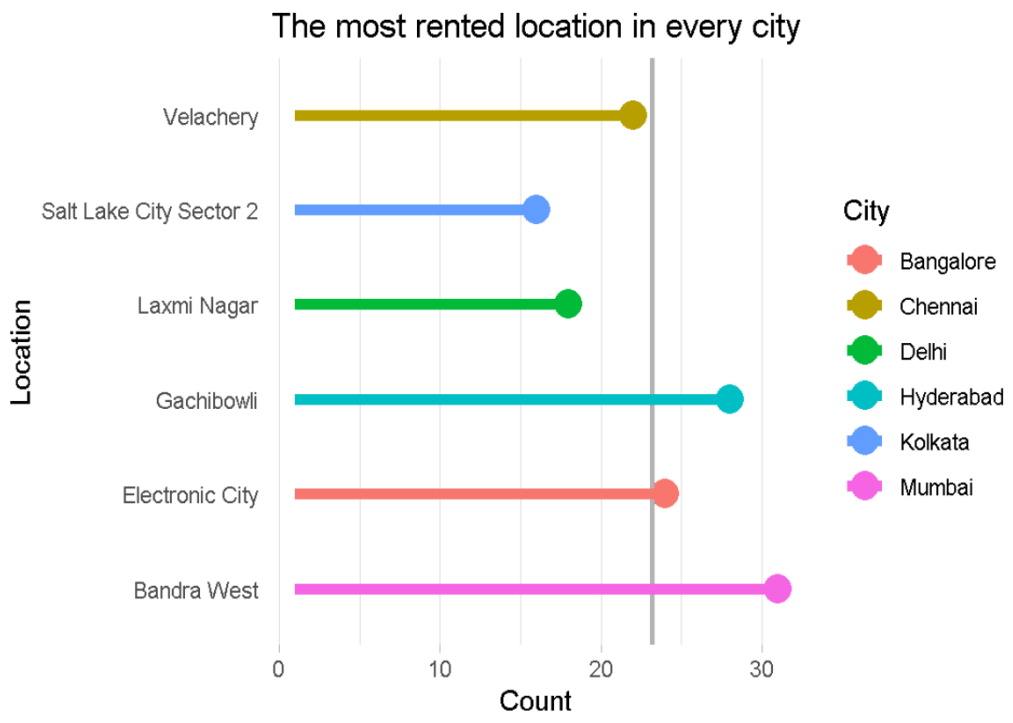

Figure 7. 5

City	Location	Count
Bangalore	Electronic City	24
Chennai	Velachery	22
Delhi	Laxmi Nagar	18
Hyderabad	Gachibowli	28
Kolkata	Salt Lake City Sector 2	16
Mumbai	Bandra West	31

Figure 7. 6

Findings:

- Bandra West location in Mumbai is the most rented location with 31 houses.
- Velachery is the most rented location in Chennai with 22 houses.
- Laxmi Nagar is the most rented location in Delhi with 18 houses.
- Gachibowli is the most rented location in Hyderabad with 28 houses.
- Salt Lake City Sector 2 is the most rented location in Kolkata with 16 houses.

7.3 Analysis: The average Rent by city

```
anal4_3<-RentalDataset%>%
  group_by(City)%>%
  summarise(Avg_price = mean(Price))
ggplot(anal4_3,aes(x=City, y=Avg_price )) +
  geom_hline(aes (yintercept = mean (Avg_price)),colour = "gray70",
             size = 0.9)+
  geom_segment(aes(x = City, xend = City,
                   y = 1, yend = Avg_price), size = 1,
               colour = "sky blue")+
  geom_point(color="blue",size=4, alpha=0.6) +
  theme_light() +
  theme (legend.position = "none") +
  theme(
    panel.grid.major.y = element_blank(),
    panel.border = element_blank(),
    axis.ticks.y = element_blank()
  )+
  labs (title = "The average Rent by city", y = "Average Price")
```

Figure 7. 7

The code above is to calculate the average size by city, and function mean() is used to get the mean of the square feet of all cities

Output:

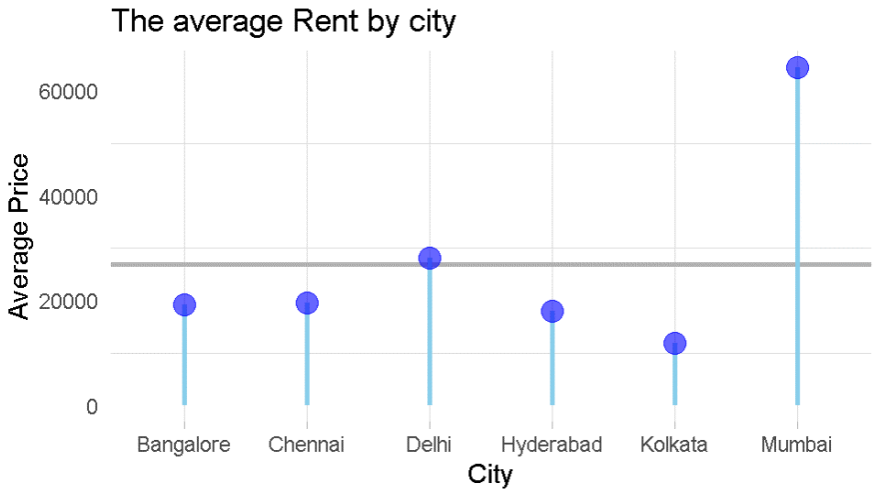

Figure 7. 8

Findings:

- The most expensive city for renting a house is Mumbai with an average of 64,000 per year.
- The cheapest city for renting a house is Kolkata with an average of 12,00 per year.
- The average of renting houses in all cities is 27,000 per year.

7.4 Analysis: The Average Rent by the city ina most rented locations

```
anal4_4 = RentalDataset
anal4_4<-RentalDataset%>%
  group_by(City, Location)%>%
  summarise(Avg_price = mean(Price),Count = length(City))%>%
  top_n(1)%>%
  subset(select = c("City","Location","Avg_price" ))

ggplot(anal4_4,aes(x=Location, y=Avg_price, colour = City)) +
  geom_segment( aes(x=Location, xend=Location, y=1, yend=Avg_price)) +
  geom_point(  size=4, alpha=0.6) +
  stat_summary(fun = mean, geom = "point", size = 5)+
  geom_hline(aes (yintercept = mean (Avg_price)),colour = "gray70",
             size = 0.9)+
  geom_segment(aes(x = Location, xend = Location,
                   y = 1, yend = Avg_price), size = 2)+
  theme_light() +
  coord_flip() +
  theme_dark()+
  theme(
    panel.grid.major.y = element_blank(),
    panel.border = element_blank(),
    axis.ticks.y = element_blank())+
  labs (title = "Average Rent by city in most rented location ",
        y = "Average Rent")
view(anal4_4)
```

Figure 7. 9

The analysis above is to get the average rent in the most rented location based on analysis 4.3,

Output:

	City	Location	Avg_price
1	Bangalore	Electronic City	13507.54
2	Chennai	Velachery	16772.73
3	Delhi	Laxmi Nagar	11072.22
4	Hyderabad	Gachibowli	43553.57
5	Kolkata	Salt Lake City Sector 2	23187.50
6	Mumbai	Bandra West	137096.77

Figure 7. 10

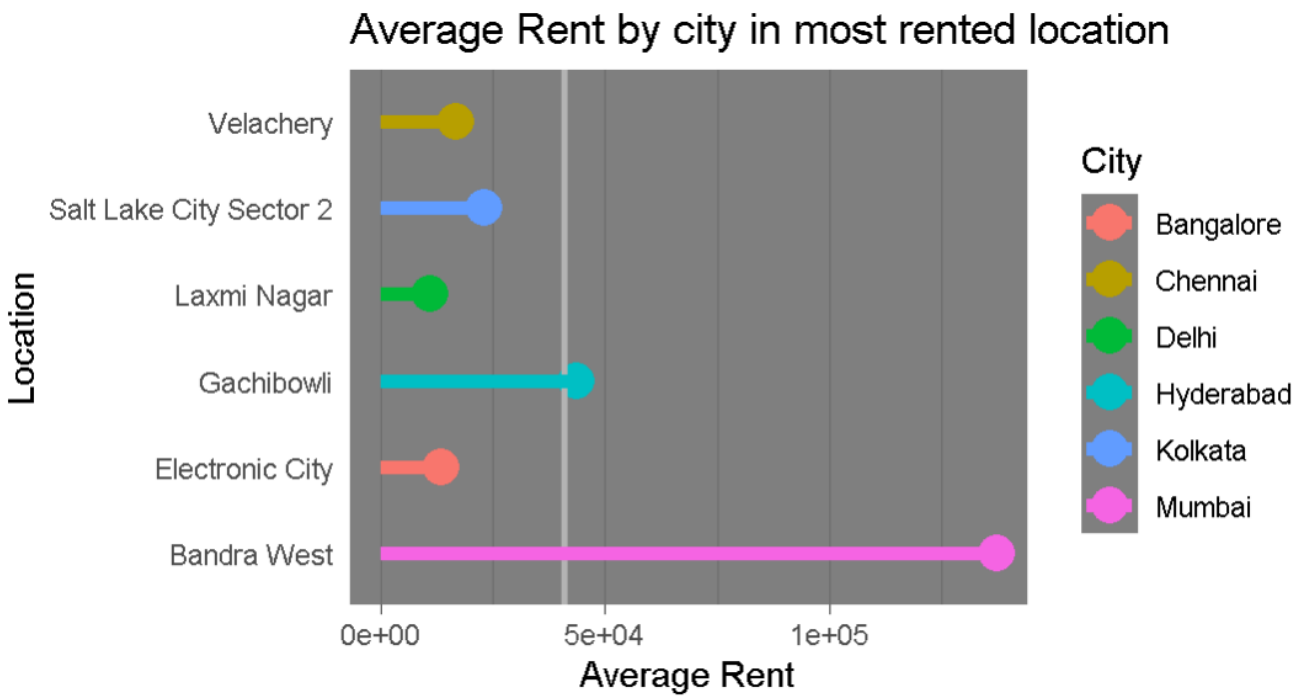

Figure 7. 11

Findings:

- Bandra West location in Mumbai has to most expensive rent.
- Laxmi Nagar location in Delhi has the cheapest average rent price
- The average of renting houses in the most rented locations is 41,000 per year.

7.5 Analysis: The contact type preferred by the city

```
anal4_5 = RentalDataset
anal4_5<-RentalDataset%>%
  group_by(City, Contact_Type)%>%
  subset( Contact_Type != "Contact Builder")%>%
  summarise(Count = length(City))
ggplot(anal4_5,aes(x=Contact_Type, y=Count, colour = Contact_Type)) +
  geom_segment( aes(x=Contact_Type, xend=Contact_Type, y=1, yend=Count)) +
  geom_point(  size=3, alpha=0.6) +
  stat_summary(fun = mean, geom = "point", size = 4)+
  theme_light() +
  facet_wrap(~City)+
  theme(
    panel.grid.major.y = element_blank(),
    panel.border = element_blank(),
    axis.ticks.y = element_blank()
  )+
  labs (title = "The contact type preferred by city",
        x = "Contact Type")
```

Figure 7. 12

The analysis above is to display what kind of contact renters prefer whether is contact with an Agent or Owner, the function subset() is used to remove a random value found in the dataset.

Output:

	City	Contact_Type	Count
1	Bangalore	Contact Agent	168
2	Bangalore	Contact Owner	694
3	Chennai	Contact Agent	126
4	Chennai	Contact Owner	740
5	Delhi	Contact Agent	235
6	Delhi	Contact Owner	312
7	Hyderabad	Contact Agent	95
8	Hyderabad	Contact Owner	726
9	Kolkata	Contact Agent	80
10	Kolkata	Contact Owner	416
11	Mumbai	Contact Agent	671
12	Mumbai	Contact Owner	189

Figure 7. 13

Figure 7. 14

Findings:

- Renters in Bangalore city prefer contacting Owners.
- Renters in Chennai city prefer contacting Owners.
- Renters in Delhi city prefer contacting Owners.
- Renters in Hyderabad city prefer contacting Owners.
- Renters in Kolkata city prefer contacting Owners.
- Renters in Mumbai is the only city that prefers contacting agents.

7.6 Analysis: The contact type preferred by the city in most rented locations

```
anal4_6 = RentalDataset
anal4_6<-RentalDataset%>%
  filter(Location %in% c('Velachery','Laxmi Nagar',
                         'Gachibowli', 'Salt Lake City Sector 2'))%>%
  group_by(Contact_Type,Location, City)%>%
  subset( Contact_Type != "Contact Builder")%>%
  summarise(Count = length(City))
ggplot(anal4_6,aes(x=City, y=Count, colour = Location,fill = )) +
  geom_segment( aes(x=City, xend=City, y=1, yend=Count)) +
  geom_point(  size=3, alpha=0.6) +
  stat_summary(fun = mean, geom = "point", size = 4)+
  theme_light() +
  facet_wrap(~Contact_Type)+
  theme(
    panel.grid.major.y = element_blank(),
    panel.border = element_blank(),
    axis.ticks.y = element_blank())+
  labs (title = "Contact type in most rented locations ",
        x = "Cities")
```

Figure 7. 15

The analysis above is to display what type of contact renters prefer in most rented locations, the function filter() is used to filter the most rented locations that have been found in the previous analysis.

Output:

	Contact_Type	Location	City	Count
1	Contact Agent	Gachibowli	Hyderabad	16
2	Contact Agent	Laxmi Nagar	Delhi	13
3	Contact Agent	Salt Lake City Sector 2	Kolkata	12
4	Contact Agent	Velachery	Chennai	3
5	Contact Owner	Gachibowli	Hyderabad	12
6	Contact Owner	Laxmi Nagar	Delhi	5
7	Contact Owner	Salt Lake City Sector 2	Kolkata	4
8	Contact Owner	Velachery	Chennai	19

Figure 7. 16

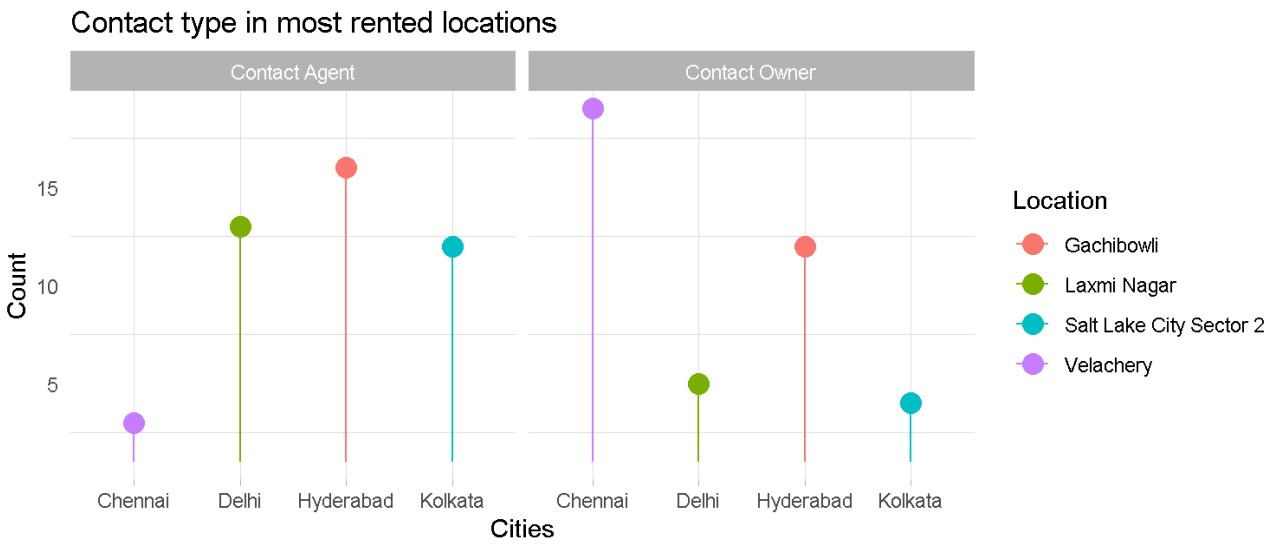

Figure 7. 17

Findings:

- Velachery location from Chennai city has the most rented house by Owner
- Gachibowli location from Hyderabad city has the most rented house by Agent
- Renters are most likely prefer contact to Agents in most rented locations

7.7 Analysis: The Area Type preferred by tenants in most rented locations

```
anal4_7 %>%
  filter(Location %in% c('Velachery','Laxmi Nagar',
                         'Gachibowli', 'Salt Lake City Sector 2'))%>%
  group_by(Area_Class,Location , Renter_Preferred)%>%
  subset(Contact_Type != "Contact Builder")%>%
  summarise(Count = length(Location))%>%
  view()%>%
  ggplot(aes(x=Area_Class, y=Count,
             color = Renter_Preferred)) +
  geom_segment( aes(x=Area_Class, xend=Area_Class, y=1, yend=Count)) +
  geom_point( size=2, alpha=0.6) +
  theme_light() +
  facet_wrap(~Location)+
  theme(
    panel.grid.major.y = element_blank(),
    panel.border = element_blank(),
    axis.ticks.y = element_blank()
  )+
  coord_flip()+
    labs (title = "Area Type preferred by tenants in most rented locations ",
          x = "Area Class",
          y = "Count")
```

Figure 7. 18

The above code is to analyze the type area for most rented locations based on the bachelor and family. geom_sugment() is used to display a line for the area types. Coord_flip() is used to flip the data to display another style.

Output:

	Area_Class	Location	Renter_Preferred	Count
1	Carpet Area	Gachibowli	Bachelors	3
2	Carpet Area	Gachibowli	Bachelors/Family	2
3	Carpet Area	Gachibowli	Family	7
4	Carpet Area	Laxmi Nagar	Bachelors/Family	15
5	Carpet Area	Salt Lake City Sector 2	Bachelors	11
6	Carpet Area	Salt Lake City Sector 2	Bachelors/Family	3
7	Carpet Area	Velachery	Bachelors	3
8	Carpet Area	Velachery	Family	1

Figure 7. 19

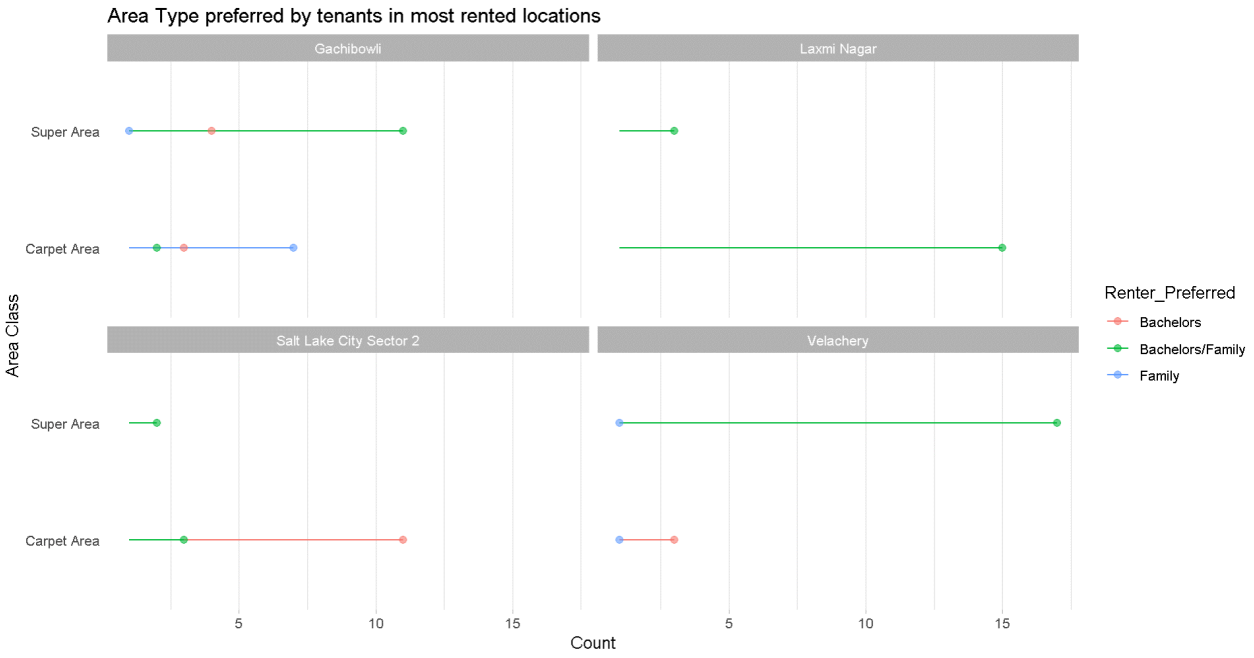

Figure 7. 20

Findings:

- Most of the Renters in the Laxima Nagar location prefer the Carpet area.
- Most of the Renters in the Gachibowli location prefer the Super area.
- Most of the Renters in the Velacherylocation prefer the Super area..
- Most of the Renters in the Salt Lake City Sector 2 location prefer the Carpet area.

7.8 Analysis: When renters usually rent in the most rented location bases on the furnishing type.

```
anal4_8 %>%
  filter(Location %in% c('Velachery','Laxmi Nagar',
                         'Gachibowli', 'Salt Lake City Sector 2'))%>%
  group_by(Published_date = format(Published_date, "%B"),Furnishing_Types, Location)%>%
  summarise(Count = length(Location))%>%
  view()%>%
  ggplot(aes(x=Location, y=Count,
             color = Furnishing_Types)) +
  geom_segment( aes(x=Location, xend=Location, y=1, yend=Count)) +
  geom_point( size=2, alpha=0.6) +
  theme_light() +
  facet_wrap(~factor(Published_date, levels = c("April","May", "June","July")))+
  theme(
    panel.grid.major.y = element_blank(),
    panel.border = element_blank(),
    axis.ticks.y = element_blank()
  )+
  coord_flip()+
  labs (title = "Whaen renters rent in most rented location based on furnishing type ",
        x = "Location",
        y = "Count")
```

Figure 7. 21

The code above is to display the months and when renters usually rent houses based on furnishing type, the function filter() is used to filter the ost rented location.

Output:

	Published_date	Furnishing_Types	Location	Count
1	April	Semi-Furnished	Laxmi Nagar	1
2	April	Semi-Furnished	Salt Lake City Sector 2	1
3	July	Furnished	Gachibowli	3
4	July	Furnished	Velachery	1
5	July	Semi-Furnished	Gachibowli	5
6	July	Semi-Furnished	Laxmi Nagar	6
7	July	Semi-Furnished	Velachery	3
8	July	Unfurnished	Gachibowli	4

Figure 7. 22

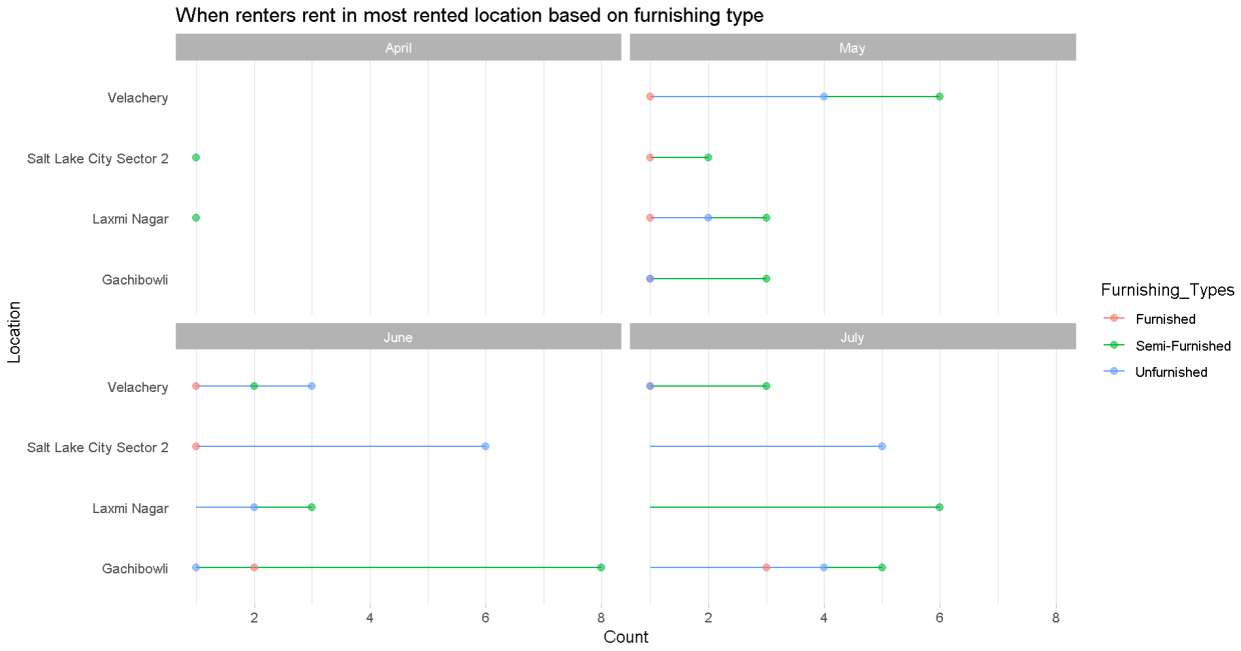

Figure 7. 23

Findings:

- Most renters in May rented semi-furnished houses from Velachery location.
- Most renters in June rented semi-furnished houses from Gachibowli location.
- Most renters in July rented semi-furnished houses from Laxmi Nagar location.

7.9 Analysis: What is the average of BHKs in most rented locations?

```
anal4_9<-RentalDataset
anal4_9 %>%
  filter(Location %in% c('Velachery','Laxmi Nagar',
                         'Gachibowli', 'Salt Lake City Sector 2'))%>%
  group_by(BHK, Location)%>%
  summarise(Avg_BHK = mean(BHK))%>%
  view()%>%
  ggplot(aes(x=BHK, y=Avg_BHK)) +
  geom_segment( aes(x=BHK, xend=BHK, y=1, yend=Avg_BHK)) +
  geom_point( size=2, alpha=0.6, color = "white") +
  theme_light() +
  theme(
    panel.grid.major.y = element_blank(),
    panel.border = element_blank(),
    axis.ticks.y = element_blank()
  )+
  facet_wrap(~Location)+
  theme_dark()+
  labs (title = "Average of BHKs in most rented locations ",
        x = "BHK",
        y = "Count")
```

Figure 7. 24

The above analysis is to display the average of BHKs in the most rented location, the function facer_wrap() is used to wrap the data by the most rented location.

Output:

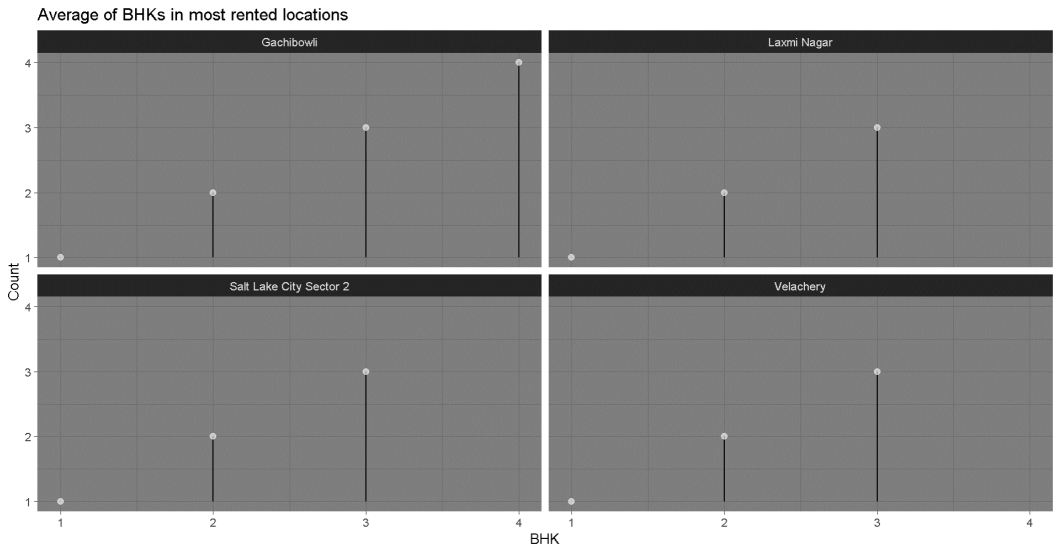

Figure 7. 25

Findings:

- Most of the houses in Gachibowli location have four BHKs.
- Most of the houses in Laxmi Nagar location have three BHKs.
- Most of the houses in Salt Lake City Sector 2 location have three BHKs.
- Most of the houses in Velachery location have three BHKs.

Github: aimancsv

7.10 Analysis (conclusion)

```
anal4_10 %>%
  filter(Location %in% c('Velachery','Laxmi Nagar',
                         'Gachibowli', 'Salt Lake City Sector 2'))%>%
  ggplot(aes(x=Square_Feet, y=Price,color = Location)) +
  geom_point(size = 4) +
  theme_light() +
  facet_wrap(~City)+
  theme(
    panel.grid.major.y = element_blank(),
    panel.border = element_blank(),
    axis.ticks.y = element_blank()
  )+
  labs (title = "City displaied by most rented location, size and price ",
        x = "Square Feet",
        y = "Price")
```

Figure 7. 26

This code describes the city by the most rented location and other factors such as size and price, Lollipop is used here to display the data through out ponts.

Output:

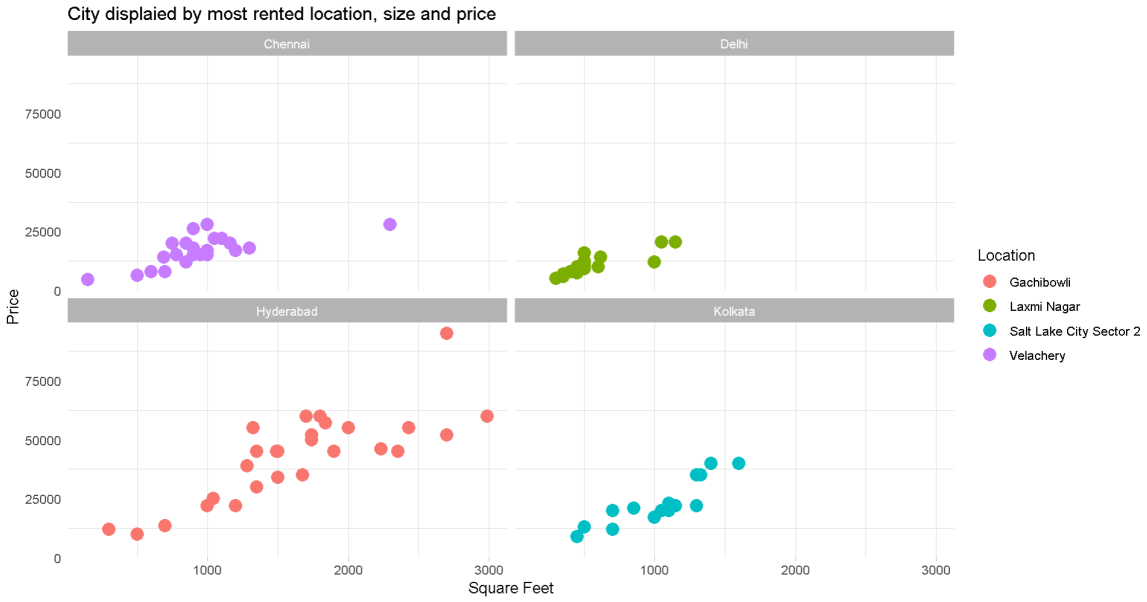

Figure 7. 27

Conclusion:

Bandra West location in Mumbai is the most rented location with 31 houses. also, Bandra West location in Mumbai is the most rented location with 31 houses. The most expensive city for renting a house in Mumbai as well with an average of 64,000 per year. That was one of the analysis found in this question. All the cities and most rented locations were analyzed by price, size, and other factors.

8.0 Question 5: Distribution of how the number of floors will affect tenants from renting houses.

8.1 Analysis: Number of houses by the total number of floors

```
anal5_1 = RentalDataset
anal5_1%>%
  group_by(Total_Floor)%>%
  summarise(Count = length(Total_Floor))%>%
  top_n(8)%>%
  ggplot(aes(Total_Floor, Count,fill = Count)) +
  geom_bar(stat='Identity') +
  labs(title='Houses by the number of floors') +
  theme_bw()
```

Figure 8. 1

This analysis is to count the number of floors by the house, the function top_n() is used to display the highest values. Geom_bar from ggolot2 is used for this analysis.

Output:

	Total_Floor	Count
1	1	314
2	2	827
3	3	878
4	4	901
5	5	402
6	6	90
7	7	168
8	8	86

Figure 8. 2

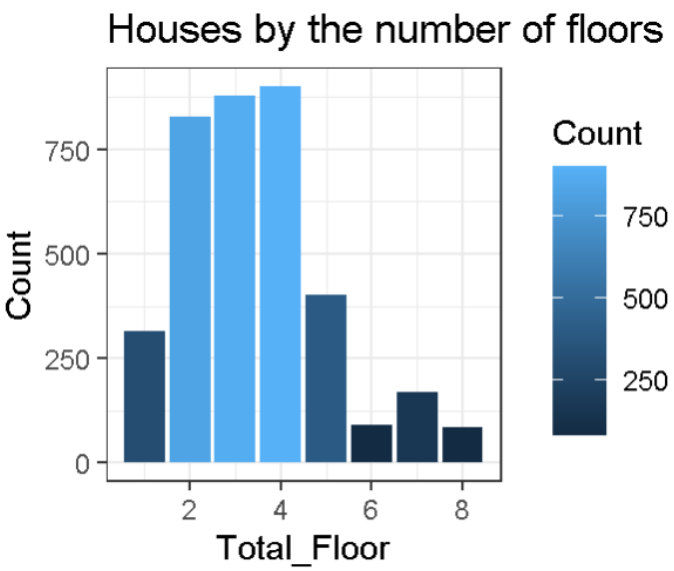

Figure 8. 3

Findings:

- Houses with four floors are recorded as the most rented houses.
- Houses with more than five total floors are likely unwanted.

8.2 Analysis: What floors do renters prefer in houses

```
anal5_2 = RentalDataset
anal5_2%>%
  group_by(Renter_Preferred,Total_Floor,Floor_Number )%>%
  summarise(Count = length(Total_Floor))%>%
  top_n(1)%>%
  view()%>%
  ggplot( aes(Floor_Number, Total_Floor, color =Renter_Preferred))+
  geom_point(size = 3, alpha = 0.8)+
  geom_smooth(method = 'glm', color = "black")+
  theme_bw()+
  facet_wrap(~Renter_Preferred)+
  labs(title = "What floors do renters prefer",
       x = "Floor_Number",
       y = "Count")
```

Figure 8. 4

As shown in the image above, this code is applied to display what renters prefer and what number of floors they have rented the most, “glm” method is used to display a clear line to show the relationship between the current floor number and how many renters have rented.

Output:

	Renter_Preferred	Floor_Number	Count
1	Bachelors	1	202
2	Bachelors/Family	1	826
3	Family	1	86

Figure 8. 5

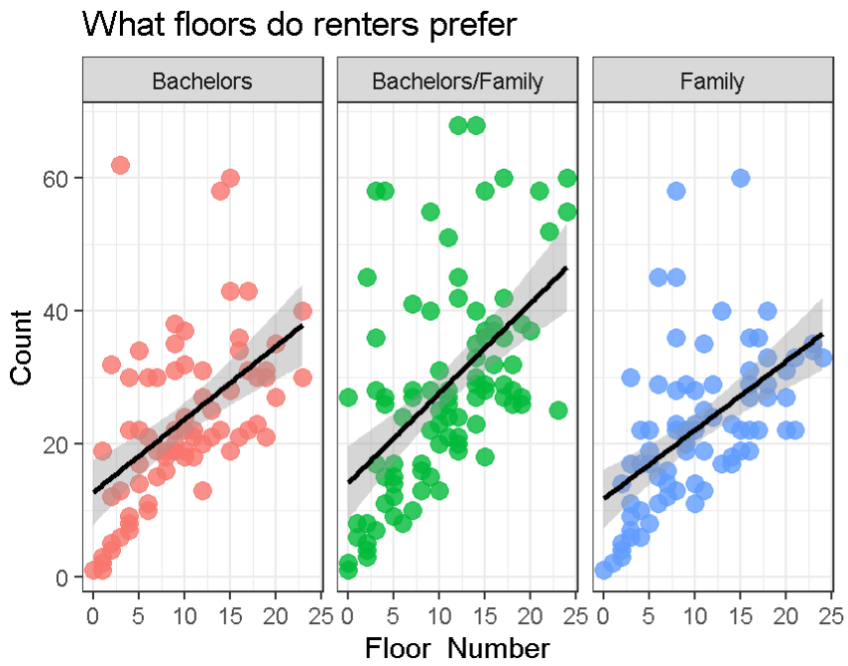

Figure 8. 6

Findings:

- The most rented houses for bachelors and families based on the current floor is the first floor
- The line clearly shows the less the house has floor the more rested.
- Bachelors tend to rent more than families on the first floor.

8.3 Analysis: What is the Average price for bachelors and families of houses by the total number of floors

```
anal5_3 = RentalDataset
anal5_3%>%
  filter(Total_Floor <= 30)%>%
  group_by(Total_Floor, Renter_Preferred)%>%
  summarise(Avg_rent = mean(Price))%>%
  view()%>%
  ggplot(aes(Total_Floor, Avg_rent,fill = Renter_Preferred)) +
  geom_bar(stat='Identity') +
  labs(title='Average price for renters of houses by total number of floors',
       x = "Total Floor",
       y = 'Avarage rent price'
       ) +
  xlim(1:30,main='')+
  theme_bw()
```

Figure 8. 7

The above code is to find the mean of the price for each house by total floors and display who bachelors or family prefer renting more floors or fewer floors. The function xlim() is used to change the scale of the axis as shown in the Output. The function filter() is used to get the first 30 houses with 30 total floors.

Output:

	Total_Floor	Renter_Preferred	Avg_rent
1	1	Bachelors	15185.48
2	1	Bachelors/Family	14126.96
3	1	Family	21475.00
4	2	Bachelors	16208.58
5	2	Bachelors/Family	13475.62
6	2	Family	14572.31
7	3	Bachelors	25692.80
8	3	Bachelors/Family	16398.73

Figure 8. 8

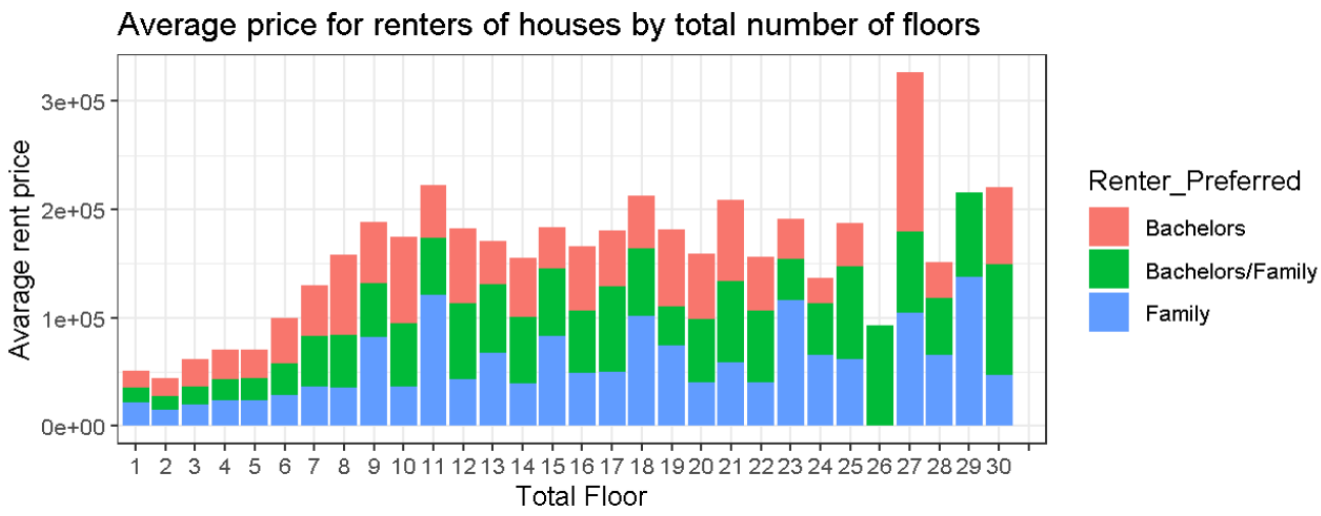

Figure 8. 9

Findings:

- The houses with less number of floors mostly are the cheapest.
- The houses with more floors mostly are the more expensive.
- The houses with 27 floors are recorded as the most expensive houses.

8.4 Analysis: Is there a relationship between the size of the house and which floor is the house on

```
anal5_4<-RentalDataset%>%
  group_by(Floor_Number)%>%
  summarise(Avg_size = mean(Square_Feet))%>%
  view()%>%
  ggplot( aes(Floor_Number, Avg_size))+
  geom_point(size = 3, alpha = 0.8)+
  geom_smooth(method = 'glm', color = "Red")+
  theme_bw()+
  labs(title = "Size of the house by which floor is the house on",
       x = "Floor Number",
       y = "Avarege size")
```

Figure 8. 10

The code above is to display if there is a relationship between the size and which floor is the house on, geaom_point() is applied to draw points on the graph then by geom_amooth(), a clean line will be displayed in the analysis to show if there any relationship.

Output:

	Floor_Number	Avg_size
1	0	860.5916
2	1	903.8106
3	2	910.1594
4	3	942.5562
5	4	948.1667
6	5	993.5723
7	6	1022.0652
8	7	1029.5915

Figure 8. 11

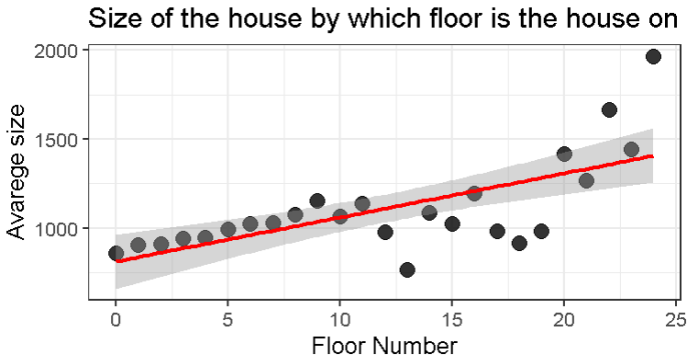

Figure 8. 12

Findings:

- A relationship between the average price and the floor number is found
- There is a direct relationship, which is the higher the floor, the larger the size of the ho

8.5 Analysis: Which city has the highest number of houses?

```
anal5_5 %>%
  ggplot( aes(x=Total_Floor, y=City, fill=City)) +
  geom_boxplot() +
  scale_fill_viridis(discrete = TRUE, alpha=0.6, option="A") +
  theme_ipsum() +
  theme(
    legend.position="none",
    plot.title = element_text(size=11)
  ) +
  xlab("")+
  labs(title = "Floors of the houses by cities",
       x = "Total floor of the house",
       y = "City")
```

Figure 8. 13

The analysis above is to check if the total floors of the house can be different from one city to another city, Boxplot() plot is used in this analysis to give a clear idea and display the range.

Output:

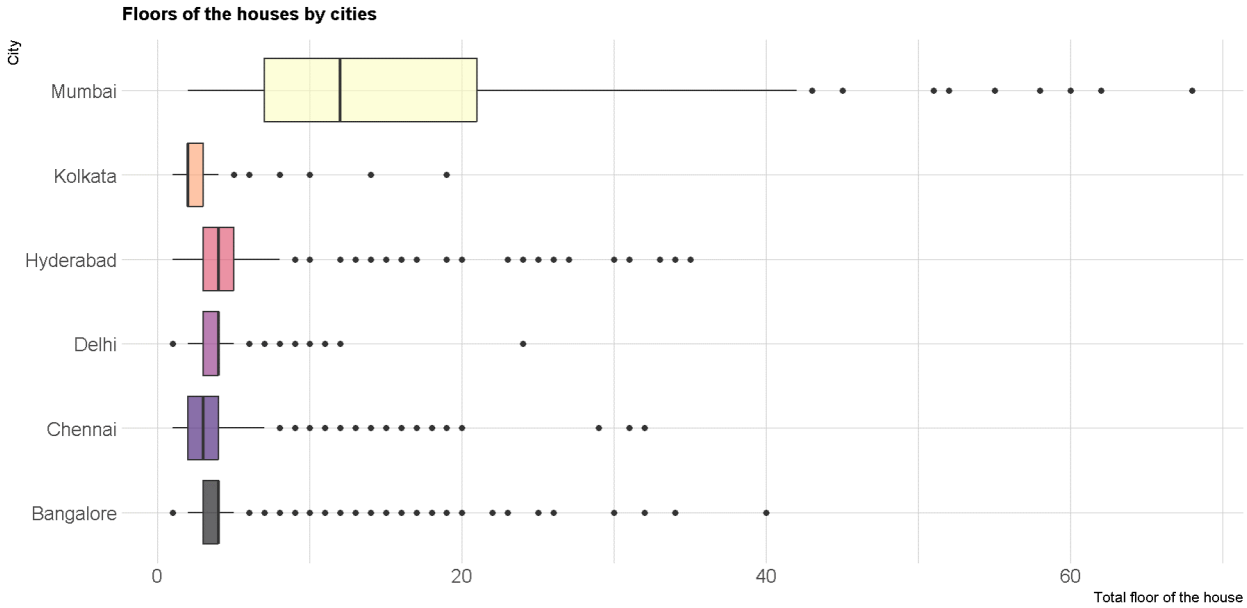

Figure 8. 14

Findings:

- The number of floors of the house differs from one city to another city.
- Mumbai has the most total floor by houses
- Kolkata has the lowest floor number by houses.

8.6 Analysis: From the most rented location, which location has the most total floors by city

```
anal5_6 = RentalDataset
anal5_6%>%
  filter(Location %in% c('Velachery','Laxmi Nagar',
                         'Gachibowli', 'Salt Lake City Sector 2'))%>%
  ggplot( aes(x=Floor_Number, y=City, fill=Location)) +
  geom_boxplot() +
  scale_fill_viridis(discrete = TRUE, alpha=0.6, option="h") +
  theme_ipsum() +
  coord_flip()+
  theme(
    plot.title = element_text(size=11)
  ) +
  xlab("")+
  labs(title = "ocation has the most total floors by city",
       x = "Total floor of the house",
       y = "City")
```

Figure 8. 15

The analysis above is to analyze if the floor of the house differs from one city to another as well, starting with filtering the most rented location that we found in the previews question. ,coord_flip() function is used here to display another style of the data

Output:

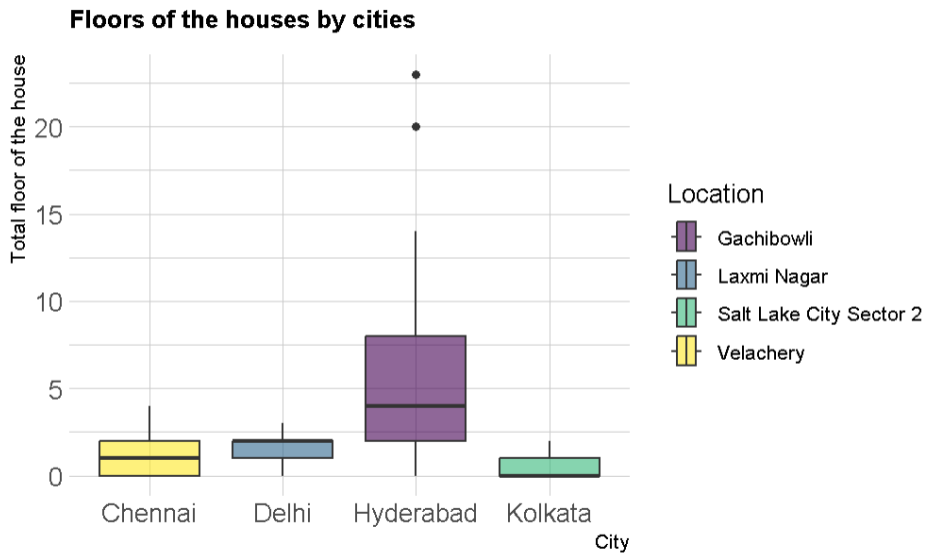

Figure 8. 16

Findings:

- The number of floors of the house differs from location to location.
- Gachibiwli located in Mumbai city has the most total floor by houses
- Salt Lake City Sector2 located in Kolkata city has the lowest total floor by houses

8.7 Analysis: Do Bachelors differ from families in choosing the floor number

```
anal5_7<-RentalDataset
anal5_7 %>%
  ggplot(aes(x=Floor_Number, y=Total_Floor, color = Renter_Preferred)) +
  geom_segment( aes(x=Floor_Number, xend=Floor_Number, y=1, yend=Total_Floor)) +
  geom_point(  size=2, alpha=0.6) +
  theme_light() +
  theme (legend.position = "none") +
  facet_wrap(~Renter_Preferred)+
  theme(
    panel.grid.major.y = element_blank(),
    panel.border = element_blank(),
    axis.ticks.y = element_blank()
  )+
  labs (title = "Bachelors and families in choosing the floor number",
        x = "Floor number",
        y = "Total floor")
```

Figure 8. 17

The analysis above is to check if there is any difference in choosing the floor number between the bachelors and families, geom_segment() is used to show a clear line between the floor number and the total number of floors.

Output:

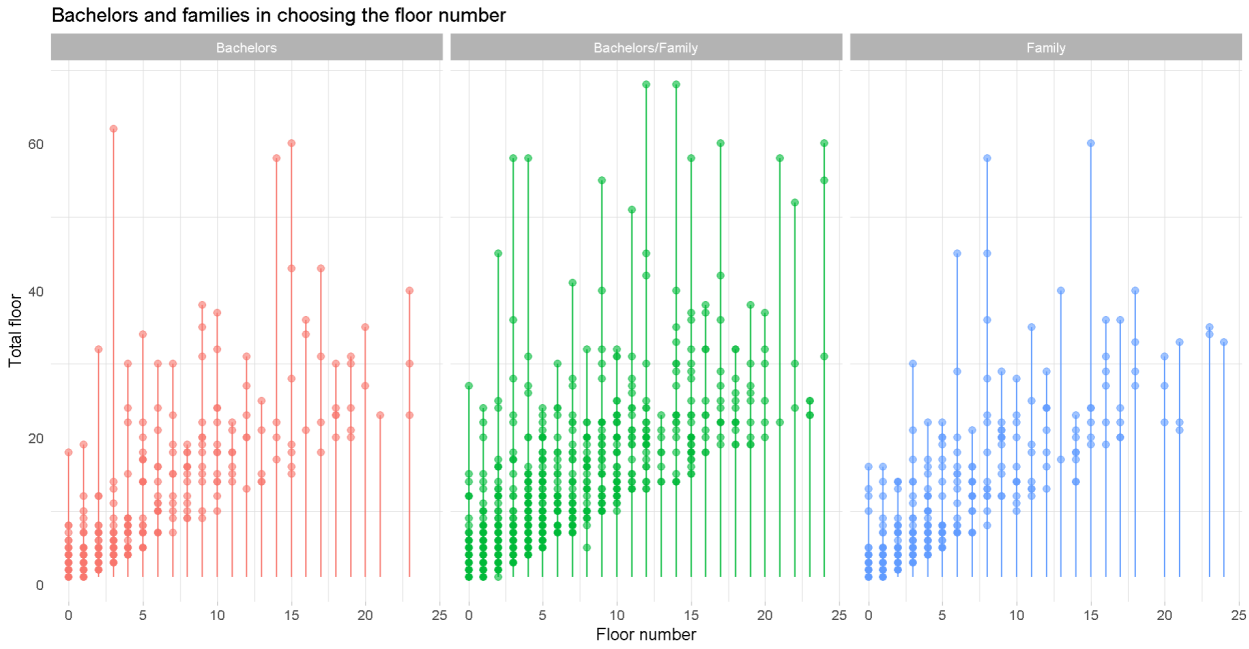

Figure 8. 18

Findings:

- There is no relationship between choosing the floor between the bachelor and families
- The more floors a house has, the more likely tenants are to rent the upper floors

8.8 Analysis: Area class differences in choosing the floor number

```
anal5_8<-RentalDataset
anal5_8 %>%
  subset( Area_Class != "Built Area")%>%
  ggplot(aes(x=Floor_Number, y=Total_Floor)) +
  geom_segment( aes(x=Floor_Number, xend=Floor_Number, y=1, yend=Total_Floor)) +
  geom_point(  size=2, alpha=0.6, color = "white") +
  theme_light() +
  facet_wrap(~Area_Class)+
  theme_dark()+
  theme(
    panel.grid.major.y = element_blank(),
    panel.border = element_blank(),
    axis.ticks.y = element_blank()
  )+
  labs (title = "Area class differences in choosing the floor number",
        x = "Floor number",
        y = "Total floor")
```

Figure 8. 19

The analysis above is to display the differences in the area type which is the Carpet area or Super area in choosing the floor number, Lollipop plot is used here to show the analysis, and facet wrap() is used to wrap every area separately.

Output:

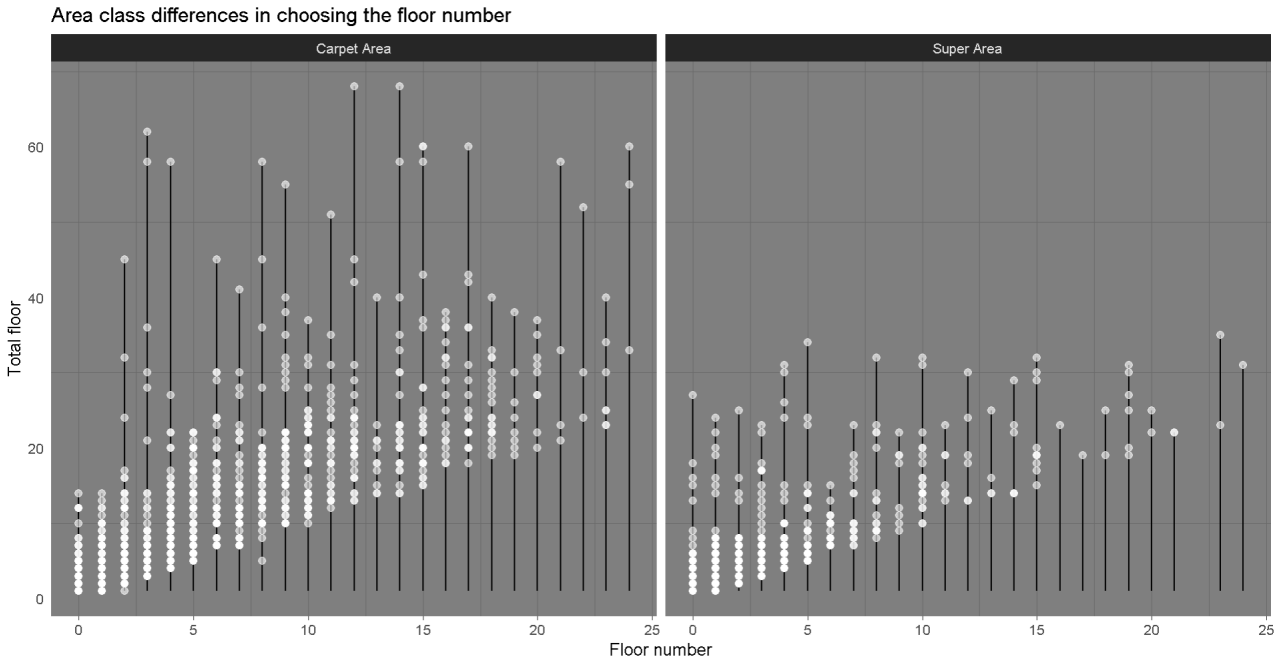

Figure 8. 20

Findings:

- There is no relationship between choosing the floor between the Area class
- Renters have chosen the carpet area more than the super area

8.9 Analysis: Contact type differences in choosing the floor number

```
anal5_9<-RentalDataset
anal5_9 %>%
  subset( Contact_Type != "Contact Builder")%>%
  ggplot(aes(x=Floor_Number, y=Total_Floor)) +
  geom_segment( aes(x=Floor_Number, xend=Floor_Number, y=1, yend=Total_Floor)) +
  geom_point(shape = 15, size=2, alpha=0.6, color = "#800000") +
  theme_light() +
  facet_wrap(~Contact_Type)+
  coord_flip()+
  theme(
    panel.grid.major.y = element_blank(),
    panel.border = element_blank(),
    axis.ticks.y = element_blank()
  )+
  labs (title = "Contact type differences in choosing the floor number ",
        x = "Floor number",
        y = "Total floor")
```

Figure 8. 21

The above code is to check if there are any differences in contact type between renters who stay on different floors, the shape of the point is changed to square by applying shape = 15 in geom_pont,

Output:

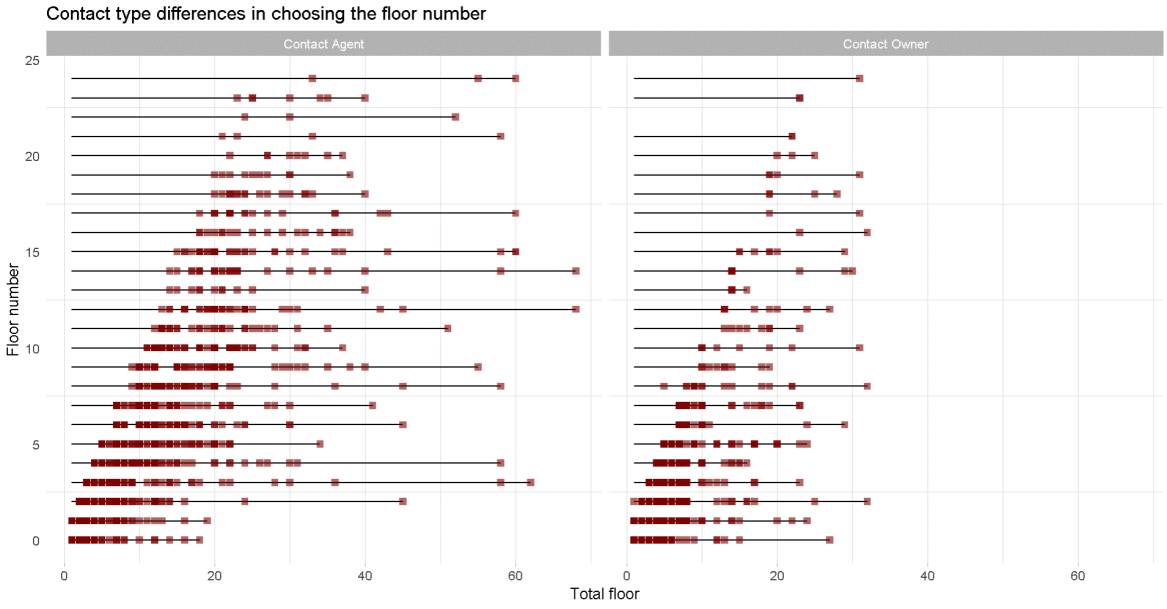

Figure 8. 22

Findings:

- There is no relationship between choosing the floor between the bachelor and families.
- Contact agents are more likely by renters than by contacting the owner.

8.10 Analysis: which furnishing type do families and bachelors prefer based on floor number

```
anal5_10<-RentalDataset
anal5_10 %>%
  ggplot(aes(x=Floor_Number, y=Total_Floor)) +
  geom_segment( aes(x=Floor_Number, xend=Floor_Number, y=1, yend=Total_Floor,
                    color = Furnishing_Types)) +
  geom_point( size=2, alpha=0.6) +
  theme_light() +
  facet_wrap(~Renter_Preferred)+
  theme(
    panel.grid.major.y = element_blank(),
    panel.border = element_blank(),
    axis.ticks.y = element_blank()
  )+
  labs (title = "Furnished type differences in choosing the floor number ",
        x = "Floor number",
        y = "Total floor")
```

Figure 8. 23

The analysis above is to check if there is any relationship between furnishing type and floor number, Furnishing type data is added as a color in geom_sugment to display a different color for each furnished type whether is fully furnished or semi-furnished, or unfurnished.

Output:

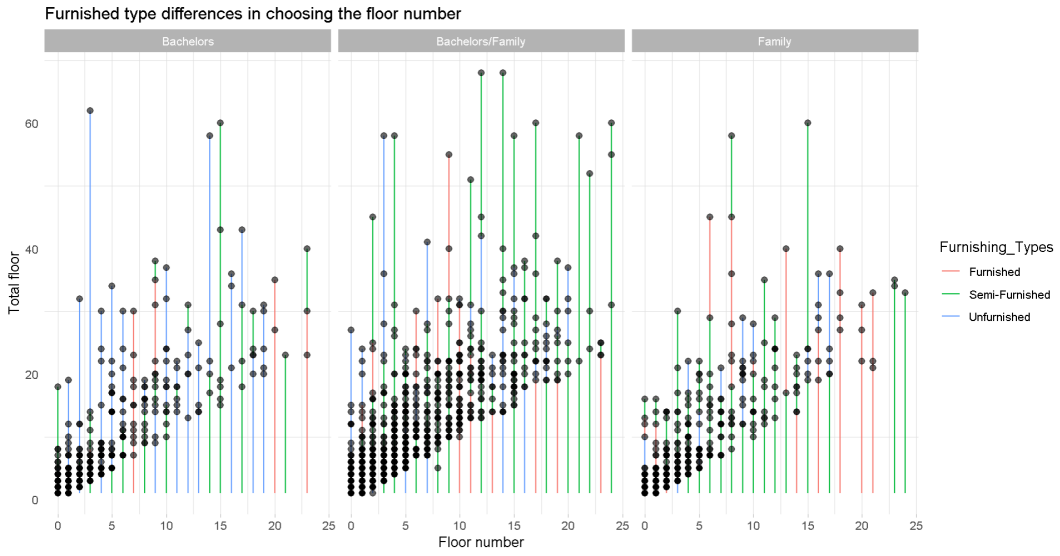

Figure 8. 24

Findings:

- There is no relationship between choosing the floor based on furnishing type between the bachelor and families.
- Bachelors are more likely to rent unfurnished houses.
- families are more likely to rent semi-furnished.

8.11 Analysis (conclusion)

```
ggplot(anal5_10,aes(x=Floor_Number, y=Total_Floor, size= Square_Feet,
                    fill = Renter_Preferred)) +
  geom_point(alpha=0.5, shape=21, color="black") +
  scale_size(range = c(.1, 10)) +
  scale_fill_viridis(discrete=TRUE, option="A") +
  facet_wrap(~City)+
  theme_ipsum() +
  guides(size = FALSE)+
  ylab("Floor Number") +
  xlab("Total Floor")
```

Figure 8. 25

The code above is to display the population of houses according to the floor number and type of renters, Bubble plot is used in this analysis, it's similar to the scatterplot bit it has three variables. It's usually used to display sizes.

Output:

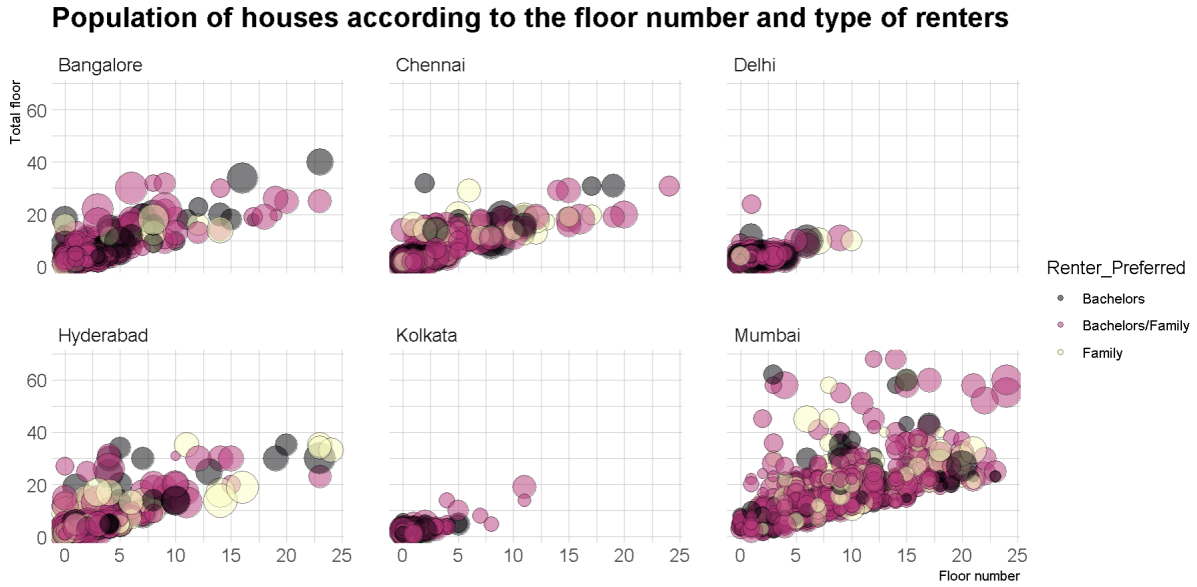

Figure 8. 26

Conclusion:

There is no relationship between choosing the floor between the bachelor and families. But on the other hand, a relationship between the average price and the floor number is found, the houses with less number of floors mostly are the cheapest. and based on the analysis the most rented houses for bachelors and families based on the current floor is the first floor, contact agents are more likely by renters than by contacting the owner. also, there is no relationship between choosing the floor between the Area class. Renters have chosen the carpet area more than the super area.

9.0 Features

9.1 Outlier Analysis

Outlier is one of the most important concepts for designers to implement before start doing any analysis, Outlier is to check if you have any values that lie away from the other values in your dataset. if the analysis started without the Outlier Implementation, it may cause some issues and not effective analyzing or visualizations.

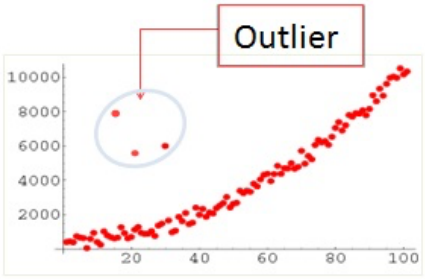

Figure 9. 1

9.1.2 Check Outlier:

```
boxplot(RentalDataset$Price, main = "Price Outlier")
```

Figure 9. 2

By using the boxolot() function, the outlier can be checked.

Output:

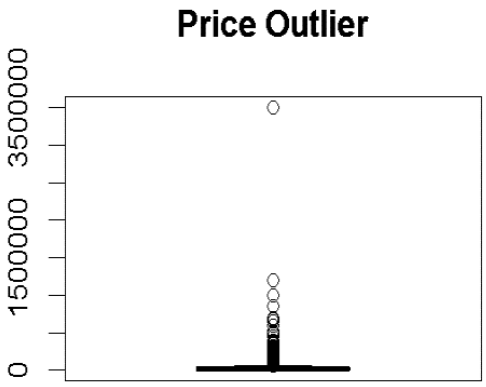

Figure 9. 3

As shown in the above figure, the is a point that lies away from other points.

9.1.3 Outlier Implementation:

```
lowwerPrice <- quantile(RentalDataset$Price, 0.010)
upperPrice <- quantile(RentalDataset$Price, 0.990)
RentalDataset <- subset(RentalDataset, RentalDataset$Price >
                          lowwerPrice & RentalDataset$Price < upperPrice)
```

Figure 9. 4

One of the methods to implement the Outlier is by quantile() functions. Quantile is used to create sample quantiles within a data set with probability zero and one.

Output:

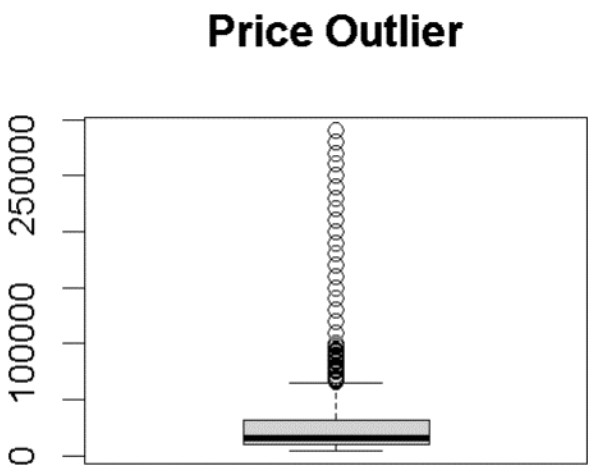

Figure 9. 5

After the quantile function is used, the Outline implementation is done by choosing the wanted range between zero and one.

Example:

Github: aimancsv

9.2 Scatterplot

Scatterplot is a plot from ggplot2 using geom_point()_ it is applied if there are two relationships between x and y in points style.

9.2.1 Scatterplot Implementation:

```
x <- runif(300, min=-30, max=30)
y <- -1.2*x^3 + 1.1 * x^2 - x + 10 + rnorm(length(x),0,100*abs(x))
```

Figure 9. 6

```
plot(x,y,col=rgb(0.4,0.4,0.8,0.6),
     pch=16 , cex=1.3 , xlab="" , ylab="")
```

Figure 9. 7

Starting by declaring the two varble needed (x, y) for the relashinship, then by using the function plot() we can implement the Scatterplot.

Output:

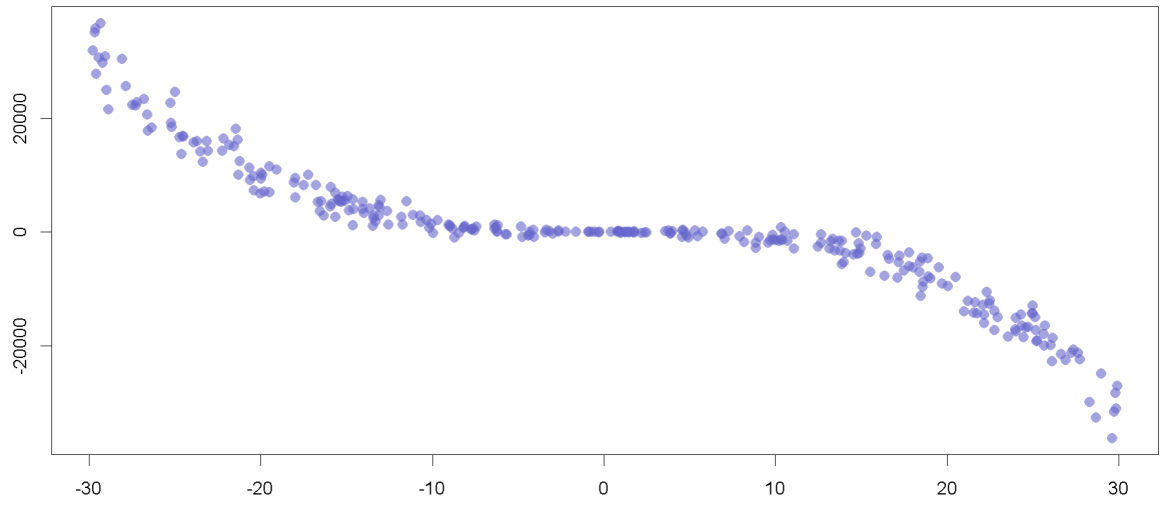

Figure 9. 8

Example:

Github: aimancsv

9.3 Boxplot

The boxplot from ggplot2, is likely one of the most popular types if there is a distribution from different factors to compare. it can be used by applying geom_jitter()

9.3.1 Boxplot Implementation:

```
data %>%
  ggplot( aes(x=name, y=value, fill=name)) +
  geom_boxplot() +
  scale_fill_viridis(discrete = TRUE, alpha=0.6) +
  geom_jitter(color="black", size=0.4, alpha=0.9) +
  theme_ipsum() +
  theme(
    legend.position="none",
    plot.title = element_text(size=11)
  ) +
  ggtitle("A boxplot with jitter") +
  xlab("")
```

Figure 9. 9

After creating the data, values for x and y should be assigned. Inside the ggplot() function, then by using geom_gitter() function and decide the features for how it should be. Boxplot will be arcived.

Output:

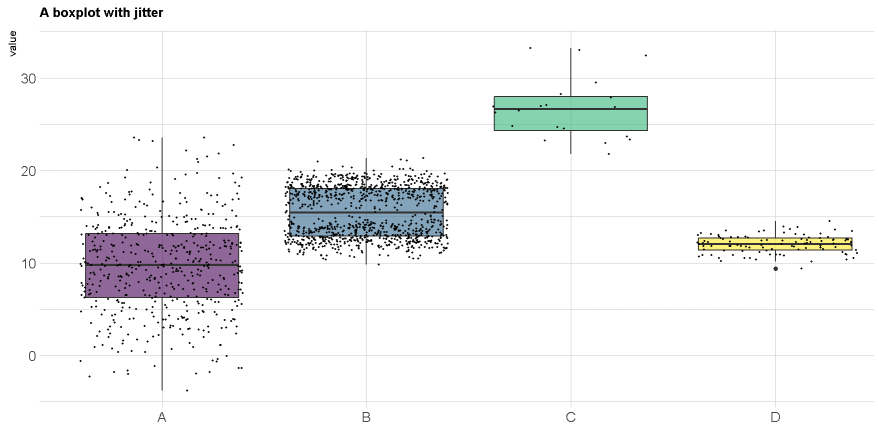

Figure 9. 10

Example:

Github: aimancsv

9.4 Bubble Plot

A Bubble Plot is a Scatterplot but the difference between Bubble Plot and Scatterplot is Bubble Plot can be used with three variables mapped. This means it can be used for more advanced data analysis and visualizations.

9.4.1 Bubble Plot Implementation:

```
data <- gapminder %>%
  filter(year=="2007") %>%
  dplyr::select(-year)
ggplot(data, aes(x=gdpPercap, y=lifeExp, size = pop)) +
  geom_point(alpha=0.7)
```

Figure 9. 11

Bobble Plot can be implemented by using ggplot()and geom_pont() functions, making sure to add the a value In the size argument in ggplot to make the bubbles.

Output:

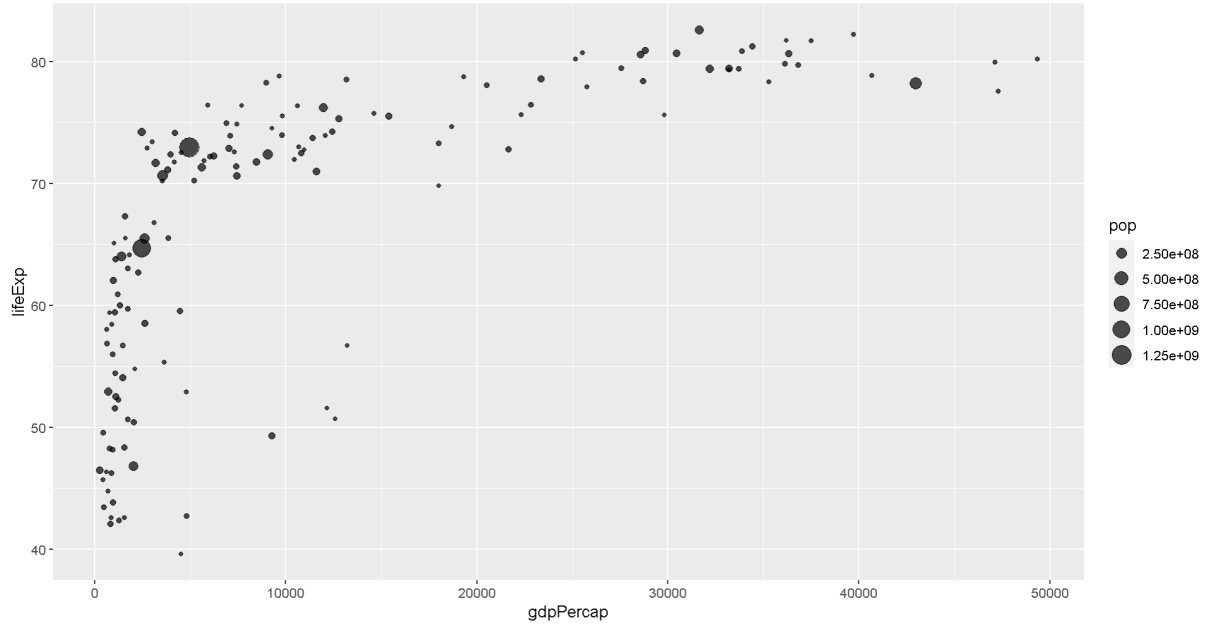

Figure 9. 12

Example: