# ASSIGNMENT

## TECHNOLOGY PARK MALAYSIA

## CT127-3-2-PFDA

## PROGRAMMING FOR DATA ANALYSIS

## APD2F2209CS(DA)

**HAND OUT DATE:** 10 OCTOBER 2022

**HAND IN DATE:** 28 NOVEMBER 2022

**WEIGHTAGE:** 50%

**STUDENT NAME:** FOO JING TZE

**STUDENT ID:** TP066056

---

**INSTRUCTIONS TO CANDIDATES:**

1. Submit your assignment at the administrative counter.
2. Students are advised to underpin their answers with the use of references (cited using the American Psychological Association (APA) Referencing).
3. Late submission will be awarded zero (0) unless Extenuating Circumstances (EC) are upheld.
4. Cases of plagiarism will be penalized.
5. The assignment should be bound in an appropriate style (comb bound or stapled).
6. Where the assignment should be submitted in both hardcopy and softcopy, the softcopy of the written assignment and source code (where appropriate) should be on a CD in an envelope / CD cover and attached to the hardcopy.
7. You must obtain 50% overall to pass this module.

# Table of Contents

# 1.0 Introduction and Assumption

## 1.1. Introduction

Nowadays, data is being collected all the time. However, without any data analytics skills, this data will not have any meanings. Therefore, by using data analytics skills such as data cleaning, data exploration, data manipulation, data transformation and data visualisation, meaningful and actionable insights will be drawn out. In this course assignment, the data set that is being deal with is about house rental and the language that will be used to perform analysis is R language using RStudio. Investigation on the data problems related to House Rent Prediction Dataset need to be made. This data contains the details of varied house rents that could determine how people have an impact to choose houses for rental based on multiple situations and provide meaningful insight for decision making.

## 1.2. Assumption

1. Assumed that the Area.Locality in this dataset is unnecessary for the analysis due to some reasons
2. Assume that the outliers of the Rent or Rental_Fee for this dataset falls below the lower limit and above upper limit
3. Assume that the Floor column refer to the floor preference in total floor numbers with the “ out of ” delimiter

## 2.0 Data Import

```
# Import data
house_rental_data = read.csv("D:\\PDFA\\House_Rent_Dataset.csv", header=TRUE)
```

*Figure 2.1: Source Code - Import Data*

```
# install packages
install.packages("plyr")
install.packages("dplyr")
install.packages("tidyr")
install.packages("ggplot2")
install.packages("tidyverse")
install.packages("scales")
install.packages("ggrepel")
install.packages("lubridate")
```

*Figure 2.2: Source Code - Install Needed Packages*

```
# call packages
library(plyr)
library(dplyr)
library(tidyr)      # used for function separate()
library(ggplot2)    # used for Visualization
library(tidyverse)
library(scales)
library(ggrepel)
library(lubridate)
```

*Figure 2.3: Source Code - Call Needed Packages*

# 3.0 Data Cleaning

## 3.1. Remove Unnecessary Columns

The purpose of removing unnecessary columns is to make it easier to focus on the variables that are used for analysis.

### Step 1: Count Unique Values of Each Column

```
/**
    * Following source code obtained from (Jim, 2022)
 */
```

```
> count_unique <- sapply(house_rental_data,function(x) n_distinct(x))
> count_unique
     Posted.On            BHK           Rent           Size          Floor      Area.Type
            81              6            243            615            480              3
 Area.Locality           City Furnishing.Status Tenant.Preferred      Bathroom Point.of.Contact
          2235              6              3              3              8              3
```

*Figure 3.1.1: Source Code and Output – Count of Unique Values in Each Column*

### Step 2: Identify Area.Locality has Many Unique Values

```
> count_unique <- sapply(house_rental_data,function(x) n_distinct(x))
> count_unique
     Posted.On            BHK           Rent           Size          Floor      Area.Type
            81              6            243            615            480              3
 Area.Locality           City Furnishing.Status Tenant.Preferred      Bathroom Point.of.Contact
          2235              6              3              3              8              3
```

*Figure 3.1.2: Identifying Column with Many Unique Values*

### Step 3: Drop Area.Locality Column

```
/**
    * Following source code obtained from (Kumar, 2022)
 */
```

```
> house_rental_data <- house_rental_data[,!names(house_rental_data) %in% c("Area.Locality")]
```

*Figure 3.1.3: Source Code and Output - Drop Area.Locality Column*

### Result

```
> names(house_rental_data)
 [1] "Posted.On"         "BHK"               "Rent"              "Size"              "Floor"
 [6] "Area.Type"         "City"              "Furnishing.Status" "Tenant.Preferred"  "Bathroom"
[11] "Point.of.Contact"
```

*Figure 3.1.4:Source Code and Output – Names of Column Headings*

Area.Locality is considered as an unnecessary column because there are too many unique data values, and these data will not be that helpful for any in-depth analysis unlike the other columns.

## 3.2. Rename Columns' Names

The purpose of renaming columns' names is to provide more meaningful heading to every column.

**Step 1: Show Names of Columns Before Modification**

```
>   names(house_rental_data)
 [1] "Posted.On"          "BHK"               "Rent"
 [4] "Size"               "Floor"             "Area.Type"
 [7] "City"               "Furnishing.Status" "Tenant.Preferred"
[10] "Bathroom"           "Point.of.Contact"
```

*Figure 3.2.1: Source Code and Output - Column Names Before Modification*

**Step 2: Rename Names of Columns**

```
/**
    * Following source code obtained from (Vermani, 2022)
*/
>   colnames(house_rental_data) <- c("Date_Posted","Bedroom_Hall_Kitchen","Rental_Fee","House_Size",
+                                    "Floor","Area_Type","City","Furnishing_Status",
+                                    "Tenant_Type","Number_of_Bathroom","Point_of_Contact")
```

*Figure 3.2.2: Source Code - Rename Columns' Names*

**Result: Names of Columns After Modification**

```
>   names(house_rental_data)
 [1] "Date_Posted"          "Bedroom_Hall_Kitchen" "Rental_Fee"           "House_Size"
 [5] "Floor"                "Area_Type"            "City"                 "Furnishing_Status"
 [9] "Tenant_Type"          "Number_of_Bathroom"   "Point_of_Contact"
```

*Figure 3.2.3: Source and Output - Column Names After Modification*

The function colnames() shown in Figure 3.2.2 is used to rename the column headings. All Columns' Names are changed by replacing dots (.) with underscore (_). The reason changing "BHK" to "Bedroom_Hall_Kitchen" is for better understanding.

## 3.3. Format Date_Posted to Date

The purpose of formatting Date_Posted is to make it easier to analyse variables among time.

```
house_rental_data$Date_Posted <- as.Date(house_rental_data$Date_Posted,"%m/%d/%Y")
View(head(house_rental_data))
```

*Figure 3.3.1: Source Code - Date Formatting*

**Result**

| | Date_Posted | Bedroom_Hall_Kitchen | Rental_Fee | House_Size | Floor_Preference | Total_Floor_Numbers | Area_Type | City | Furnishing_Status | Tenant_Type | Number_of_Bathroom | Point_of_Contact |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2022-05-18 | 2 | 10000 | 1100 | Ground | 2 | Super Area | Kolkata | Unfurnished | Bachelors/Family | 2 | Contact Owner |
| 2 | 2022-05-13 | 2 | 20000 | 800 | 1 | 3 | Super Area | Kolkata | Semi-Furnished | Bachelors/Family | 1 | Contact Owner |
| 3 | 2022-05-16 | 2 | 17000 | 1000 | 1 | 3 | Super Area | Kolkata | Semi-Furnished | Bachelors/Family | 1 | Contact Owner |
| 4 | 2022-07-04 | 2 | 10000 | 800 | 1 | 2 | Super Area | Kolkata | Unfurnished | Bachelors/Family | 1 | Contact Owner |
| 5 | 2022-05-09 | 2 | 7500 | 850 | 1 | 2 | Carpet Area | Kolkata | Unfurnished | Bachelors | 1 | Contact Owner |
| 6 | 2022-04-29 | 2 | 7000 | 600 | Ground | 1 | Super Area | Kolkata | Unfurnished | Bachelors/Family | 2 | Contact Owner |

*Figure 3.3.2: Output - Date Formatting*

## 3.4. Extract Month From Date_Posted

/**

* *Following source code obtained from (Zach, 2022)*

*/

```
house_rental_data$Month <- with(house_rental_data,month(ymd(Date_Posted)))
```

*Figure 3.4.1: Source Code – Extract Month from Date_Posted*

**Result**

| | Date_Posted | Bedroom_Hall_Kitchen | Rental_Fee | House_Size | Floor | Area_Type | City | Furnishing_Status | Tenant_Type | Number_of_Bathroom | Point_of_Contact | Month |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2022-05-18 | 2 | 10000 | 1100 | Ground out of 2 | Super Area | Kolkata | Unfurnished | Bachelors/Family | 2 | Contact Owner | 5 |
| 2 | 2022-05-13 | 2 | 20000 | 800 | 1 out of 3 | Super Area | Kolkata | Semi-Furnished | Bachelors/Family | 1 | Contact Owner | 5 |
| 3 | 2022-05-16 | 2 | 17000 | 1000 | 1 out of 3 | Super Area | Kolkata | Semi-Furnished | Bachelors/Family | 1 | Contact Owner | 5 |
| 4 | 2022-07-04 | 2 | 10000 | 800 | 1 out of 2 | Super Area | Kolkata | Unfurnished | Bachelors/Family | 1 | Contact Owner | 7 |
| 5 | 2022-05-09 | 2 | 7500 | 850 | 1 out of 2 | Carpet Area | Kolkata | Unfurnished | Bachelors | 1 | Contact Owner | 5 |
| 6 | 2022-04-29 | 2 | 7000 | 600 | Ground out of 1 | Super Area | Kolkata | Unfurnished | Bachelors/Family | 2 | Contact Owner | 4 |

*Figure 3.4.2: Output – Month Column*

## 3.5. Split Floor Column to Floor Preference and Total Floor Numbers

The purpose of splitting "Floor" column into two columns named "Floor_Preference" and "Total_Floor_Numbers" is to make it easier to analyse which floor do the tenants prefer to choose.

**Step 1: Show First 6 Data Before Splitting**

```
> View(head(house_rental_data))
```

*Figure 3.5.1: Source Code – Top 6 Data Before Splitting*

| | Date_Posted | Bedroom_Hall_Kitchen | Rental_Fee | House_Size | Floor | Area_Type | City | Furnishing_Status | Tenant_Type | Number_of_Bathroom | Point_of_Contact | Month |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2022-05-18 | 2 | 10000 | 1100 | Ground out of 2 | Super Area | Kolkata | Unfurnished | Bachelors/Family | 2 | Contact Owner | 5 |
| 2 | 2022-05-13 | 2 | 20000 | 800 | 1 out of 3 | Super Area | Kolkata | Semi-Furnished | Bachelors/Family | 1 | Contact Owner | 5 |
| 3 | 2022-05-16 | 2 | 17000 | 1000 | 1 out of 3 | Super Area | Kolkata | Semi-Furnished | Bachelors/Family | 1 | Contact Owner | 5 |
| 4 | 2022-07-04 | 2 | 10000 | 800 | 1 out of 2 | Super Area | Kolkata | Unfurnished | Bachelors/Family | 1 | Contact Owner | 7 |
| 5 | 2022-05-09 | 2 | 7500 | 850 | 1 out of 2 | Carpet Area | Kolkata | Unfurnished | Bachelors | 1 | Contact Owner | 5 |
| 6 | 2022-04-29 | 2 | 7000 | 600 | Ground out of 1 | Super Area | Kolkata | Unfurnished | Bachelors/Family | 2 | Contact Owner | 4 |

*Figure 3.5.2: Output – First 6 Rows of Data*

**Step 2: Split Floor using separate()**

/**

* *Following source code obtained from (Schork, 2020)*

*/

```
> house_rental_data <- house_rental_data %>% separate(Floor, c("Floor_Preference","Total_Floor_Numbers")," out of ")
```

*Figure 3.5.3: Source Code - Splitting Floor Column*

**Step 3: Show First 6 Data After Splitting**

```
> View(head(house_rental_data))
```

*Figure 3.5.4: Source Code – First 6 Data After Splitting*

**Result**

| | Date_Posted | Bedroom_Hall_Kitchen | Rental_Fee | House_Size | Floor_Preference | Total_Floor_Numbers | Area_Type | City | Furnishing_Status | Tenant_Type | Number_of_Bathroom | Point_of_Contact | Month |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2022-05-18 | 2 | 10000 | 1100 | Ground | 2 | Super Area | Kolkata | Unfurnished | Bachelors/Family | 2 | Contact Owner | 5 |
| 2 | 2022-05-13 | 2 | 20000 | 800 | 1 | 3 | Super Area | Kolkata | Semi-Furnished | Bachelors/Family | 1 | Contact Owner | 5 |
| 3 | 2022-05-16 | 2 | 17000 | 1000 | 1 | 3 | Super Area | Kolkata | Semi-Furnished | Bachelors/Family | 1 | Contact Owner | 5 |
| 4 | 2022-07-04 | 2 | 10000 | 800 | 1 | 2 | Super Area | Kolkata | Unfurnished | Bachelors/Family | 1 | Contact Owner | 7 |
| 5 | 2022-05-09 | 2 | 7500 | 850 | 1 | 2 | Carpet Area | Kolkata | Unfurnished | Bachelors | 1 | Contact Owner | 5 |
| 6 | 2022-04-29 | 2 | 7000 | 600 | Ground | 1 | Super Area | Kolkata | Unfurnished | Bachelors/Family | 2 | Contact Owner | 4 |

*Figure 3.5.5: Output - Data after Splitting in Table View*

## 3.6. Checking for Missing Values

It is necessary to check missing values to ensure the results are not affected by them.

```
>   colSums(is.na(house_rental_data))
        Date_Posted Bedroom_Hall_Kitchen         Rental_Fee         House_Size
                  0                    0                  0                  0
   Floor_Preference  Total_Floor_Numbers          Area_Type               City
                  0                    4                  0                  0
  Furnishing_Status         Tenant_Type Number_of_Bathroom   Point_of_Contact
                  0                    0                  0                  0
              Month
                  0
```

*Figure 3.6.1: Source Code and Output – Check Missing Values in Each Column*

The function is.na() is to check whether or not the data contains missing values. The function colSums() is used to calculate the sum of the values in each column. When they are used together, this could help to calculate the sum of missing values in each column. As shown in Figure 3.6.1, there are 4 missing values found in Total_Floor_Numbers after splitting columns.

## 3.7. Handling Missing Values

The purpose of handling missing values is to avoid biased result during analysis.

**Step 1: Show Which Rows Have Missing Values**

```
/**
    * Following source code obtained from (Zach, 2021)
 */
```

```
>   which(is.na(house_rental_data$Total_Floor_Numbers))
[1] 2554 2884 4491 4561
```

*Figure 3.7.1: Source Code and Output –Rows with Missing Values*

As shown in Figure 3.7.1, rows number 2554, 2884, 4491 and 4561 have missing values in Total_Floor_Numbers column. The reason with missing values in Total_Floor_Numbers column is that some of the values in the previous Floor column only showed single value such as 3, Ground and 1 without the delimiter " out of ".

## Step 2: Calculate Mean to Perform Mean Imputation

/**

** Following source code obtained from (Zach, 2020)*

*/

```
> floor_mean <-round(mean(as.integer(house_rental_data$Total_Floor_Numbers),na.rm=TRUE),0)
> floor_mean
[1] 7
```

*Figure 3.7.2: Mean Calculation of Total_Floor_Numbers*

In Figure 3.7.2, function as.integer() is used to convert the data type of Total_Floor_Numbers from character to integer. Function mean() is used to calculate the mean of Total_Floor_Numbers. Function round() is to round the calculated mean value to 0 decimal place and the reason of rounding is because this column is considered as discrete data.

## Step 3: Mean Imputation

/**

** Following source code obtained from (Zach, 2020)*

*/

```
> house_rental_data$Total_Floor_Numbers[is.na(house_rental_data$Total_Floor_Numbers)] <- floor_mean
```

*Figure 3.7.3: Fill in Missing Values with Calculated Mean of Total_Floor_Numbers*

## Step 4: Check Missing Values Again

```
>   colSums(is.na(house_rental_data))
       Date_Posted Bedroom_Hall_Kitchen            Rental_Fee             House_Size
                 0                    0                     0                      0
  Floor_Preference   Total_Floor_Numbers             Area_Type                   City
                 0                    0                     0                      0
 Furnishing_Status            Tenant_Type    Number_of_Bathroom      Point_of_Contact
                 0                    0                     0                      0
             Month
                 0
```

*Figure 3.7.4: Second Check on Missing Values*

## Step 5: Check Row Numbers that have Missing Values Previously

```
> # Step 5: Check row number 2554,2884,4491 & 4561
> View(house_rental_data[c(2554,2884,4491,4561),])
```

*Figure 3.7.5: Source Code - View Row Number 2554, 2884, 4491 and 4561*

## Result

| | Date_Posted | Bedroom_Hall_Kitchen | Rental_Fee | House_Size | Floor_Preference | Total_Floor_Numbers | Area_Type | City | Furnishing_Status | Tenant_Type | Number_of_Bathroom | Point_of_Contact | Month |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2554 | 2022-06-18 | 2 | 20000 | 400 | 3 | 7 | Super Area | Delhi | Unfurnished | Bachelors/Family | 1 | Contact Owner | 6 |
| 2884 | 2022-05-23 | 1 | 18000 | 450 | Ground | 7 | Carpet Area | Delhi | Furnished | Bachelors/Family | 1 | Contact Owner | 5 |
| 4491 | 2022-06-12 | 3 | 15000 | 900 | 1 | 7 | Super Area | Hyderabad | Semi-Furnished | Bachelors/Family | 3 | Contact Owner | 6 |
| 4561 | 2022-05-31 | 3 | 15000 | 1270 | 1 | 7 | Carpet Area | Hyderabad | Furnished | Family | 2 | Contact Owner | 5 |

*Figure 3.7.6: Output - Row Number 2554, 2884, 4491 and 4561*

The missing values in Total_Floor_Numbers in these 4 rows have filled with the mean of Total_Floor_Numbers which is 7.

## 3.8. Replace Values which are Character in Floor_Preference

The purpose of replacing values in Floor_Preference is to standardize the format type so that it is standardized for the graph plotting later.

**Step 1: Show the Categories in Floor_Preference**

```
> levels(factor(house_rental_data$Floor_Preference))
 [1] "1"             "10"            "11"            "12"            "13"
 [6] "14"            "15"            "16"            "17"            "18"
[11] "19"            "2"             "20"            "21"            "22"
[16] "23"            "24"            "25"            "26"            "27"
[21] "28"            "29"            "3"             "30"            "32"
[26] "33"            "34"            "35"            "36"            "37"
[31] "39"            "4"             "40"            "41"            "43"
[36] "44"            "45"            "46"            "47"            "48"
[41] "49"            "5"             "50"            "53"            "6"
[46] "60"            "65"            "7"             "76"            "8"
[51] "9"             "Ground"        "Lower Basement" "Upper Basement"
```

*Figure 3.8.1: Categories in Floor_Preference*

**Step 2: Represent the Categories with Numbers**

```
> # represent Ground as 0
> house_rental_data$Floor_Preference[which(house_rental_data$Floor_Preference == "Ground")] <- 0
>
> # represent Upper Basement as -1
> house_rental_data$Floor_Preference[which(house_rental_data$Floor_Preference == "Upper Basement")] <- -1
>
> # represent Lower Basement as -2
> house_rental_data$Floor_Preference[which(house_rental_data$Floor_Preference == "Lower Basement")] <- -2
```

*Figure 3.8.2: Source Code - Representation of Characters with Values*

**Result**

```
> levels(factor(house_rental_data$Floor_Preference))
 [1] "-1" "-2" "0"  "1"  "10" "11" "12" "13" "14" "15" "16" "17" "18" "19"
[15] "2"  "20" "21" "22" "23" "24" "25" "26" "27" "28" "29" "3"  "30" "32"
[29] "33" "34" "35" "36" "37" "39" "4"  "40" "41" "43" "44" "45" "46" "47"
[43] "48" "49" "5"  "50" "53" "6"  "60" "65" "7"  "76" "8"  "9"
```

*Figure 3.8.3: Source Code and Output - Categories in Floor_Preference*

As shown in Figure 3.8.3, there are no more “Ground”, Lower Basement” and “Upper Basement”. Instead, they have been replaced with “-1”, “-2” and “0”.

## 3.9. Swap Values if Floor_Preference is Greater than Total_Floor_Numbers

The purpose of swapping values is to make sure there is no error. Logically, tenants could not choose house which the total floor numbers is lower than the one they preferred. Therefore, swapping is needed for those data that have Floor_Preference greater than the Total_Floor_Numbers.

**Step 1: Checking Class of Columns before Compare Values between Two Columns**

```
> class(house_rental_data$Floor_Preference)
[1] "character"
> class(house_rental_data$Total_Floor_Numbers)
[1] "character"
```

*Figure 3.9.1: Source Code and Output - Class of Floor_Preference and Total_Floor_Numbers*

## Step 2: Change the Class of Floor_Preference and Total_Floor_Numbers to Numeric

```
> house_rental_data$Floor_Preference <- as.numeric(house_rental_data$Floor_Preference)
> house_rental_data$Total_Floor_Numbers <- as.numeric(house_rental_data$Total_Floor_Numbers)
```

*Figure 3.9.2: Source Code - Convert the Class of Floor_Preference and Total_Floor_Numbers to Numeric*

This step is to convert the class of both columns into numeric so that they can be used in the calculation.

## Step 3: Show the Class of Floor_Preference and Total_Floor_Numbers

```
> class(house_rental_data$Floor_Preference)
[1] "numeric"
> class(house_rental_data$Total_Floor_Numbers)
[1] "numeric"
```

*Figure 3.9.3: Source Code and Output - Class of Floor_Preference and Total_Floor_Numbers*

## Step 4: Show Rows with Greater Floor_Preference than Total_Floor_Numbers

```
View(subset(house_rental_data,Floor_Preference>Total_Floor_Numbers))
```

*Figure 3.9.4: Source Code - Rows with Greater Floor_Preference than Total_Floor_Numbers*

| | Date_Posted | Bedroom_Hall_Kitchen | Rental_Fee | House_Size | Floor_Preference | Total_Floor_Numbers | Area_Type | City | Furnishing_Status | Tenant_Type | Number_of_Bathroom | Point_of_Contact | Month |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 106 | 2022-06-06 | 1 | 6000 | 600 | 8 | 5 | Carpet Area | Kolkata | Unfurnished | Bachelors/Family | 1 | Contact Owner | 6 |
| 162 | 2022-06-27 | 2 | 10000 | 450 | 2 | 1 | Carpet Area | Kolkata | Semi-Furnished | Bachelors/Family | 2 | Contact Owner | 6 |

*Figure 3.9.5: Output - Rows with Greater Floor_Preference than Total_Floor_Numbers*

As shown in Figure 3.9.5, rows number 106 and 162 have the values of Floor_Preference greater than Total_Floor_Numbers.

## Step 5: Swap between Floor_Preference and Total_Floor_Numbers

```
temp_max = pmax(house_rental_data$Floor_Preference,house_rental_data$Total_Floor_Numbers)
house_rental_data$Floor_Preference = pmin(house_rental_data$Floor_Preference,house_rental_data$Total_Floor_Numbers)
house_rental_data$Total_Floor_Numbers = temp_max
View(house_rental_data[c(106,162),])
```

*Figure 3.9.6: Source Code - Swap between Floor_Preference and Total_Floor_Numbers*

## Result

| | Date_Posted | Bedroom_Hall_Kitchen | Rental_Fee | House_Size | Floor_Preference | Total_Floor_Numbers | Area_Type | City | Furnishing_Status | Tenant_Type | Number_of_Bathroom | Point_of_Contact | Month |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 106 | 2022-06-06 | 1 | 6000 | 600 | 5 | 8 | Carpet Area | Kolkata | Unfurnished | Bachelors/Family | 1 | Contact Owner | 6 |
| 162 | 2022-06-27 | 2 | 10000 | 450 | 1 | 2 | Carpet Area | Kolkata | Semi-Furnished | Bachelors/Family | 2 | Contact Owner | 6 |

*Figure 3.8.4: Output - Data after Swapping in Table Form*

## 3.10. New Column Rental_Fee_Status

```
house_rental_data$Rental_Fee_Status <- with(house_rental_data,
                                            ifelse(Rental_Fee > mean(Rental_Fee),"high","low"))
```

*Figure 3.10.1: Source Code – Create New Column Rental_Fee_Status with Condition of Rental_Fee greater than Mean of Rental_Fee*

**Result**

| | Date_Posted | Bedroom_Hall_Kitchen | Rental_Fee | House_Size | Floor_Preference | Total_Floor_Numbers | Area_Type | City | Furnishing_Status | Tenant_Type | Number_of_Bathroom | Point_of_Contact | Month | Rental_Fee_Status |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2022-05-18 | 2 | 10000 | 1100 | 0 | 2 | Super Area | Kolkata | Unfurnished | Bachelors/Family | 2 | Contact Owner | 5 | low |
| 2 | 2022-05-13 | 2 | 20000 | 800 | 1 | 3 | Super Area | Kolkata | Semi-Furnished | Bachelors/Family | 1 | Contact Owner | 5 | low |
| 3 | 2022-05-16 | 2 | 17000 | 1000 | 1 | 3 | Super Area | Kolkata | Semi-Furnished | Bachelors/Family | 1 | Contact Owner | 5 | low |
| 4 | 2022-07-04 | 2 | 10000 | 800 | 1 | 2 | Super Area | Kolkata | Unfurnished | Bachelors/Family | 1 | Contact Owner | 7 | low |
| 5 | 2022-05-09 | 2 | 7500 | 850 | 1 | 2 | Carpet Area | Kolkata | Unfurnished | Bachelors | 1 | Contact Owner | 5 | low |
| 6 | 2022-04-29 | 2 | 7000 | 600 | 0 | 1 | Super Area | Kolkata | Unfurnished | Bachelors/Family | 2 | Contact Owner | 4 | low |

*Figure 3.10.2: Output: First 6 Data in Table Form*

## 3.11. Rearrange Columns Order

```
house_rental_data <- select(house_rental_data,Date_Posted,Month,Point_of_Contact,Tenant_Type,Bedroom_Hall_Kitchen,
                            Number_of_Bathroom,Floor_Preference,Total_Floor_Numbers,House_Size,Rental_Fee,
                            Rental_Fee_Status,Furnishing_Status,Area_Type,City)
```

*Figure 3.11.1: Source Code – Rearrange Columns Order*

**Result**

| | Date_Posted | Month | Point_of_Contact | Tenant_Type | Bedroom_Hall_Kitchen | Number_of_Bathroom | Floor_Preference | Total_Floor_Numbers | House_Size | Rental_Fee | Rental_Fee_Status | Furnishing_Status | Area_Type | City |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2022-05-18 | 5 | Contact Owner | Bachelors/Family | 2 | 2 | 0 | 2 | 1100 | 10000 | low | Unfurnished | Super Area | Kolkata |
| 2 | 2022-05-13 | 5 | Contact Owner | Bachelors/Family | 2 | 1 | 1 | 3 | 800 | 20000 | low | Semi-Furnished | Super Area | Kolkata |
| 3 | 2022-05-16 | 5 | Contact Owner | Bachelors/Family | 2 | 1 | 1 | 3 | 1000 | 17000 | low | Semi-Furnished | Super Area | Kolkata |
| 4 | 2022-07-04 | 7 | Contact Owner | Bachelors/Family | 2 | 1 | 1 | 2 | 800 | 10000 | low | Unfurnished | Super Area | Kolkata |
| 5 | 2022-05-09 | 5 | Contact Owner | Bachelors | 2 | 1 | 1 | 2 | 850 | 7500 | low | Unfurnished | Carpet Area | Kolkata |
| 6 | 2022-04-29 | 4 | Contact Owner | Bachelors/Family | 2 | 2 | 0 | 1 | 600 | 7000 | low | Unfurnished | Super Area | Kolkata |

*Figure 3.11.2: Output – New Columns Arrangement*

## 3.12. Remove Outliers by Creating Function

/**

* *Following source code obtained from (Zach, 2021)*

*/

```
> outliers <- function(x) {
+   Q1 <- quantile(x, probs=.25)
+   Q3 <- quantile(x, probs=.75)
+   iqr = Q3-Q1
+
+   upper_limit = Q3 + (iqr*1.5)
+   lower_limit = Q1 - (iqr*1.5)
+
+   x > upper_limit | x < lower_limit
+ }
>
> remove_outliers <- function(house_rental_data, cols = names(house_rental_data)) {
+   for (col in cols) {
+     house_rental_data <- house_rental_data[!outliers(house_rental_data[[col]]),]
+   }
+   house_rental_data
+ }
```

*Figure 3.12.1: Source Code – Creating Function to Remove Outliers*

# 4.0 Data Exploration

## 4.1. Show Structure of Data

```
> str(house_rental_data)
'data.frame':   4746 obs. of  14 variables:
 $ Date_Posted         : Date, format: "2022-05-18" "2022-05-13" ...
 $ Month               : num  5 5 5 7 5 4 6 6 6 6 ...
 $ Point_of_Contact    : chr  "Contact Owner" "Contact Owner" "Contact Owner" "Contact Owner" ...
 $ Tenant_Type         : chr  "Bachelors/Family" "Bachelors/Family" "Bachelors/Family" "Bachelors/Family" ...
 $ Bedroom_Hall_Kitchen: int  2 2 2 2 2 2 2 1 2 2 ...
 $ Number_of_Bathroom  : int  2 1 1 1 1 2 2 1 2 2 ...
 $ Floor_Preference    : num  0 1 1 1 1 0 0 1 1 1 ...
 $ Total_Floor_Numbers : num  2 3 3 2 2 1 4 2 2 3 ...
 $ House_Size          : int  1100 800 1000 800 850 600 700 250 800 1000 ...
 $ Rental_Fee          : int  10000 20000 17000 10000 7500 7000 10000 5000 26000 10000 ...
 $ Rental_Fee_Status   : chr  "low" "low" "low" "low" ...
 $ Furnishing_Status   : chr  "Unfurnished" "Semi-Furnished" "Semi-Furnished" "Unfurnished" ...
 $ Area_Type           : chr  "Super Area" "Super Area" "Super Area" "Super Area" ...
 $ City                : chr  "Kolkata" "Kolkata" "Kolkata" "Kolkata" ...
```

*Figure 4.1.1: Source Code and Output - Structure of house_rental_data*

Function str() is used to display the internal structure of a given object. In this case, the object is the dataset being imported in the beginning. The purpose of using this is to know about the column objects and its constituents (Zach, 2022). As shown in Figure 4.1.1, the object has a class of data.frame. The data frame has 4746 observations (rows) and 14 variables (columns).

## 4.2. Show Number of Rows and Columns

```
> dim(house_rental_data)
[1] 4746   14
```

*Figure 4.2.1: Source Code and Output - Number of Rows and Columns*

Function dim() is used to get the dimensions of data frame. This will retrieve the number of rows and columns in the data frame. There are 4746 rows and 14 columns in the data frame.

## 4.3. Show First 10 Data

```
> View(head(house_rental_data,10))
```

*Figure 4.3.1: Source Code - First 10 Rows of Data*

**Result**

| | Date_Posted | Month | Point_of_Contact | Tenant_Type | Bedroom_Hall_Kitchen | Number_of_Bathroom | Floor_Preference | Total_Floor_Numbers | House_Size | Rental_Fee | Rental_Fee_Status | Furnishing_Status | Area_Type | City |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2022-05-18 | 5 | Contact Owner | Bachelors/Family | 2 | 2 | 0 | 2 | 1100 | 10000 | low | Unfurnished | Super Area | Kolkata |
| 2 | 2022-05-13 | 5 | Contact Owner | Bachelors/Family | 2 | 1 | 1 | 3 | 800 | 20000 | low | Semi-Furnished | Super Area | Kolkata |
| 3 | 2022-05-16 | 5 | Contact Owner | Bachelors/Family | 2 | 1 | 1 | 3 | 1000 | 17000 | low | Semi-Furnished | Super Area | Kolkata |
| 4 | 2022-07-04 | 7 | Contact Owner | Bachelors/Family | 2 | 1 | 1 | 2 | 800 | 10000 | low | Unfurnished | Super Area | Kolkata |
| 5 | 2022-05-09 | 5 | Contact Owner | Bachelors | 2 | 1 | 1 | 2 | 850 | 7500 | low | Unfurnished | Carpet Area | Kolkata |
| 6 | 2022-04-29 | 4 | Contact Owner | Bachelors/Family | 2 | 2 | 0 | 1 | 600 | 7000 | low | Unfurnished | Super Area | Kolkata |
| 7 | 2022-06-21 | 6 | Contact Agent | Bachelors | 2 | 2 | 0 | 4 | 700 | 10000 | low | Unfurnished | Super Area | Kolkata |
| 8 | 2022-06-21 | 6 | Contact Agent | Bachelors | 1 | 1 | 1 | 2 | 250 | 5000 | low | Unfurnished | Super Area | Kolkata |
| 9 | 2022-06-07 | 6 | Contact Agent | Bachelors | 2 | 2 | 1 | 2 | 800 | 26000 | low | Unfurnished | Carpet Area | Kolkata |
| 10 | 2022-06-20 | 6 | Contact Owner | Bachelors/Family | 2 | 2 | 1 | 3 | 1000 | 10000 | low | Semi-Furnished | Carpet Area | Kolkata |

*Figure 4.3.2: Output - First 10 Rows of Data*

Function head() is used to retrieve the first n rows of the data. Function View() is used to display a spreadsheet style data viewer within RStudio.

## 4.4. Show Last 10 Data

```
> View(tail(house_rental_data,10))
```

*Figure 4.4.1: Source Code - Last 10 Rows of Data*

**Result**

| | Date_Posted | Month | Point_of_Contact | Tenant_Type | Bedroom_Hall_Kitchen | Number_of_Bathroom | Floor_Preference | Total_Floor_Numbers | House_Size | Rental_Fee | Rental_Fee_Status | Furnishing_Status | Area_Type | City |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4737 | 2022-06-28 | 6 | Contact Owner | Family | 3 | 3 | -2 | 2 | 1500 | 15000 | low | Semi-Furnished | Super Area | Hyderabad |
| 4738 | 2022-07-07 | 7 | Contact Owner | Bachelors/Family | 3 | 3 | -2 | 2 | 1500 | 15000 | low | Semi-Furnished | Super Area | Hyderabad |
| 4739 | 2022-07-06 | 7 | Contact Agent | Bachelors | 2 | 2 | 4 | 5 | 855 | 17000 | low | Unfurnished | Carpet Area | Hyderabad |
| 4740 | 2022-07-06 | 7 | Contact Owner | Bachelors | 2 | 2 | 2 | 4 | 1040 | 25000 | low | Unfurnished | Carpet Area | Hyderabad |
| 4741 | 2022-06-02 | 6 | Contact Owner | Bachelors/Family | 2 | 2 | 2 | 2 | 1350 | 12000 | low | Unfurnished | Super Area | Hyderabad |
| 4742 | 2022-05-18 | 5 | Contact Owner | Bachelors/Family | 2 | 2 | 3 | 5 | 1000 | 15000 | low | Semi-Furnished | Carpet Area | Hyderabad |
| 4743 | 2022-05-15 | 5 | Contact Owner | Bachelors/Family | 3 | 3 | 1 | 4 | 2000 | 29000 | low | Semi-Furnished | Super Area | Hyderabad |
| 4744 | 2022-07-10 | 7 | Contact Agent | Bachelors/Family | 3 | 3 | 3 | 5 | 1750 | 35000 | high | Semi-Furnished | Carpet Area | Hyderabad |
| 4745 | 2022-07-06 | 7 | Contact Agent | Family | 3 | 2 | 23 | 34 | 1500 | 45000 | high | Semi-Furnished | Carpet Area | Hyderabad |
| 4746 | 2022-05-04 | 5 | Contact Owner | Bachelors | 2 | 2 | 4 | 5 | 1000 | 15000 | low | Unfurnished | Carpet Area | Hyderabad |

*Figure 4.4.2: Output - Last 10 Rows of Data*

Function tail() is used to retrieve the last n rows of the data.

## 4.5. Show All Columns Names

```
> names(house_rental_data)
 [1] "Date_Posted"          "Month"                "Point_of_Contact"
 [4] "Tenant_Type"          "Bedroom_Hall_Kitchen" "Number_of_Bathroom"
 [7] "Floor_Preference"     "Total_Floor_Numbers"  "House_Size"
[10] "Rental_Fee"           "Rental_Fee_Status"    "Furnishing_Status"
[13] "Area_Type"            "City"
```

*Figure 4.5.1: Source Code and Output - Columns Names*

Function names() is used to get the names of the object. The object in this case is the data set in data frame format.

## 4.6. Count the Unique Values in Bedroom_Hall_Kitchen

/**

** Following source code obtained from (Zach, 2021)*

**/*

```
> # Show Number of Rows with the category of BHK
> house_rental_data %>% count(Bedroom_Hall_Kitchen)
  Bedroom_Hall_Kitchen    n
1                    1 1167
2                    2 2265
3                    3 1098
4                    4  189
5                    5   19
6                    6    8
```

*Figure 4.6.1: Source Code and Output - Number of House by Bedroom_Hall_Kitchen*

The pipe operator is represented as %>% and this is used to perform a sequence of operations on a data frame. Function count() is used to count the unique values of one or more variables (Sturis, 2021).

## 4.7. Count the Unique Values in Area Type

/**

* *Following source code obtained from (Zach, 2021)*

*/

```
> # Show Number of Houses with the category of area
> house_rental_data %>% count(Area_Type)
    Area_Type    n
1  Built Area    2
2 Carpet Area 2298
3  Super Area 2446
```

*Figure 4.7.1: Source Code and Output - Number of Houses by Area_Type*

## 4.8. Count the Unique Values in City and Area Type

/**

* *Following source code obtained from (Zach, 2021)*

*/

```
> # Show Number of Houses in each area with the category of City
> house_rental_data %>% count(City,Area_Type)
        City   Area_Type   n
1  Bangalore Carpet Area 336
2  Bangalore  Super Area 550
3    Chennai  Built Area   1
4    Chennai Carpet Area 315
5    Chennai  Super Area 575
6      Delhi Carpet Area 315
7      Delhi  Super Area 290
8  Hyderabad  Built Area   1
9  Hyderabad Carpet Area 246
10 Hyderabad  Super Area 621
11   Kolkata Carpet Area 264
12   Kolkata  Super Area 260
13    Mumbai Carpet Area 822
14    Mumbai  Super Area 150
```

*Figure 4.8.1: Source Code and Output - Number of Houses by City and Area_Type*

## 4.9. Count the Unique Values in City and Furnishing Status

```
> # show Number of Houses in each city with the category of Furnishing_Status
> house_rental_data %>% count(City,Furnishing_Status)
        City Furnishing_Status   n
1  Bangalore         Furnished  91
2  Bangalore    Semi-Furnished 584
3  Bangalore       Unfurnished 211
4    Chennai         Furnished  81
5    Chennai    Semi-Furnished 448
6    Chennai       Unfurnished 362
7      Delhi         Furnished  97
8      Delhi    Semi-Furnished 278
9      Delhi       Unfurnished 230
10 Hyderabad         Furnished 111
11 Hyderabad    Semi-Furnished 402
12 Hyderabad       Unfurnished 355
13   Kolkata         Furnished  61
14   Kolkata    Semi-Furnished 142
15   Kolkata       Unfurnished 321
16    Mumbai         Furnished 239
17    Mumbai    Semi-Furnished 397
18    Mumbai       Unfurnished 336
```

*Figure 4.9.1: Source Code and Output – Number of Houses by City and Furnishing Status*

### 4.10. Show the Details of Each Column

```
> summary(house_rental_data)
  Date_Posted            Month          Point_of_Contact   Tenant_Type
 Min.   :2022-04-13   Min.   :4.000   Length:4746        Length:4746
 1st Qu.:2022-05-20   1st Qu.:5.000   Class :character   Class :character
 Median :2022-06-10   Median :6.000   Mode  :character   Mode  :character
 Mean   :2022-06-07   Mean   :5.756
 3rd Qu.:2022-06-28   3rd Qu.:6.000
 Max.   :2022-07-11   Max.   :7.000
 Bedroom_Hall_Kitchen Number_of_Bathroom Floor_Preference Total_Floor_Numbers
 Min.   :1.000        Min.   : 1.000     Min.   :-2.000   Min.   : 1.000
 1st Qu.:2.000        1st Qu.: 1.000     1st Qu.: 1.000   1st Qu.: 2.000
 Median :2.000        Median : 2.000     Median : 2.000   Median : 4.000
 Mean   :2.084        Mean   : 1.966     Mean   : 3.435   Mean   : 6.974
 3rd Qu.:3.000        3rd Qu.: 2.000     3rd Qu.: 3.000   3rd Qu.: 6.000
 Max.   :6.000        Max.   :10.000     Max.   :76.000   Max.   :89.000
   House_Size       Rental_Fee      Rental_Fee_Status  Furnishing_Status
 Min.   :  10.0   Min.   :   1200   Length:4746        Length:4746
 1st Qu.: 550.0   1st Qu.:  10000   Class :character   Class :character
 Median : 850.0   Median :  16000   Mode  :character   Mode  :character
 Mean   : 967.5   Mean   :  34993
 3rd Qu.:1200.0   3rd Qu.:  33000
 Max.   :8000.0   Max.   :3500000
  Area_Type             City
 Length:4746        Length:4746
 Class :character   Class :character
 Mode  :character   Mode  :character
```

*Figure 4.10.1: Source Code and Output - Details of Each Column*

Function summary() is used to summarize the values in a vector, data frame, regression model or ANOVA model in R. In this case, the data set that was being stored as data frame is summarized as shown in Figure 4.10.1.

# 5.0 Data Manipulation

## 5.1. Remove Outliers

The purpose of removing outliers is to ensure there will be no biased results produced later in the analysis.

**Remove in Rental_Fee**

```
>   rf_no_outliers <- remove_outliers(house_rental_data,c('Rental_Fee'))
```

*Figure 5.1.1: Remove Outliers in Rental_Fee*

# 6.0 Data Visualization

## Question 1: What do Tenant Based on to Choose Their Houses?

By creating this question to find out how tenants choose their house, comparison among the columns is conducted.

### Analysis 1-1: Find the Count of Tenant Choose Houses Based on Date_Posted

This analysis is conducted to find out how many tenants choose their houses based on the posted date of the house.

```
# Frequency Polygon
ggplot(house_rental_data,aes(x=Date_Posted)) +
  geom_freqpoly(bins=100) +
  xlab("Date Posted") +
  ylab("Number of House Based on Posted Date") +
  ggtitle("Count of Tenant Choose House Based on Date_Posted") +
  theme_bw() +
  facet_wrap(~Tenant_Type)
```

*Figure 6.1.1: Source Code – Frequency Polygon Graph to Show Count of Tenant Choose House Based on Date_Posted*

**Analysis Technique: Data Visualization**

Figure 6.1.1 depicts the source code used to generate a frequency polygon to study the distribution of residences based on each tenant's preference. The geom_freqpoly function's bins is used to organise the data by a range of values. The xlab function modifies the a-axis label, whereas the ylab function modifies the y-axis label. The ggtitle function is used to edit the plot's main title. The function theme_bw provides a graph with a white backdrop and black gridlines. The facet_wrap method is utilised to generate graphics tables that show the same graph for each group of tenants.

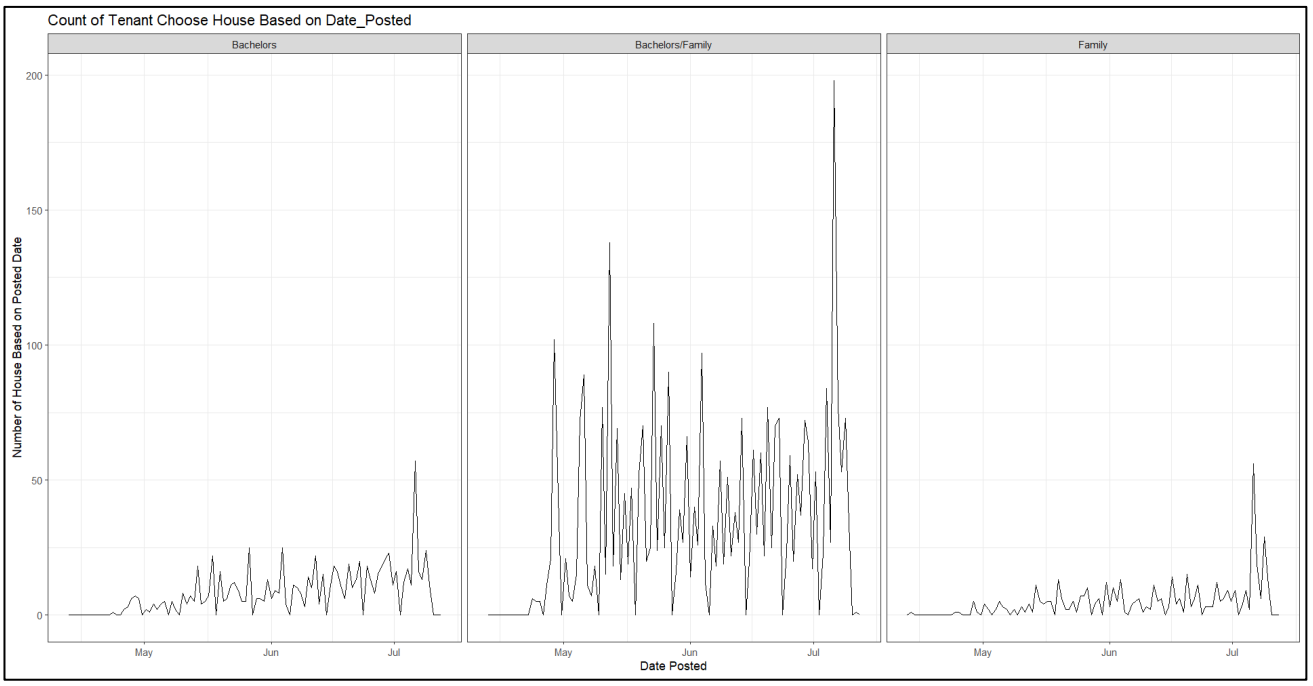



*Figure 6.1.2: Output – Frequency Polygon Graph of Date_Posted against Number of Houses*

**Explanation**

Figure 6.1.2 reveals that the maximum number of residences rented by bachelors, bachelors/family and families occurs in July, while the lowest number occurs in approximately May. It might be anticipated that July is the highest month for house rentals.

**Findings**

- More bachelors rent house in July and fewer rent house in May and June
- More bachelors or family rent house in May and in July and fewer rent house in June
- More family rent house in July and fewer rent houses in May and in June

## Analysis 1-2: Find the Distribution of Tenant Choose House Based on Point of Contact

This analysis is conducted to find out the percentage on tenant choose house based on the point of contact.

```
# Calculate Percentage Grouped by Point_of_Contact
group_poc <- house_rental_data %>% group_by(Point_of_Contact) %>%
  count() %>% ungroup() %>% mutate(perc=n/sum(n)) %>% arrange(perc) %>%
  mutate(labels=scales::percent(perc))
# Pie Chart
ggplot(group_poc,aes(x="",y=perc,fill=Point_of_Contact)) +
  geom_bar(stat="identity") +
  guides(fill=guide_legend(title="Furnishing Status")) +
  geom_text(aes(label=labels),position=position_stack(vjust=0.5)) +
  coord_polar(theta="y") +
  theme_void() +
  theme(legend.position = "bottom") +
  scale_fill_brewer() +
  ggtitle("Percentage of Tenant Choose House Based on Point of Contact")
```

*Figure 6.1.3: Source Code – Pie Chart to Show the Percentage of Tenant Choose House Based on Point_of_Contact*

**Analysis Technique: Data Visualization and Manipulation**

Figure 6.1.3 depicts the source code used to create the percentage of tenant choose house based on point of contact. The number of tenants choose house based on point of contact is calculate using **group_by()** and **count()**. Then the percentage is calculated by finding the total numbers using **sum()** and use **mutate()** to calculate the percentage and assigned it to variable “perc”. The percentage is then arranged in ascending using **arrange()**. In order to show the percentage symbol in the graph, **scales::percent()** is used and assigned it to labels.

The next source code is used to generate a pie chart to study the distribution of tenant based on point of contact. The **guides()** is used to provide guides for each scale and the **guide_legend()** is used to customize the title of the legend of discrete scale. The position with **position_stack()** in geom_text stacks the bars on top of each other and the alignment of the text can be adjusted by vjust argument with 0.5 for middle. The **coord_polar()** is used to create pie chart from a stacked bar chart and can depend on x-axis or y-axis. The **theme_void**() is used to show the absolutely plot components. Most of the less necessary metrics and components of the graph are removed. The **theme()** is used to customize the non-data components such as the position of the legend. **scale_fill_brewer()** is the function that is enabled to change the colour of the plot. Lastly, the title name is set using **ggtitle()**.

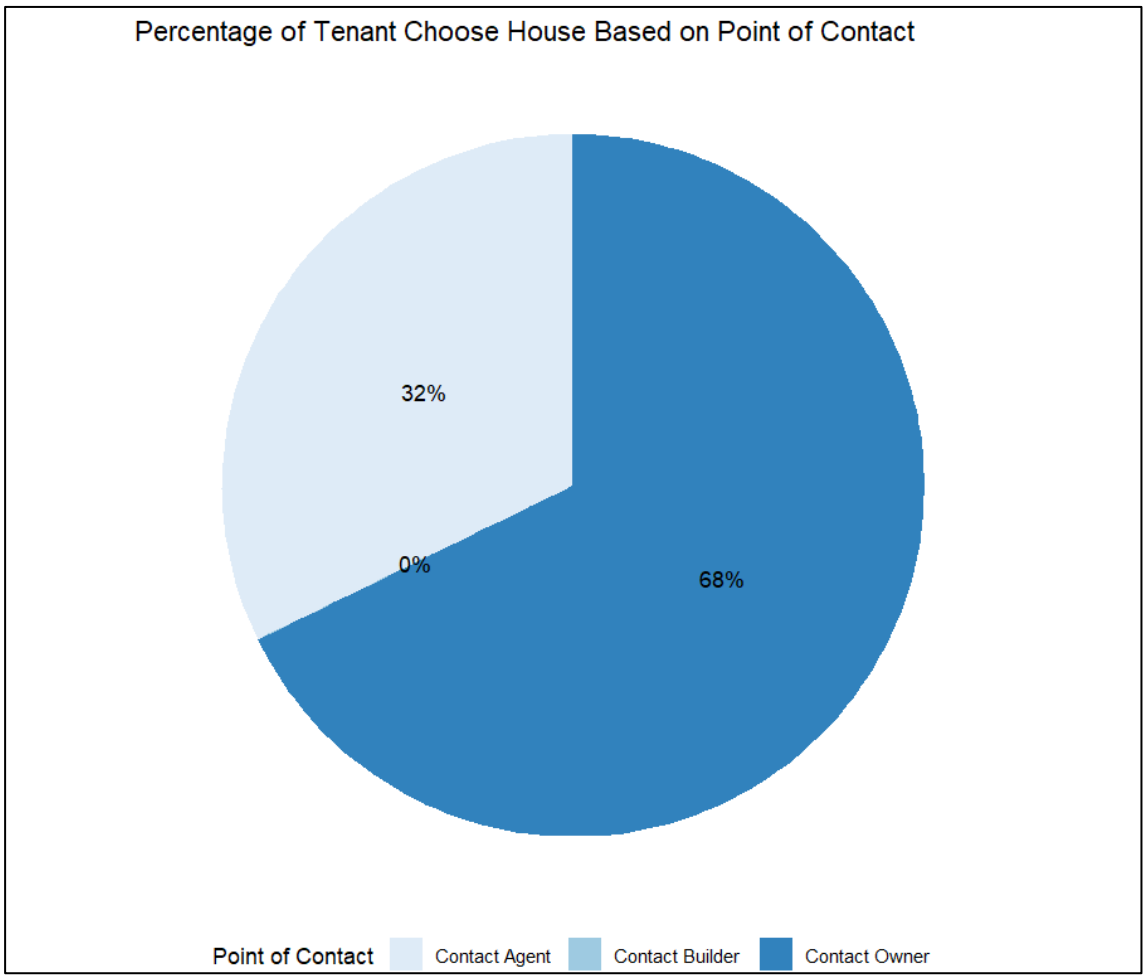



*Figure 6.1.4: Output – Pie Chart (Percentage of Tenant Choose House Based on Point_of_Contact)*

**Explanation**

Figure 6.1.4 reveals that 68% of the tenants choose their houses by contacting the owner, while 32% of them choose their houses by contacting the agent. None of them contact the builder. It might be anticipated that owner is the one who contacted by tenants the most .

**Findings**

- More tenants contact owner than agent and builder

## Analysis 1-3: Find the Percentage of Tenant Choose House Based on Number of Bedroom, Hall and Kitchen

This analysis is conducted to find the distribution of tenant prefer their houses to have how many bedrooms with one hall, and one kitchen in it.

```
ggplot(house_rental_data,aes(x=Bedroom_Hall_Kitchen)) +
  geom_bar(aes(y=after_stat(prop),fill=factor(after_stat(x))),stat="count") +
  geom_text(aes(label=scales::percent(after_stat(prop)),
                y=after_stat(prop)),stat="count",vjust=-0.5) +
  labs(x="Number of Bedroom, Hall and Kitchen",y="Percentage",
       fill="Number of Bedroom, Hall and Kitchen") +
  scale_x_continuous(breaks=seq(1,6,1)) +
  scale_y_continuous(labels=scales::percent) +
  theme(legend.position="bottom",
        legend.background = element_rect(fill="antiquewhite")) +
  ggtitle("Percentage of Tenant Choose House Based on Number of Bedroom, Hall and Kitchen")
```

*Figure 6.1.5: Source Code – Bar Chart to Show the Percentage of Tenant Choose House Based on Bedroom_Hall_Kitchen*

**Analysis Technique: Data Visualization**

Figure 6.1.5 depicts the source code used to create bar chart to show the percentage of tenants choose house based on each number of bedrooms, hall, and kitchen. The proportion of tenant choose house based on bedroom, hall and kitchen is calculated by **after_stat(prop)** and is applied to y-axis. The colour of the bar is filled using **after_stat(x)**. **scales::percent()** is used to show percentages on the graph. **scale_x_continuous()** is the position scales for continuous data of x-axis, while **scale_y_continuous()** is the position scales for continuous data of y-axis. The x-axis label is modified so that it produce 1 to 6 with 1 space between them in the graph and it is referred to breaks. The y-axis label is modified so that it shows percentage, and this is referred to labels. **theme()** is used to customize the non-data components such as the position of the legend to the bottom and the background of the legend to antiquewhite. Lastly, the title name is set using **ggtitle()**.

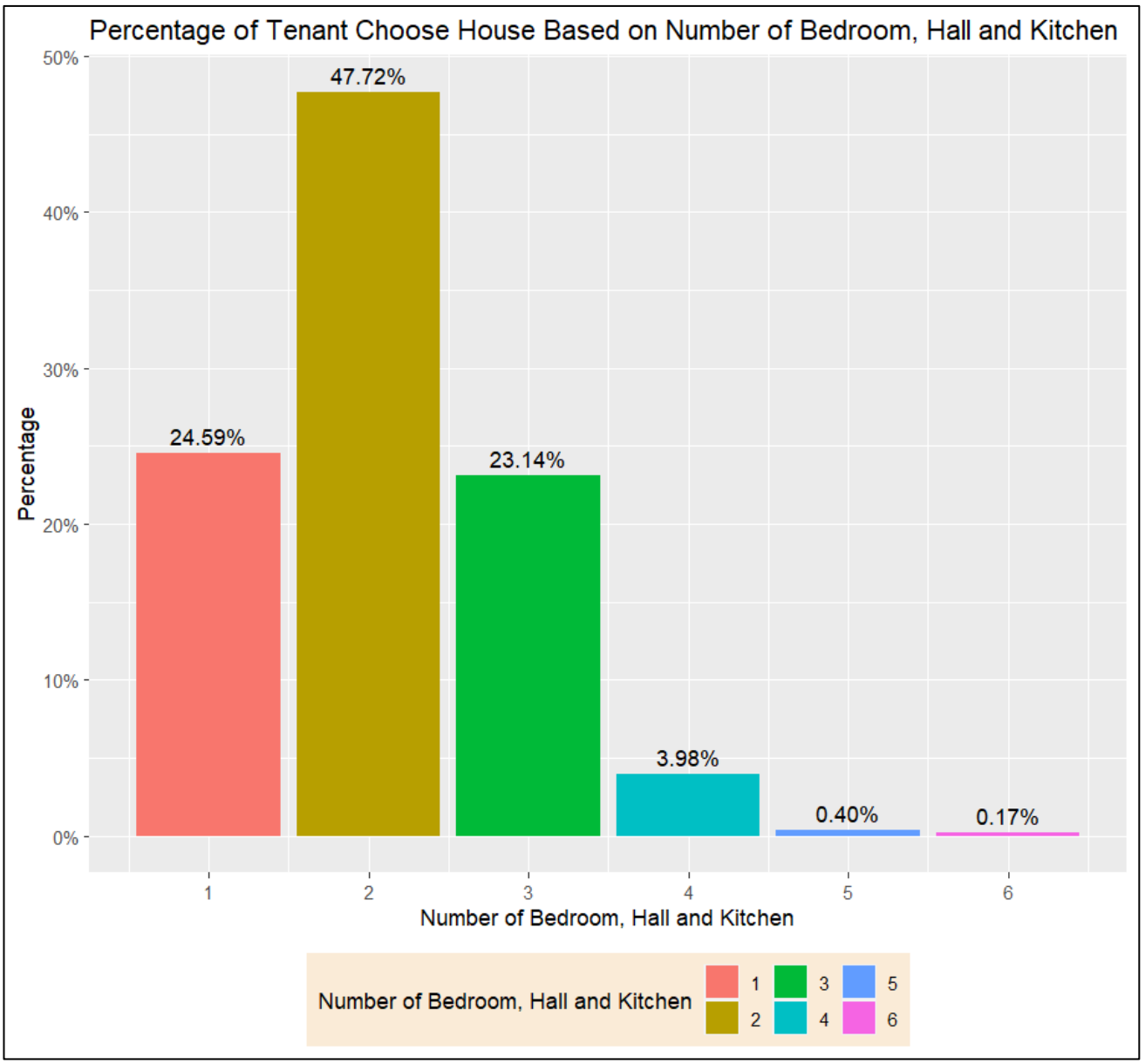



*Figure 6.1.6: Output – Bar Chart (Percentage of Tenant Choose House Based on Bedroom_Hall_Kitchen)*

**Explanation**

Figure 6.1.6 reveals that 47.72% of the tenants prefer 2 bedroom, a hall, and a kitchen, while 24.59% of them and 23.14% of them prefer houses with 1 bedroom, a hall, a kitchen and 3 Bedroom, a hall and a kitchen respectively. Fewer of them prefer having 4 or 5 or 6 bedrooms, a hall and a kitchen.

**Findings**

- More tenants prefer either 1 or 2 or 3 bedrooms, a hall, and a kitchen
- Fewer tenants prefer either 4 or 5 or 6 bedrooms, a hall, and a kitchen

## Analysis 1-4: Find the Percentage of Tenant Choose House Based on Number of Bathroom

This analysis is conducted to find the distribution of tenant prefer their houses to have how many bathrooms in it.

```
ggplot(house_rental_data,aes(x=Number_of_Bathroom)) +
  geom_bar(aes(y=after_stat(prop),fill=factor(after_stat(x))),stat="count",width=0.5) +
  geom_text(aes(label=scales::percent(after_stat(prop)),
                y=after_stat(prop)),stat="count",vjust=-0.4,size=4) +
  labs(x="Number of Bathroom",y="Percentage",fill="Number of Bathroom") +
  scale_x_continuous(breaks=seq(1,10,1)) +
  scale_y_continuous(labels=scales::percent) +
  guides(fill=guide_legend(nrow=1)) +
  theme(legend.position="bottom",
        legend.background = element_rect(fill="lightyellow")) +
  ggtitle("Percentage of Tenants Choose House Based on Number of Bathroom")
```

*Figure 6.1.7: Source Code – Bar Chart to Show the Percentage of Tenant Choose House Based on Number of Bathroom*

**Analysis Technique: Data Visualization**

Figure 6.1.7 depicts the source code used to create bar chart to show the percentage of tenants choose house based on number of bathrooms. The proportion of tenant choose house based on bathrooms is calculated by **after_stat(prop)** and is applied to y-axis. The colour of the bar is filled using **after_stat(x)**. **scales::percent()** is used to show percentages on the graph. **scale_x_continuous()** is the position scales for continuous data of x-axis, while **scale_y_continuous()** is the position scales for continuous data of y-axis. The x-axis label is modified so that it produce 1 to 10 with 1 space between them in the graph and it is referred to breaks. The y-axis label is modified so that it shows percentage, and this is referred to labels. The **guides()** is used to provide guides for each scale and the **guide_legend(nrow=1)** is used to customize the legend to show in one horizontal row. **theme()** is used to customize the non-data components such as the position of the legend to the bottom and the background of the legend to light yellow. Lastly, the title name is set using **ggtitle()**.

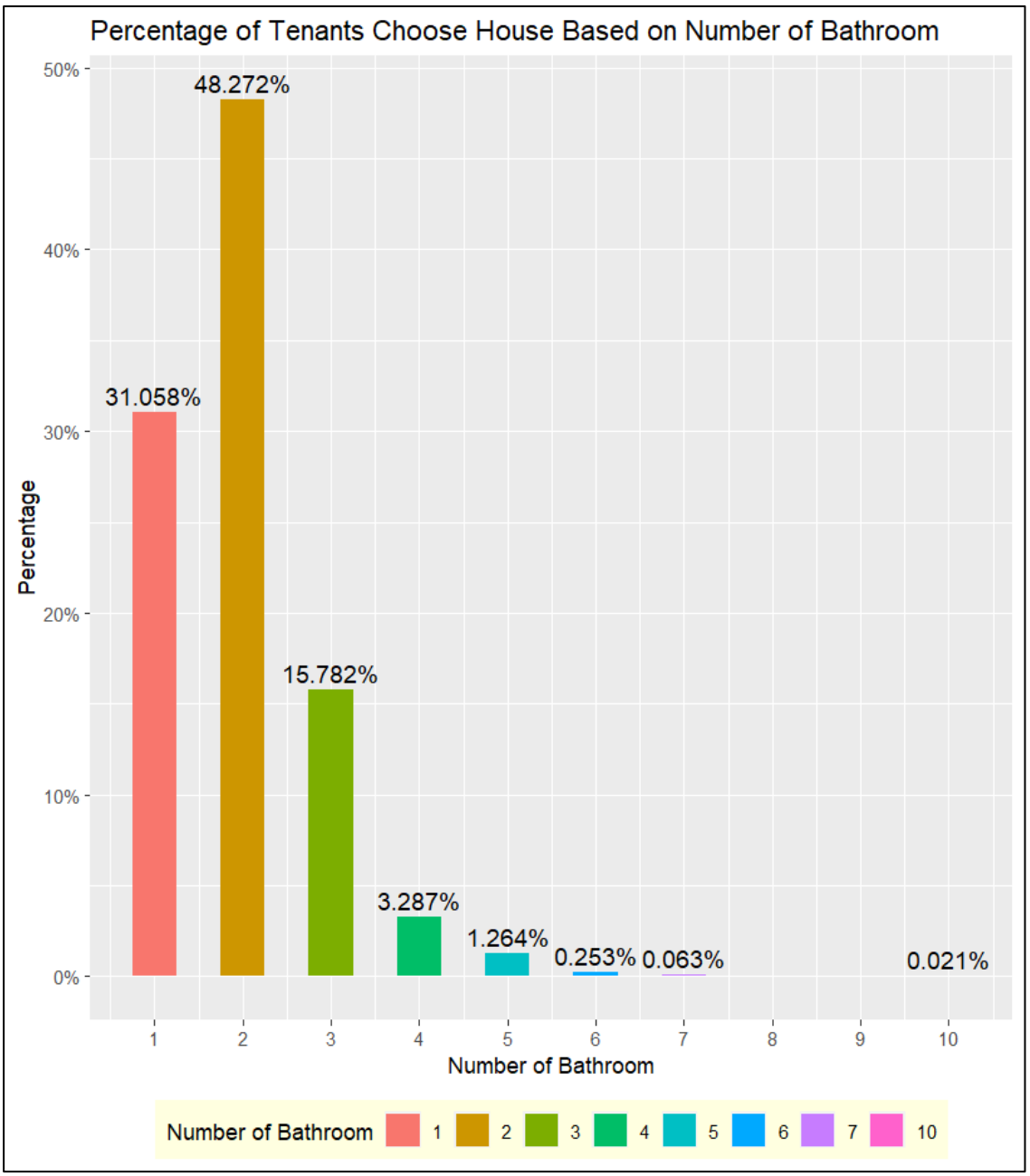



*Figure 6.1.8: Output – Bar Chart (Percentage of Tenant Choose House Based on Number_of_Bathroom)*

**Explanation**

Figure 6.1.8 reveals that almost half of the tenants (48.272%) prefer to have 2 bathrooms in their houses followed by 31.058% of them prefer to have 1 bedroom in their house. The rest of them prefer having more than 2 bathrooms in the house.

**Findings**

- More tenants prefer either 1 or 2 bathrooms
- Less than 80% of them prefer more than 2 bathrooms

## Analysis 1-5: Find the Percentage of Tenant Choose House Based on Floor Preference

```
# Calculate Percentage Grouped by Floor_Preference
group_fp <- house_rental_data %>%
  group_by(Floor_Preference) %>%
  count() %>% ungroup() %>% mutate(perc=n/sum(n)) %>%
  arrange(perc) %>% mutate(labels=scales::percent(perc))
# Bar Chart
ggplot(group_fp,aes(x=Floor_Preference,y=perc,fill=Floor_Preference,label=labels)) +
  geom_bar(stat="identity",width=0.7) +
  scale_x_continuous(breaks=seq(-2,80,5)) +
  scale_y_continuous(labels=scales::percent) +
  scale_fill_gradient(low="pink",high="purple") +
  labs(x="Floor Preference",y="Percentage",
       title="Percentage of Tenant Choose House Based on Floor Preference")
```

*Figure 6.1.9: Source Code – Bar Chart to Show the Percentage of Tenant Choose House Based on Floor_Preference*

**Analysis Technique: Data Visualization and Manipulation**

Figure 6.1.9 depicts the source code used to create the percentage of tenant choose house based on their floor preferences. The number of tenants choose house based on floor preferences is calculated using **group_by()** and **count()**. Then the percentage is calculated by finding the total numbers using **sum()** and use **mutate()** to calculate the percentage and assigned it to variable "perc". The percentage is then arranged in ascending using **arrange()**. In order to show the percentage symbol in the graph, **scales::percent()** is used and assigned it to labels.

The next source code is used to generate a bar chart to investigate the percentage of tenant based on floor preferences. **scales::percent()** is used to show percentages on the graph. **scale_x_continuous()** is the position scales for continuous data of x-axis, while **scale_y_continuous()** is the position scales for continuous data of y-axis. The x-axis label is modified so that it produce -2 to 80 with gap of 5 between them in the graph and it is referred to breaks. The y-axis label is modified so that it shows percentage, and this is referred to labels. Lastly, the main title and labels' title is modified using **labs()**.

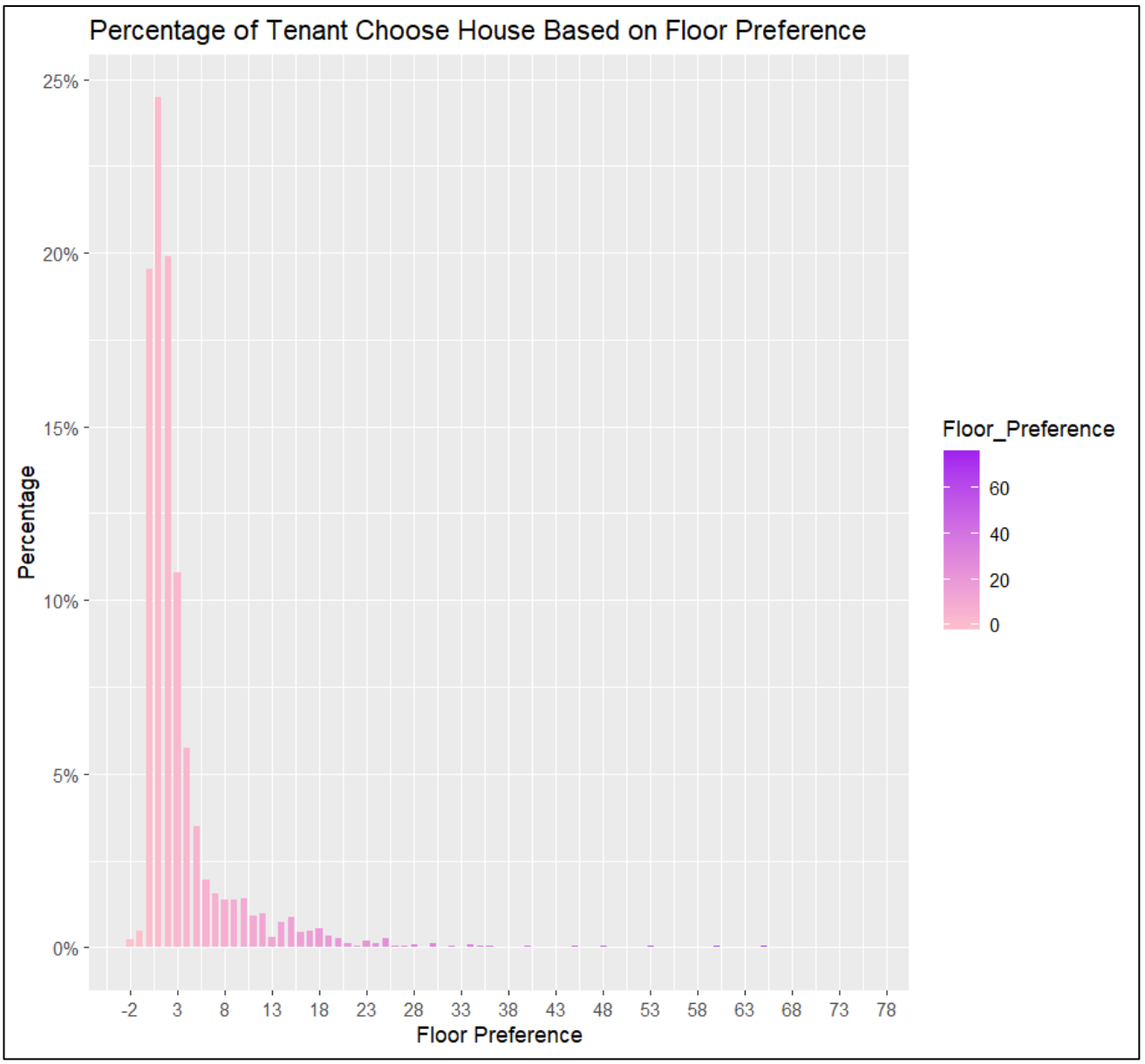



*Figure 6.1.10:Ouptut – Bar Chart (Percentage of Tenant Choose House Based on Floor_Preference)*

**Explanation**

Figure 6.1.10 reveals that around 25% of the tenants prefer to choose their house located at first floor and around 20% choose to live at ground floor and second floor, while the rest prefer to choose their house located higher.

**Findings**

- More tenants prefer either ground, first or second floor
- Fewer tenants prefer houses located between third floor and 18 floor
- Floor that is higher than 20 are less preferred by the tenants

## Analysis 1-6: Find the Percentage of Tenant Choose House Based on House Size

This analysis is conducted to find out how many tenants choose their house with smaller size and larger size in percentage.

```
# Calculate Percentage Grouped by House_Size
group_hs <- house_rental_data %>% group_by(House_Size) %>% count() %>% ungroup() %>%
  mutate(perc=n/sum(n)) %>%
  arrange(perc) %>%
  mutate(labels=scales::percent(perc))
# Scatter Plot
ggplot(hs_group,aes(x=House_Size,y=perc,label=labels)) +
  geom_point(aes(x=House_Size,y=perc)) +
  scale_x_continuous(breaks=seq(0,8000,500)) +
  scale_y_continuous(labels=scales::percent) +
  geom_text_repel(max.overlaps = 20) +
  ggtitle("Percentage of Tenant Choose House Based on Size of House") +
  labs(x="Size of House",y="Percentage")
```

*Figure 6.1.11: Source Code – Scatter Plot to Show the Percentage of Tenant Choose House Based on House_Size*

**Analysis Technique: Data Visualization and Manipulation**

Figure 6.1.11 depicts the source code used to create the percentage of tenant choose house based on the size of the house. The number of tenants choose house based on house size is calculated using **group_by()** and **count()**. Then the percentage is calculated by finding the total numbers using **sum()** and use **mutate()** to calculate the percentage and assigned it to variable "perc". The percentage is then arranged in ascending using **arrange()**. In order to show the percentage symbol in the graph, **scales::percent()** is used and assigned it to labels.

The next source code is used to generate a scatter graph to investigate the percentage of tenant choose houses based on house size. **scales::percent()** is used to show percentages on the graph. **scale_x_continuous()** is the position scales for continuous data of x-axis, while **scale_y_continuous()** is the position scales for continuous data of y-axis. The x-axis label is modified so that it produce 0 to 8000 with gap of 500 between them in the graph and it is referred to breaks. The y-axis label is modified so that it shows percentage, and this is referred to labels. **geom_text_repel()** is used to add text directly to the graph and repel overlapping text labels. Lastly, **ggtitle()** is used to set the title name of the graph and labels' title is modified using **labs()**.

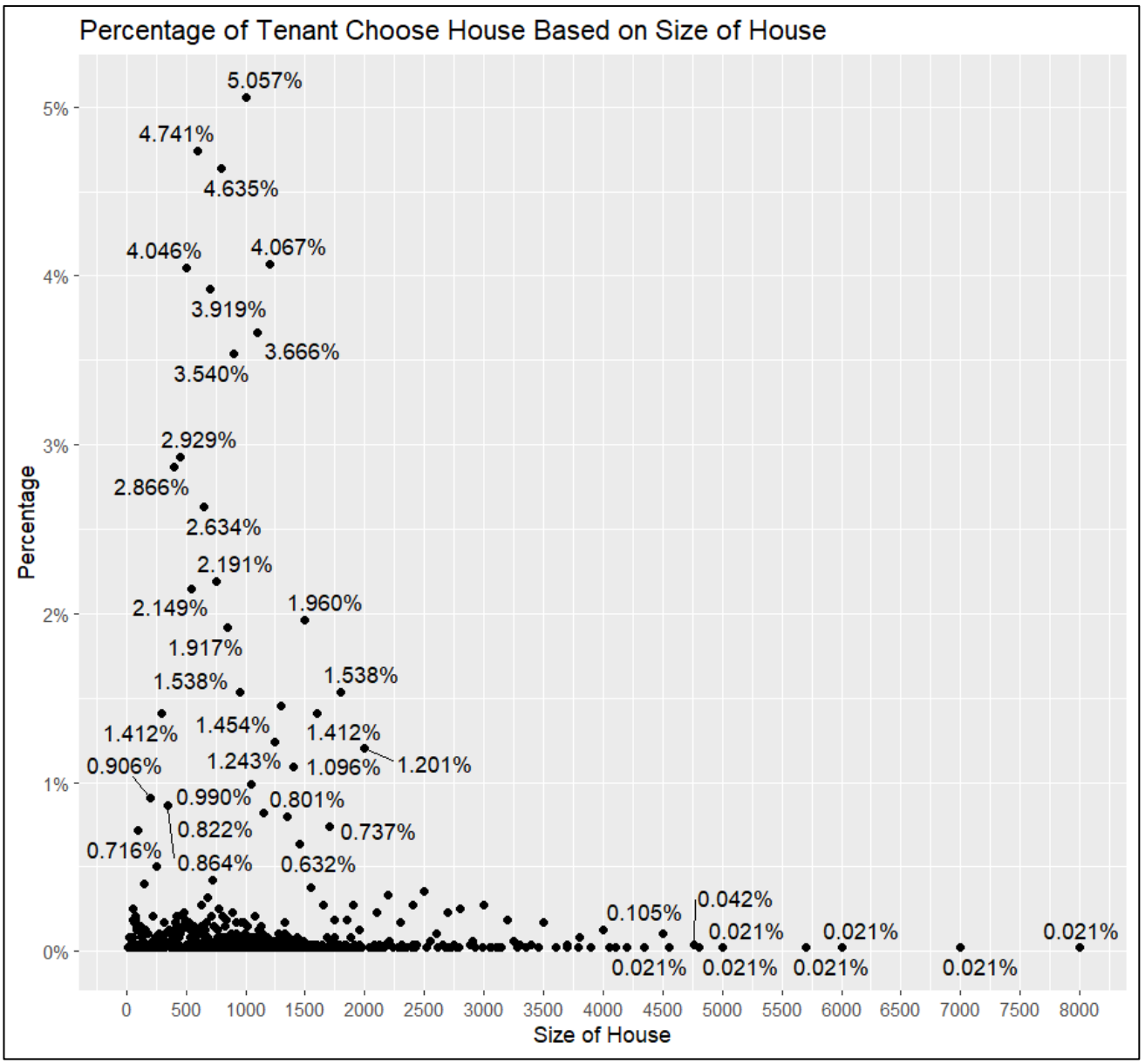



*Figure 6.1.12: Output – Scatter Plot (Percentage Choose House Based on House_Size)*

**Explanation**

Figure 6.1.12 reveals that more than 80% of the tenants prefer to live in house that is smaller than 3000 sqft, while the rest prefer to live in house that is larger than 3000 sqft.

**Findings**

- Most of the tenants prefer to choose their house that is smaller than 3000 sqft.

## Analysis 1-7: Find the Percentage of Tenant Choose Houses Based on Rental Fee

This analysis is conducted to study how many percent of the tenant choose house based on the rental fee.

```
# Calculate Percentage Grouped by Rental_Fee_Status
group_rfs <- rf_no_outliers  %>% group_by(Rental_Fee_Status) %>%
  count() %>% ungroup() %>% mutate(perc=n/sum(n)) %>%
  arrange(perc) %>% mutate(labels=scales::percent(perc))
# Pie Chart
ggplot(group_rfs, aes(x="",y=perc,fill=factor(Rental_Fee_Status))) +
  geom_bar(stat="identity") +
  geom_text(aes(x=1.8,label=labels),position=position_stack(vjust=0.5)) +
  coord_polar(theta="y") +
  theme_void() +
  ggtitle("Percentage of Tenant Choose House Based on Rental Fee") +
  scale_fill_discrete(labels=c("High = Rental Fee > Average Rental Fee",
                               "Low = Rental Fee < Average Rental Fee")) +
  guides(fill=guide_legend(title="Rental Fee Status")) +
  theme(plot.title=element_text(hjust=-0.5))
```

*Figure 6.1.13: Source Code Pie Chart to Show the Percentage of Tenant Choose Houses Based on Rental_Fee using Rental_Fee_Status*

**Analysis Technique: Data Visualization and Manipulation**

Figure 6.1.13 depicts the source code used to create the percentage of tenant choose house based on rental fee status. The number of tenants choose house based on rental fee status is calculated by grouping the rental fee status and count the numbers using **group_by()** and **count()** respectively. Then the percentage is calculated by finding the total numbers using **sum()** and divide it with the numbers. **mutate()** is used to convert the values to the percentage and assigned it to variable “perc”. The percentage is then arranged in ascending using **arrange()**. In order to show the percentage symbol in the graph, **scales::percent()** is used and assigned it to labels.

The next source code is used to generate a pie chart to study the percentage of tenant choose houses based on rental fee. The **coord_polar()** is used to create pie chart from a stacked bar chart and can depend on x-axis or y-axis. The **theme_void**() is used to show the absolutely plot components. Most of the less necessary metrics and components of the graph are removed. The title name of the plot is set using **ggtitle()**. **scale_fill_discrete()** is used to modify the legends in the way that the labels of the legends can be modified. Lastly, the **theme()** is used to customize the non-data components such as the position of the title of the plot to the middle.

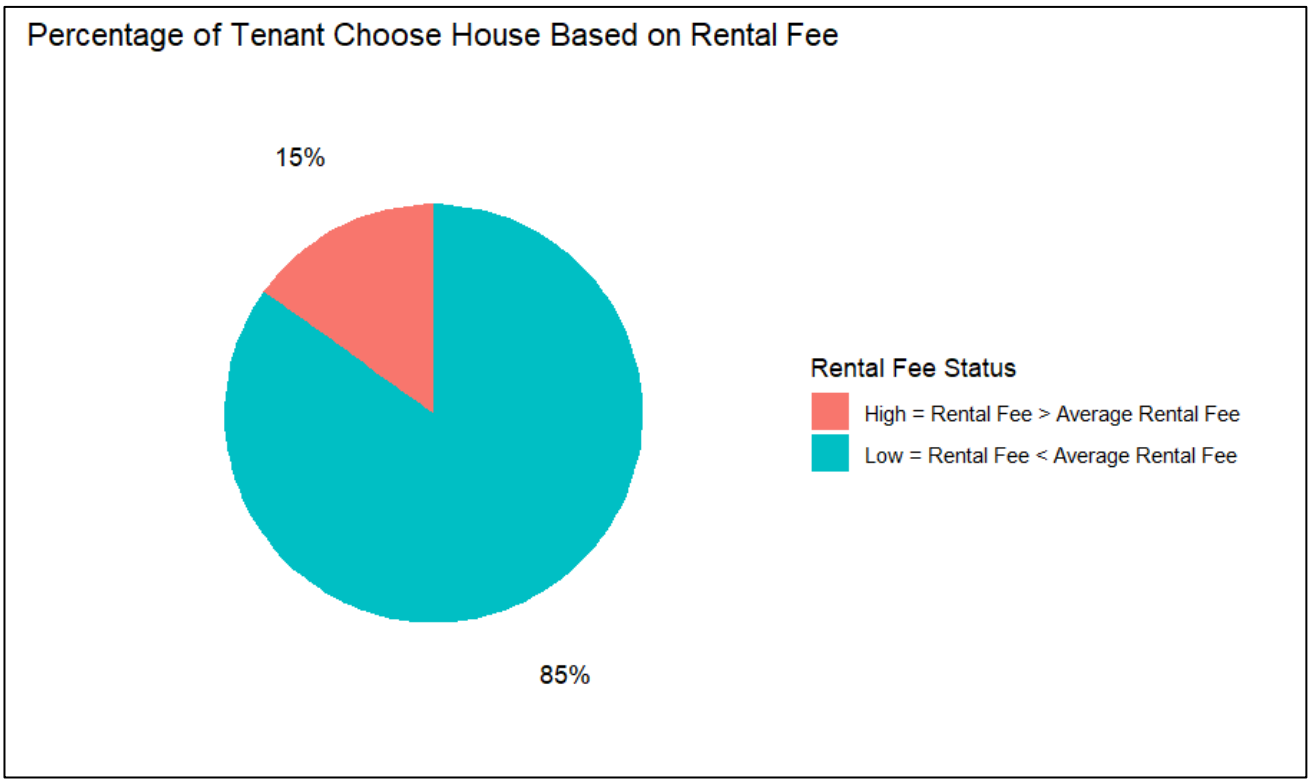



*Figure 6.1.14: Output – Pie Chart (Percentage of Tenant Choose House based on Rental_Fee using Rental_Fee_Status)*

**Explanation**

Figure 6.1.14 reveals that 85% of the tenants prefer low rental fee status which the rental fee is lower than the mean of rental fee, while the rest of them prefer high rental fee status which the rental fee is higher than the mean of rental fee.

**Findings**

- More tenants prefer to choose their house with rental fee lower than mean of rental fee

## Analysis 1-8: Find the Percentage of Tenant Choose House Based on Furnishing Status

This analysis is conducted to find out whether which status of furnishing level they preferred in their house.

```
# Calculate Percentage Grouped by Furnishing_Status
group_fs <- house_rental_data %>% group_by(Furnishing_Status) %>%
  count() %>% ungroup() %>%
  mutate(perc=n/sum(n)) %>%
  arrange(perc) %>%
  mutate(labels=scales::percent(perc))
# Pie Chart
ggplot(group_fs,aes(x="",y=perc,fill=Furnishing_Status)) +
  geom_bar(stat="identity") +
  guides(fill=guide_legend(title="Furnishing Status")) +
  geom_text(aes(label=labels),position=position_stack(vjust=0.5)) +
  coord_polar(theta="y") +
  theme_void() +
  theme(legend.position = "bottom") +
  scale_fill_brewer(palette = "PiYG") +
  ggtitle("Percentage of Tenant Choose House Based on Furnishing Status")
```

*Figure 6.1.15: Source Code – Pie Chart to Show the Percentage of Tenant Choose House Based on Furnishing_Status*

**Analysis Technique: Data Visualization and Manipulation**

Figure 6.1.15 depicts the source code used to create the percentage of tenant choose house based on the furnishing status. The number of tenants choose house based on furnishing status is calculated using **group_by()** and **count()**. Then the percentage is calculated by finding the total numbers using **sum()** and use **mutate()** to convert the values to the percentage and assigned it to variable “perc”. The percentage is then arranged in ascending using **arrange()**. In order to show the percentage symbol in the graph, **scales::percent()** is used and assigned it to labels.

The next source code is used to generate a pie chart to investigate the percentage of tenant choose houses based on furnishing status. **scales::percent()** is used to show percentages on the graph. The **guides()** is used to provide guides for each scale and the **guide_legend()** is used to customize the title of the legend of discrete scale. The **coord_polar()** is used to create pie chart from a stacked bar chart and can depend on x-axis or y-axis. The **theme_void**() is used to show the absolutely plot components. Most of the less necessary metrics and components of the graph are removed. The **theme()** is used to customize the non-data components such as the position of the legend. **scale_fill_brewer()** is the function that is enabled to change the colour of the plot. Lastly, the title name is set using **ggtitle()**.

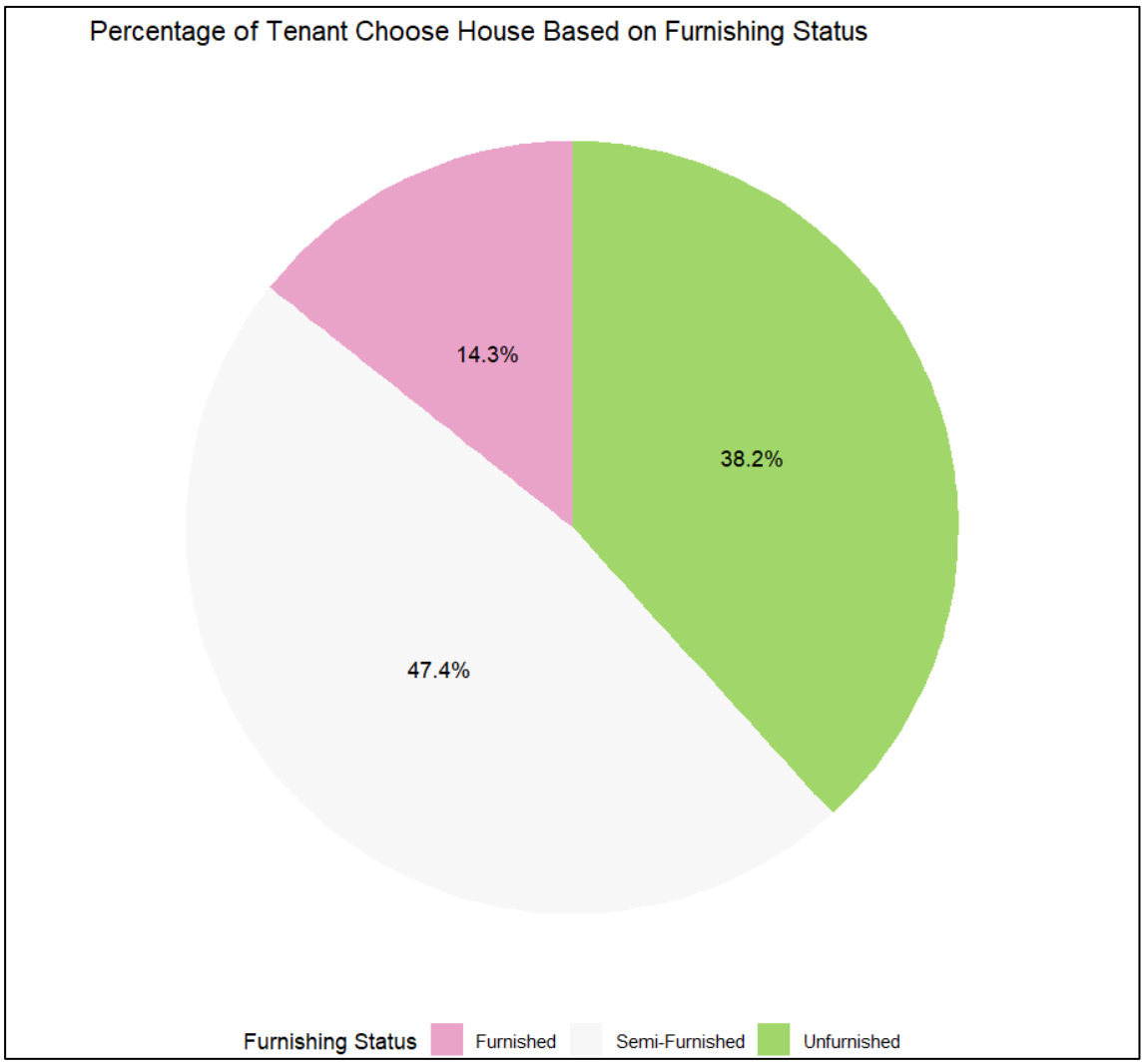



*Figure 6.1.16: Output – Pie Chart (Percentage of Tenant Choose House Based on Furnishing_Status)*

**Explanation**

Figure 6.1.16 reveals that 47.4% of the tenants choose their houses to be semi-furnished and 38.2% of them prefer their house to be unfurnished, while the rest prefer their house to be furnished.

**Findings**

- More tenants prefer unfurnished, followed by semi-furnished.
- Fewer tenants prefer furnished.

## Analysis 1-9: Find the Percentage of Tenant Choose House Based on Area Type

This analysis is conducted in order to find out how many of the tenant choose house based on the area type.

```
# Calculate Percentage Grouped by Area_Type
group_at <- house_rental_data %>% group_by(Area_Type) %>%
  count() %>% ungroup() %>% mutate(perc=n/sum(n)) %>%
  arrange(perc) %>% mutate(labels=scales::percent(perc))
# Pie Chart
ggplot(group_at,aes(x="",y=perc,fill=Area_Type)) +
  geom_bar(stat="identity") +
  guides(fill=guide_legend(title="Area Type")) +
  geom_text(aes(x=1.6,label=labels),position=position_stack(vjust=0.5)) +
  coord_polar(theta="y") +
  theme_void() +
  ggtitle("Percentage of Tenant Choose House Based on Area Type")
```

*Figure 6.1.17: Source Code – Pie Chart to Show the Percentage of Tenant Choose House Based on Area_Type*

**Analysis Technique: Data Visualization and Manipulation**

Figure 6.1.17 depicts the source code used to create the percentage of tenant choose house based on the area type. The number of tenants choose house based on area type is calculated by grouping the area type and count the numbers using **group_by()** and **count()** respectively. Then the percentage is calculated by finding the total numbers using **sum()** and divide it with the numbers. **mutate()** is used to convert the values to the percentage and assigned it to variable "perc". The percentage is then arranged in ascending using **arrange()**. In order to show the percentage symbol in the graph, **scales::percent()** is used and assigned it to labels.

The next source code is used to generate a pie chart to investigate the percentage of tenant choose houses based on furnishing status. **scales::percent()** is used to show percentages on the graph. The **guides()** is used to provide guides for each scale and the **guide_legend()** is used to customize the title of the legend of discrete scale. The **coord_polar()** is used to create pie chart from a stacked bar chart and can depend on x-axis or y-axis. The **theme_void**() is used to show the absolutely plot components. Most of the less necessary metrics and components of the graph are removed. Lastly, the title name is set using **ggtitle()**.

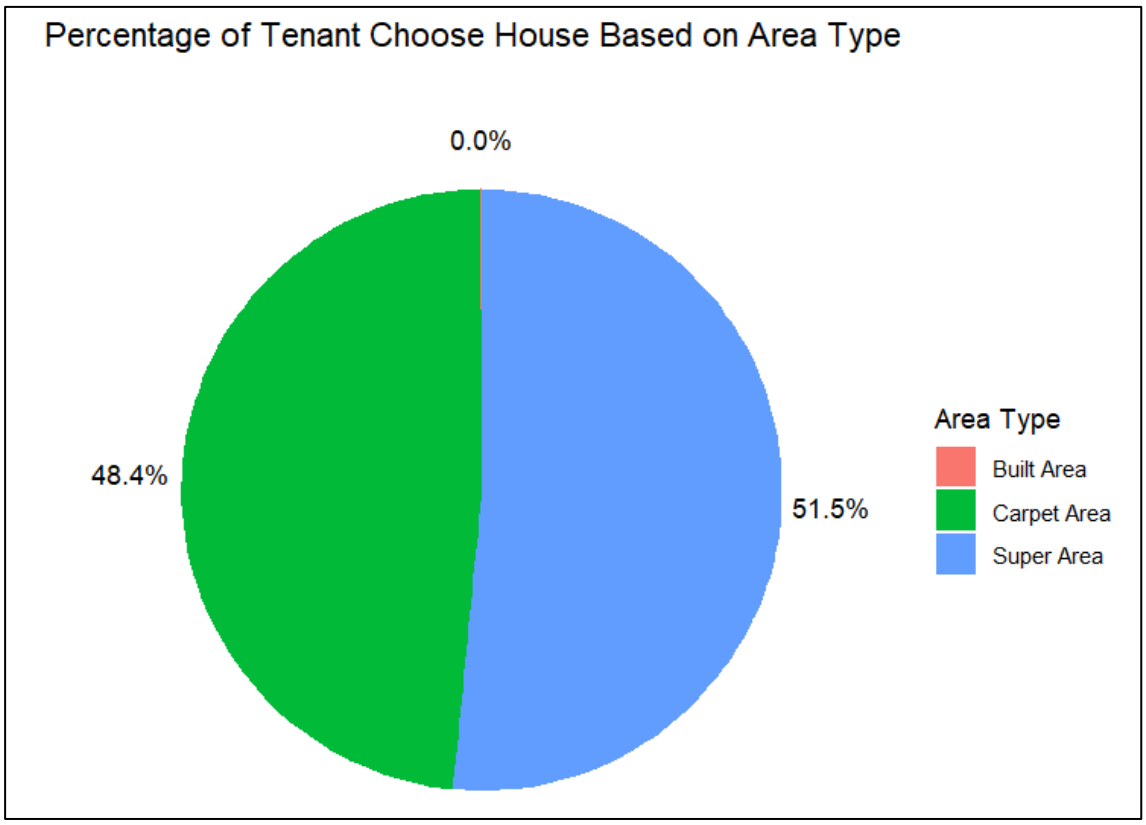



*Figure 6.1.18: Output – Pie Chart (Percentage of Tenant Choose House Based on Area_Type*

**Explanation**

Figure 6.1.18 reveals that 51.5% of the tenants prefer to choose their house which is located in super area ,whereas 48.5% of them prefer to choose their house which is located in carpet area. None of them choose houses that are located in built area

**Findings**

- Slightly more tenants choose house located in super area compared to those in carpet area.
- None of the tenants choose house located in built area.

## Analysis 1-10: Find the Percentage of Tenant Choose House Based on City

This analysis is conducted to find out how many percent of tenant choose their house based on city.

```
# Calculate Percentage Grouped by City
group_city <- house_rental_data %>%
  group_by(City) %>%
  count() %>% ungroup() %>% mutate(perc=n/sum(n)) %>%
  arrange(perc) %>% mutate(labels=scales::percent(perc))
# Bar Chart
ggplot(group_city,aes(x=City,y=perc,fill=City,label=labels)) +
  geom_bar(position="stack",stat="identity",width=0.7) +
  geom_text(aes(label=labels),vjust=-0.5) +
  scale_y_continuous(labels=scales::percent) +
  labs(x="City",y="Percentage",
       title="Percentage of Tenant Choose House Based on City")
```

*Figure 6.1.19: Source Code – Bar Chart to Show the Percentage of Tenant Choose House Based on City*

**Analysis Technique: Data Visualization and Manipulation**

Figure 6.1.19 depicts the source code used to create the percentage of tenant choose house based on the city. The number of tenants choose house based on city is calculated by grouping the area type and count the numbers using **group_by()** and **count()** respectively. Then the percentage is calculated by finding the total numbers using **sum()** and divide it with the numbers. **mutate()** is used to convert the values to the percentage and assigned it to variable “perc”. The percentage is then arranged in ascending using **arrange()**. In order to show the percentage symbol in the graph, **scales::percent()** is used and assigned it to labels.

The next source code is used to generate a pie chart to investigate the percentage of tenant choose houses based on furnishing status. **scale_y_continuous()** is the position scales for continuous data of y-axis. The y-axis label is modified so that it shows percentage using **scales::percent()**, and this is referred to labels. Lastly the main title and labels’ name are modified using **labs()**.

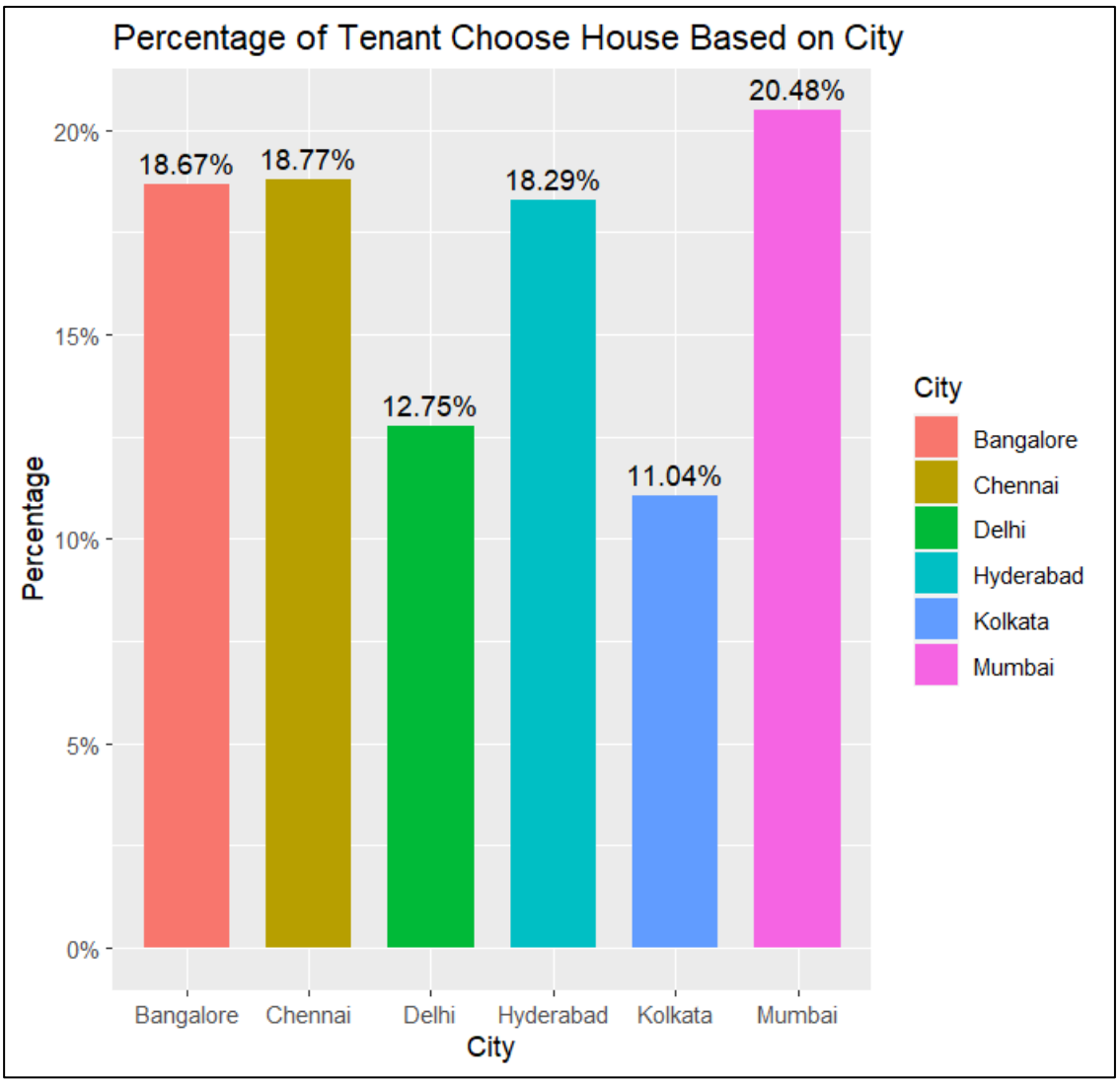



*Figure 6.1.20: Output – Bar Chart (Percentage of Tenant Choose House Based on City)*

**Explanation**

Figure 6.1.20 reveals that 20.48% of the tenants choose their houses located in Mumbai. It is followed by Chennai, Bangalore, and Hyderabad, which has 18.77%, 18.67% and 18.29% respectively. Less than 25% of them prefer their house located in Delhi and Kolkata.

**Findings**

- More tenants prefer to choose their house in Mumbai
- Some of them prefer to choose their house in Chennai, Bangalore, and Hyderabad
- Fewer of them prefer to choose Delhi and Kolkata for the location of their house.

## Conclusion for Question 1

1. Tenants prefer to rent house in May and July rather than in June.
2. Tenants prefer to rent house by contacting owner rather than contacting agent and builder.
3. Tenants prefer to rent house with either 1 or 2 or 3 bedrooms, hall, and kitchen.
4. Tenants prefer to rent house with either 1 or 2 bathrooms.
5. Tenants prefer to rent house which located lower than 20 floor level.
6. Tenants prefer to rent house which is smaller than 3000 sqft.
7. Tenants prefer to rent house which has lower rental fee than the mean rental fee
8. Tenants prefer to rent house which is either unfurnished or semi-furnished.
9. Tenants prefer to rent house located in both super area and carpet area
10. Tenants prefer to rent house in Mumbai, Chennai, Bangalore and Hyderabad.

## Question 2: What are the Factors influencing Tenants to Choose Their Houses with respect to Date_Posted/Month?

/**

** Following source code obtained from (Schork, 2021)*

*/

```
# Convert Numeric to month names
house_rental_data$Month <- month.abb[house_rental_data$Month]
```

*Figure 6.2.1: Source Code – Converting Numeric Month to Month Abbreviation*

### Analysis 2-1: Find the Relationship between Date_Posted and Number of Bedroom_Hall_Kitchen

This analysis is conducted to investigate which period and the number of bedrooms, hall and kitchen is the most preferred by bachelors, bachelors/family and family.

```
# Bar Chart
ggplot(house_rental_data,aes(x=Bedroom_Hall_Kitchen,fill=Month)) +
  geom_bar(aes(y=after_stat(count)),stat = "count") +
  labs(x="Number of Bedroom, Hall and Kitchen",y="Count",
       title="Relationship between Month and Number of Bedroom_Hall_Kitchen") +
  geom_text(aes(label=after_stat(count),
                y=after_stat(count)),stat="count",vjust=-0.4,size=4) +
  scale_fill_discrete(breaks=c('Apr','May','Jun','Jul')) +
  ylim(0,800) +
  theme_bw() +
  theme(legend.position = "bottom") +
  facet_grid(factor(Month,levels=c('Apr','May','Jun','Jul'))~Tenant_Type)
```

*Figure 6.2.2: Source Code – Bar Chart to Show the Relationship between Date_Posted and Number of Bedroom_Hall_Kitchen*

**Analysis Technique: Data Visualization**

Figure 6.2.2 depicts the source code used to create bar chart to show the percentage of tenants choose house based on the posted date and number of bedrooms, hall and kitchen. The number of tenants choose house based on this relationship is calculated by **after_stat(count)** and is applied to y-axis. **geom_text()** is used to add the text labels to the graph. **scale_fill_discrete(breaks =)** is used to rearrange to legends. **ylim()** is used to specify the lower limit and upper limit of the scale and in the graph it show the range between 0 and 800. **theme_bw()** is a function that provide white background and black guidelines to the plot. **theme()** is used to control the non-data components such as the position of the legend to the bottom. **facet grid()** produces a 2d grid of panels defined by Month and Tenant_Type which form the rows and columns.

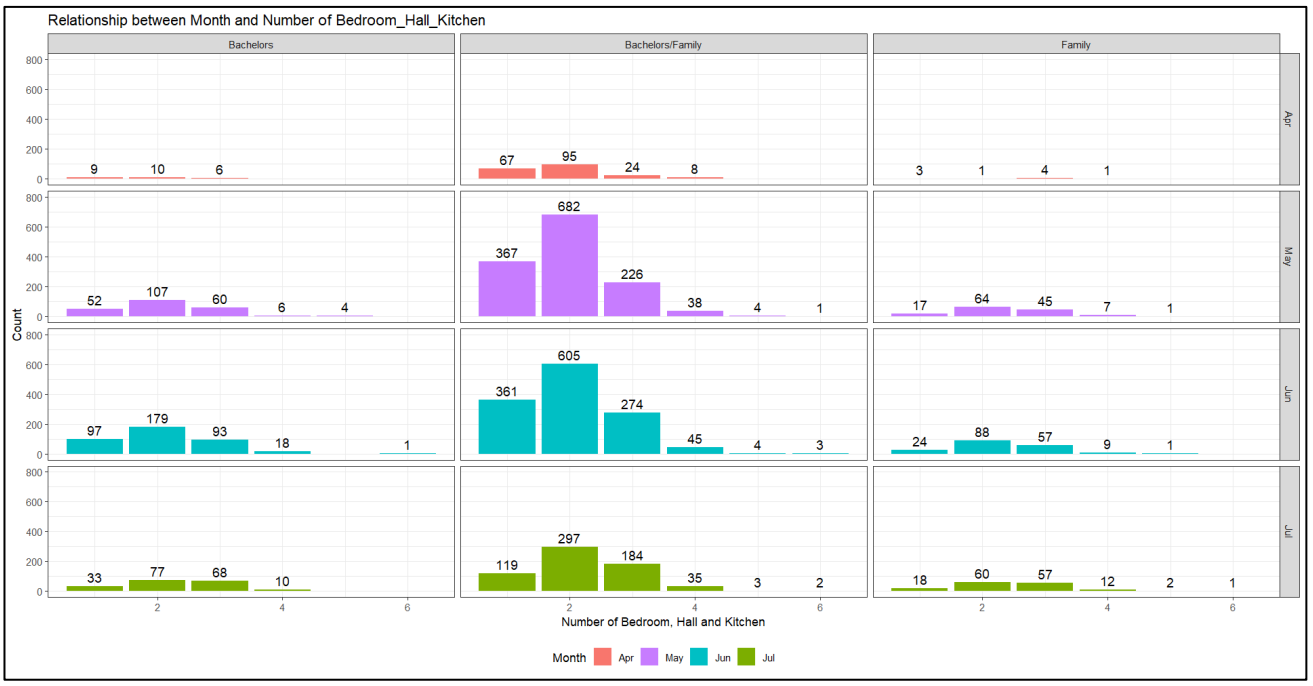



*Figure 6.2.3: Output – Bar Chart (Relationship between Date_Posted and Number of Bedroom_Hall_Kitchen)*

**Explanation**

Figure 6.2.3 reveals that 179 of the bachelors prefer to choose their house in June with 2 bedrooms, hall, and kitchen. 682 of the bachelors/family prefer to choose their house in May with 2 bedrooms, hall, and kitchen and 605 of them choose houses in June with 2 bedrooms, hall and kitchen as well. For family site, 88 of them prefer choosing house in June with 2 bedrooms, hall, and kitchen and 64 of them prefer in May with 2 bedrooms, hall, and kitchen.

**Findings**

- Bachelors and family prefer to choose 2 bedrooms, hall, and kitchen in June
- Bachelors/family prefer to choose 2 bedrooms, hall and kitchen in June and July

## Analysis 2-2: Find the Relationship between Date_Posted and Number of Bathroom

This analysis is conducted to investigate which period and the number of bathroom is the most preferred by bachelors, bachelors/family and family.

```
# Bar Chart
ggplot(house_rental_data,aes(x=Number_of_Bathroom,fill=Month)) +
  geom_bar(aes(y=after_stat(count)),stat = "count") +
  labs(x="Number of Bathroom",y="Count",
       title="Relationship between Month and Number_of_Bathroom") +
  geom_text(aes(label=after_stat(count),
                y=after_stat(count)),stat="count",vjust=-0.4,size=4) +
  scale_fill_discrete(breaks=c('Apr','May','Jun','Jul')) +
  ylim(0,800) +
  theme_bw() +
  theme(legend.position = "bottom") +
  facet_grid(factor(Month,levels=c('Apr','May','Jun','Jul'))~Tenant_Type)
```

*Figure 6.2.4: Source Code – Bar Chart to Show the Relationship between Date_Posted and Number_of_Bathroom*

**Analysis Technique: Data Visualization**

Figure 6.2.4 depicts the source code used to create bar chart to show the percentage of tenants choose house based on the posted date and number of bathrooms. The number of tenants choose house based on this relationship is calculated by **after_stat(count)** and is applied to y-axis. **geom_text()** is used to add the text labels to the graph. **scale_fill_discrete(breaks =)** is used to rearrange the order of legends. **ylim()** is used to specify the lower limit and upper limit of the scale and in the graph, it show the range between 0 and 800. **theme_bw()** is a function that provide white background and black guidelines to the plot. **theme()** is used to control the non-data components such as the position of the legend to the bottom. **facet grid()** produces a 2d grid of panels defined by Month and Tenant_Type which form the rows and columns.

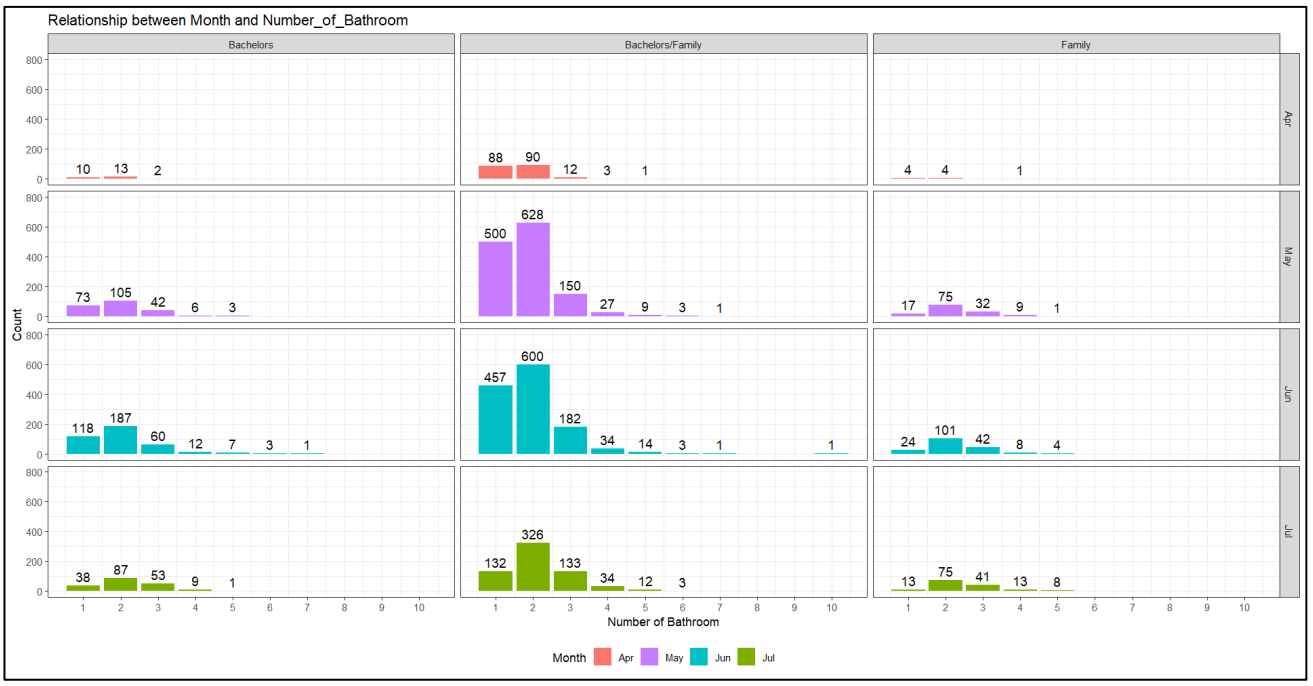



*Figure 6.2.5: Output – Bar Chart (Relationship between Date_Posted and Number_of_Bathroom)*

**Explanation**

Figure 6.2.5 reveals that 187 of the bachelors prefer to choose their house in June with 2 bathrooms and 105 of them prefer to choose their house in May with 2 bathrooms whereas 87 of them prefer to choose in July with 2 bathrooms as well.

628 of the bachelors/family prefer to choose their house in May with 2 bathrooms, followed by 600 of them prefer house with 1 bathroom whereas 600 of them prefer to choose house in June with 2 bathrooms, followed by 457 of them preferring house with 1 bathroom. In July, 326 of them prefer to have 2 bathrooms and 132 of them prefer to have 1 bathroom in their house respectively.

101 of the family prefer to choose their house in June with 2 bathrooms followed by 75 of them prefer to choose their house in May and July with 2 bathrooms respectively.

**Findings**

- More tenants choose houses with 2 bathrooms in May, June and July.
- Fewer tenants choose houses with bathrooms in April.
- Fewer tenants choose houses with more than 3 bathrooms in April, May, June and July.

## Analysis 2-3: Find the Relationship between Month and Floor_Preference

This analysis is conducted to investigate which period and which floor is the most preferred by bachelors, bachelors/family and family.

```
# Bar Chart
ggplot(house_rental_data,aes(x=Floor_Preference,fill=Month)) +
  geom_bar(stat = "count") +
  labs(x="Floor Preference",y="Count",
       title="Relationship between Month and Floor_Preference") +
  scale_fill_discrete(breaks=c('Apr','May','Jun','Jul')) +
  theme_bw() +
  theme(legend.position = "bottom") +
  facet_grid(factor(Month,levels=c('Apr','May','Jun','Jul'))~Tenant_Type)
```

*Figure 6.2.6: Source Code – Bar Chart to Show the Relationship between Month and Floor_Preference*

**Analysis Technique: Data Visualization**

Figure 6.2.6 depicts the source code used to create bar chart to show the number of tenants choose house based on month and the floor preference. The number of tenants choose house based on this relationship is calculated by **after_stat(count)** and is applied to y-axis. **labs()** is used to modify the axes label and the title of the plot. **theme_bw()** is a function that provide white background and black guidelines to the plot. **scale_fill_discrete()** is used to modify the legends in the way that the labels of the legends can be rearranged. **facet_grid()** generates a 2D grid of panels with the rows and columns given by Month and Tenant_Type.

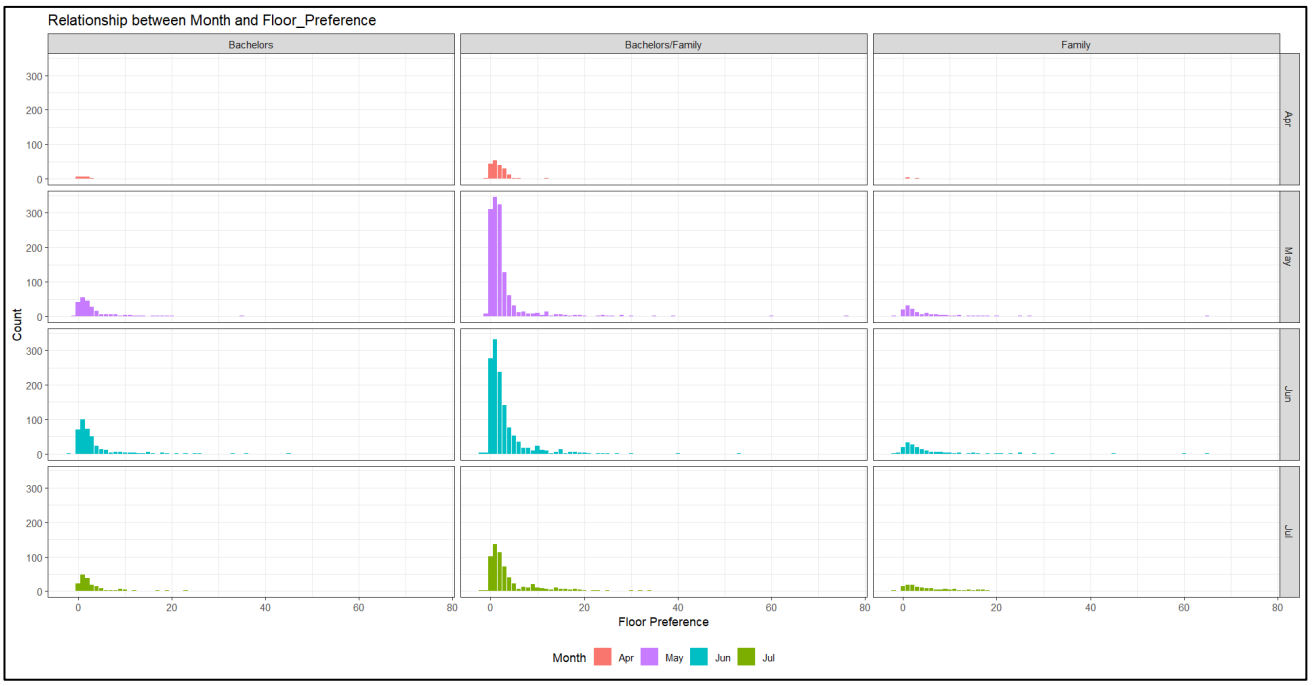



*Figure 6.2.7: Output – Bar Chart (Relationship between Month and Floor_Preference)*

**Explanation**

Figure 6.2.7 reveals that around 100 of the bachelors prefer to choose house located at the first floor in June, approximately 50 of them prefer to choose house located at the first floor in May and July.

Approximately 400 of the bachelors/family prefer to choose their house located at first floor followed by ground floor with approximately 350 of them in May whereas in June, around 375 of them prefer to choose houses located at first floor followed by ground floor with approximately 275 of them.

It can be seen that family do not have significant high peak, but almost all of them prefer to live floor level lower than 20 in May, June, and July.

**Findings**

- Bachelors prefer to rent their houses with either 1 or 2 or 3 bathrooms across May, June and July.
- Bachelors/family prefer to rent their houses more with either 1 or 2 bathrooms and some prefer with 3 bathrooms across May, June, and July.
- Family prefer to rent their houses with either 2 or 3 bedrooms across May, June, and July

## Analysis 2-4: Find the Relationship between Month and House_Size

This analysis is conducted to find out which period and what size of house is the most preferred by bachelors, bachelors/family and family.

```
# Bar Chart
ggplot(house_rental_data,
       aes(x=Month,y=House_Size,fill=Month)) +
  geom_bar(aes(factor(Month,levels=c('Apr','May','Jun','Jul')),House_Size),
           position="dodge",stat="summary",fun="mean") +
  stat_summary(aes(label=round(after_stat(y),2)),fun="mean",geom="text",vjust=-0.5,size=3) +
  labs(x="Month",y="House Size",
       title="Relationship between Month and House_Size") +
  scale_fill_discrete(breaks=c('Apr','May','Jun','Jul')) +
  scale_x_discrete(breaks=c('Apr','May','Jun','Jul')) +
  theme_bw() +
  theme(legend.position = "bottom") +
  facet_wrap(~Tenant_Type)
```

*Figure 6.2.8: Source Code – Line Graph to Show the Relationship between Date_Posted and House_Size*

**Analysis Technique: Data Visualization**

Figure 6.2.8 depicts the source code used to show the number of tenants choose house based on month and house size. The number of tenants choose house based on this relationship is calculated by find the mean using stat = "summary" and fun = "mean". **labs()** is used to modify the name of the axes and title. **scale_fill_discrete(breaks =)** is used to rearrange the order of legends. **scale_x_discrete()** is used to rearrange the order of x-axis label. **theme_bw()** is a function that provide white background and black guidelines to the plot. **theme()** is used to control the non-data components such as the position of the legend to the bottom. **facet_wrap()** is utilised to generate graphics tables that show the same graph for each group of tenants.

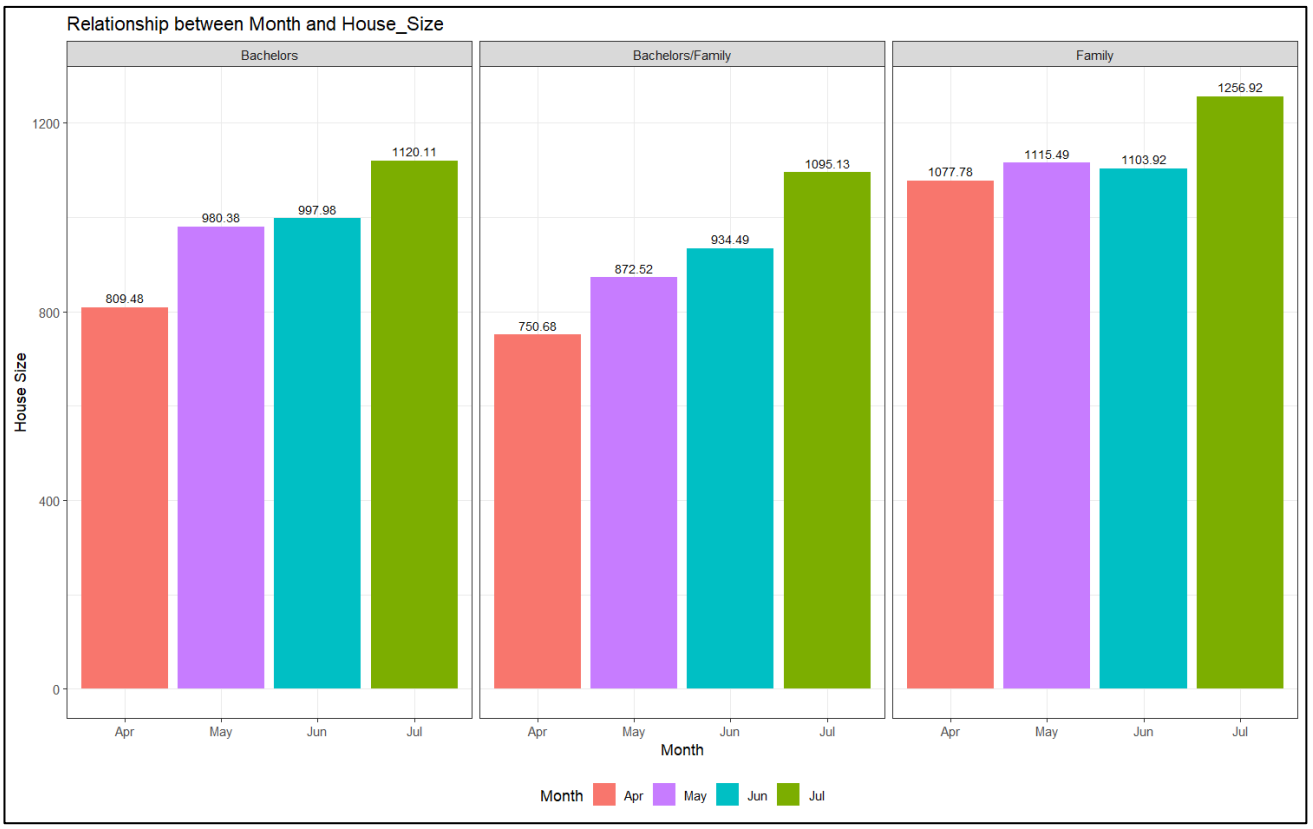



*Figure 6.2.9: Output – Line Graph (Relationship between Month and Average House_Size)*

**Explanation**

Figure 6.2.9 reveals that bachelors prefer to choose house size of about 1120 sqft in July followed by 997.98 sqft in June, 960.38sqft in May and 609.48sqft in April.

Bachelors/family also prefer to choose house in July with the house size of 1095.13 sqft, followed by 934.39 sqft in June, 873.52 sqft in May and 750.68sqft in April.

Family also prefer to choose house in July with the house size of 1256.92sqft, followed by 1115.49sqft in May, 1103.92sqft in June and 1077.78sqft in April.

**Findings**

- Bachelors prefer choosing house size greater than 1000 sqft in July
- Bachelors/family prefer choosing house size greater than 1000sqft in July
- Family prefer choosing house size greater than 1000 sqft in April, May, June, July

## Analysis 2-5: Find the Relationship between Month and Rental_Fee

This analysis is conducted to find out which period and the rental fee is the most preferred by bachelors, bachelors/family and family.

```
# Bar Chart
ggplot(house_rental_data,aes(x=Month,y=Rental_Fee,fill=Month)) +
  geom_bar(aes(factor(Month,levels=c('Apr','May','Jun','Jul')),Rental_Fee),
           position="dodge",stat="summary",fun="mean") +
  stat_summary(aes(label=round(after_stat(y),2)),fun="mean",geom="text",vjust=-0.5,size=3) +
  labs(x="Month",y="Average Monthly Rental Fee (RM)",
       title="Relationship between Month and Rental_Fee") +
  scale_y_continuous(labels=scales::comma) +
  theme_bw() +
  scale_fill_manual(values=c('#f6e8c3','#5ab4ac','#c7eae5','#d8b365')) +
  facet_wrap(~Tenant_Type)
```

*Figure 6.2.10: Source Code – Bar Chart to show the Relationship between Month and Rental_Fee*

**Analysis Technique: Data Visualization and Manipulation (Remove Outliers in Rental_Fee)**

Figure 6.2.10 depicts the source code used to show the number of tenants choose house based on month and rental fee. The number of tenants choose house based on this relationship is calculated by find the mean using stat = "summary" and fun = "mean". **stat_summary()** can be used to add mean points to a bar chart. **geom="text"** is entered so that to add the mean labels into the bar chart. **labs()** is used to modify the name of the axes and title. **theme_bw()** is a function that provide white background and black guidelines to the plot. **theme()** is used to control the non-data components such as the position of the legend to the bottom. **facet_wrap()** is utilised to generate graphics tables that show the same graph for each group of tenants.

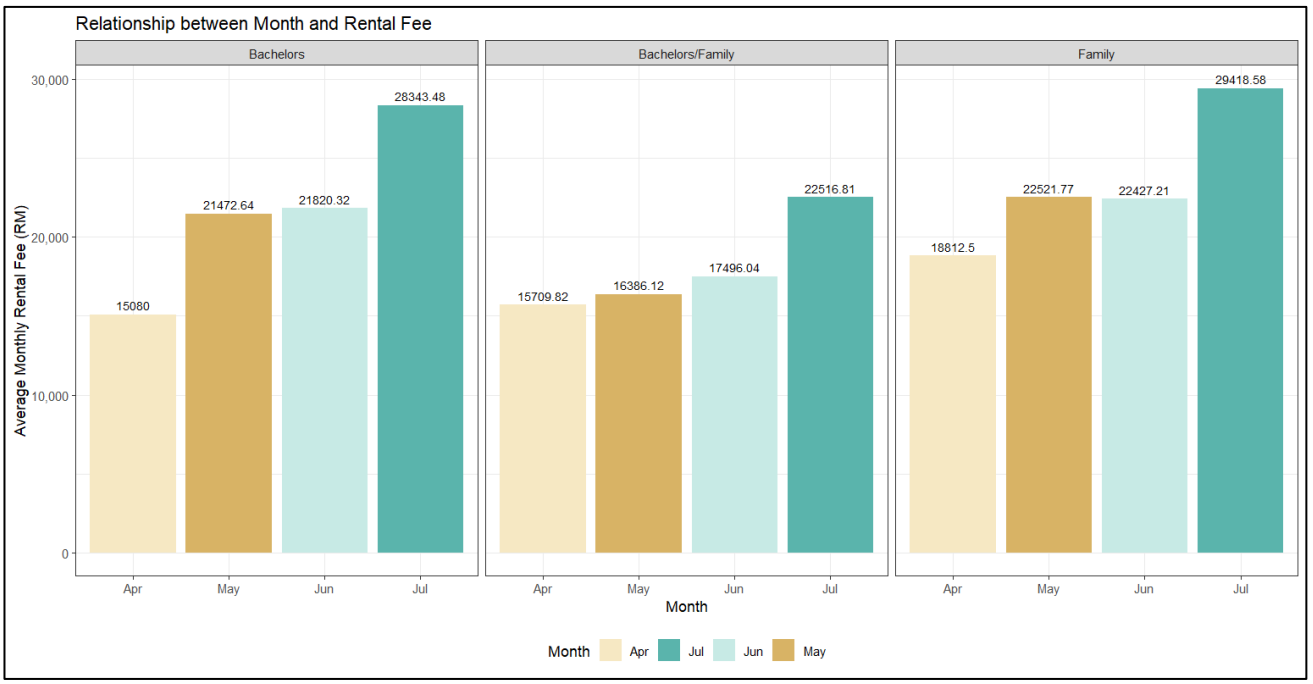



*Figure 6.2.11: Output – Bar Chart (Relationship between Month and Rental_Fee)*

**Explanation**

Figure 6.2.11 reveals that in all bachelors, bachelors/family and family, the rental fee has an increase trend. From April to July, there is an average increase of RM11,263.48 in bachelors' site whereas in bachelors/family site, there is an average increase of RM6,806.99 and in family site, there is an average increase of RM10,606,08.

**Findings**

- Average rental fee increased per month.
- Most of the tenants prefer choosing house in July with higher rental fee.

## Analysis 2-6: Find the Relationship between Date_Posted and Furnishing_Status

This analysis is conducted to find out which period and the furnishing status is the most preferred by bachelors, bachelors/family and family.

```
# Bar Chart
ggplot(house_rental_data,aes(x=Date_Posted)) +
  geom_bar(aes(y=after_stat(count)),stat="bin",binwidth=0.5) +
  geom_text(aes(label=after_stat(count)),stat="count",vjust=-0.5,size=2) +
  labs(x="Date Posted",y="Count",
       title="Relationship between Date_Posted and Furnishing_Status") +
  scale_x_date(breaks=date_breaks("2 day"),labels=date_format("%d %b")) +
  theme_bw() +
  theme(axis.line=element_line(),axis.text.x=element_text(angle=90)) +
  facet_grid(Furnishing_Status~Tenant_Type,scales="free")
```

*Figure 6.2.12: Source Code – Bar Chart to Show the Relationship between Date_Posted and Furnishing_Status*

**Analysis Technique: Data Visualization**

Figure 6.2.12 depicts the source code used to create bar chart to show the percentage of tenants choose house based on the posted date and the furnishing status. The number of tenants choose house based on this relationship is calculated by **after_stat(count)** and is applied to y-axis. **geom_text()** is used to add the text labels to the graph. **scale_x_date()** can be used to format the date to show only day and month on the axis label and set breaks to 2 day. **theme_bw()** is a function that provide white background and black guidelines to the plot. **theme()** is used to control the non-data components such as the direction of the text to 90 degrees. **facet grid()** produces a 2d grid of panels defined by Furnishing_Status and Tenant_Type which form the rows and columns.

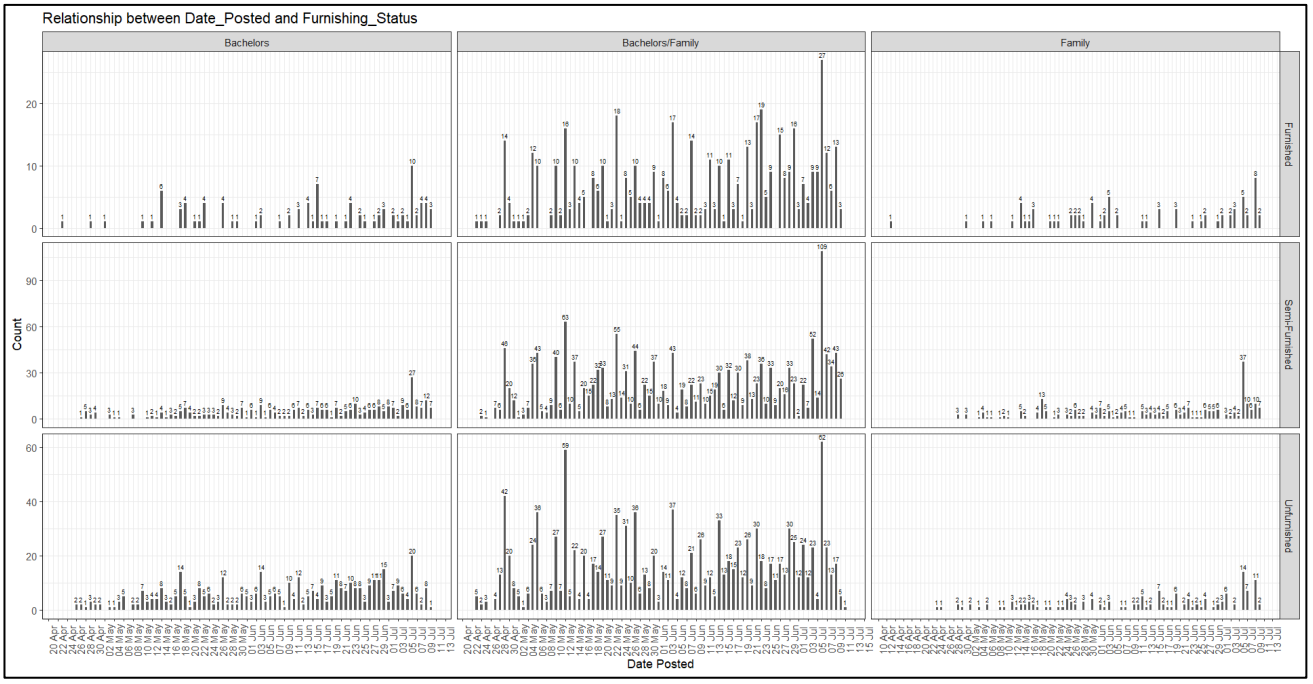



*Figure 6.2.13: Output – Bar Chart (Relationship between Date_Posted and Furnishing_Status*

**Explanation**

Figure 6.2.13 reveals that 27 of the bachelors prefer to choose their house on 4th July with semi-furnished followed by 10 and 20 of them with furnished and unfurnished respectively.

109 of the bachelors/family prefer to choose their house on 4th July with semi-furnished, followed by 27 and 62 of them prefer house with furnished and unfurnished respectively whereas 59 and 63 of them prefer to choose their house on 10th May respectively. For all furnishing status in bachelors/family site, there is a fluctuation that the count goes up and down every 2 weeks.

37 of the family prefer to choose their house on 4th July with semi-furnished followed by 8 and 14 of them prefer to choose their house on the same date with furnished and unfurnished respectively.

**Findings**

- More tenants choose houses with semi-furnished in July.
- Fewer tenants choose houses with furnishing status in April.
- Fewer tenants choose houses with furnished and unfurnished in April, May, June

## Analysis 2-7: Find the Relationship between Date_Posted and Area_Type

This analysis is conducted to find out which period and which area type is the most preferred by bachelors, bachelors/family and family.

```
# Bar Chart
ggplot(house_rental_data,aes(x=Date_Posted)) +
  geom_bar(aes(y=after_stat(count)),stat="bin",binwidth=0.5) +
  geom_text(aes(label=after_stat(count)),stat="count",vjust=-0.5,size=2) +
  labs(x="Date Posted",y="Count",
       title="Relationship between Date_Posted and Area_Type") +
  scale_x_date(breaks=date_breaks("2 day"),labels=date_format("%d %b")) +
  theme_bw() +
  theme(axis.line=element_line(),axis.text.x=element_text(angle=90)) +
  facet_grid(Area_Type~Tenant_Type,scales="free")
```

*Figure 6.2.14: Source Code – Bar Chart to Show the Relationship between Date_Posted and Area_Type*

**Analysis Technique: Data Visualization**

Figure 6.2.14 depicts the source code used to create bar chart to show the percentage of tenants choose house based on the posted date and the area type. The number of tenants choose house based on this relationship is calculated by **after_stat(count)** and is applied to y-axis. **geom_text()** is used to add the text labels to the graph. **scale_x_date()** can be used to format the date to show only day and month on the axis label and set breaks to 2 day. **theme_bw()** is a function that provide white background and black guidelines to the plot. **theme()** is used to control the non-data components such as the direction of the text to 90 degrees. **facet grid()** produces a 2d grid of panels defined by Area_Type and Tenant_Type which form the rows and columns. **scales="free"** refers to free all the axes label so that it varies among the rows.

```
>   print(house_rental_data %>% group_by(Tenant_Type,Area_Type) %>% count(),n=25)
# A tibble: 7 x 3
# Groups:   Tenant_Type, Area_Type [7]
  Tenant_Type      Area_Type       n
  <chr>            <chr>       <int>
1 Bachelors        Carpet Area   691
2 Bachelors        Super Area    139
3 Bachelors/Family Built Area      2
4 Bachelors/Family Carpet Area  1281
5 Bachelors/Family Super Area   2161
6 Family           Carpet Area   326
7 Family           Super Area    146
```

*Figure 6.2.15: Source Code and Output – Total Number of Tenants Choose House in Each Area Type*

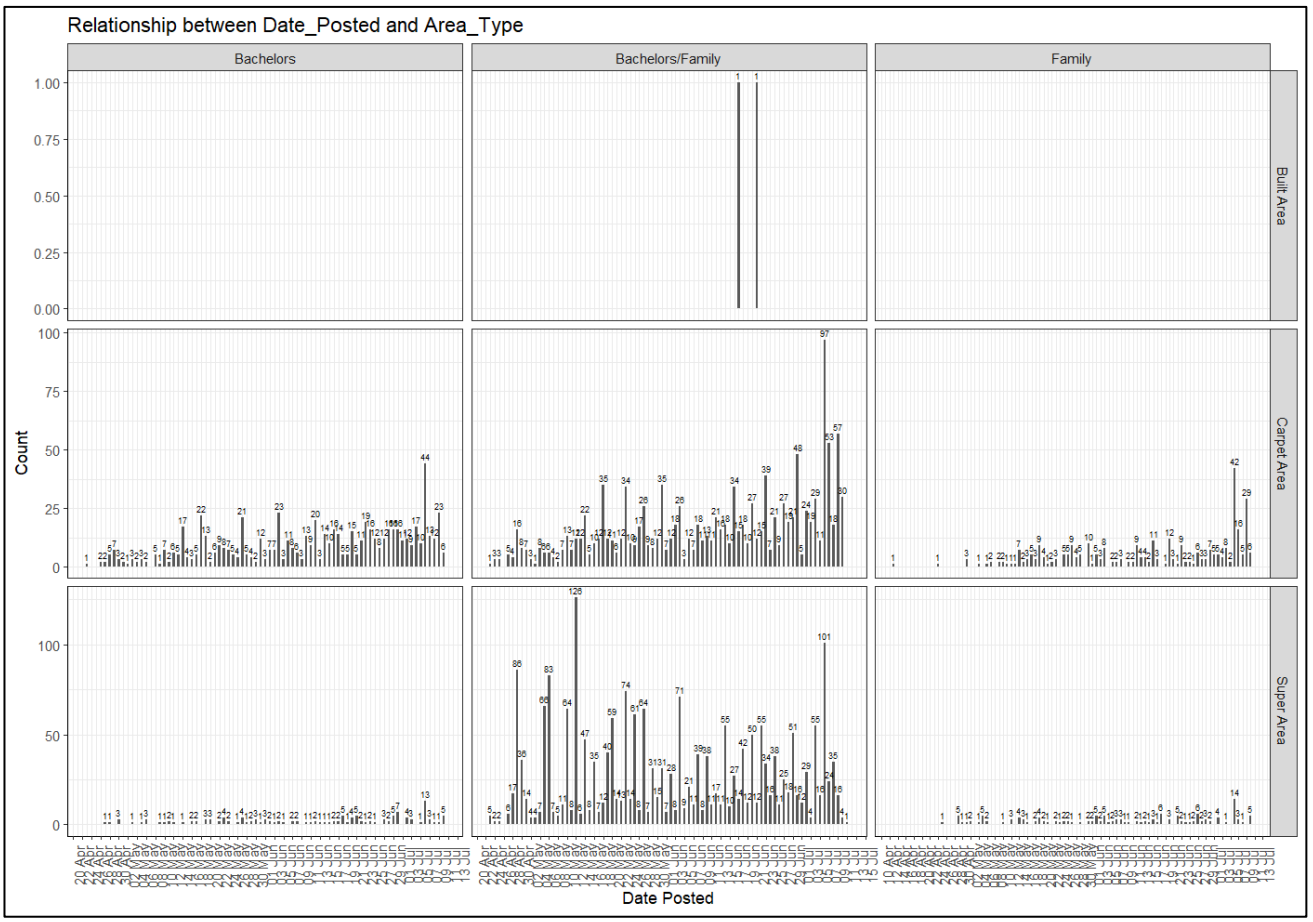



*Figure 6.2.16: Output – Bar Chart (Relationship between Date_Posted and Area_Type)*

**Explanation**

Figure 6.2.16 reveals that 44 of the bachelors prefer to choose their house located at carpet area on 4th July followed by 13 of them choose house located at super area respectively. None of them choose built area.

126 of the bachelors/family prefer to choose their house located at super area on 10th May, followed by 101 of them prefer house located at super area on 4th July. 97 of them prefer to choose house located at carpet area on 4th July.

42 of the family prefer to choose their house on 4th July with semi-furnished followed by 8 and 14 of them prefer to choose their house on the same date with furnished and unfurnished respectively.

**Findings**

- More tenants choose houses located at carpet area in July.
- None of the tenants except 2 bachelors/family choose houses located at built area.

## Analysis 2-8: Find the Relationship between Month and City

This analysis is conducted to investigate which period and which city is the most preferred by bachelors, bachelors/family and family.

```
# Bar Chart
ggplot(house_rental_data,aes(x=City,fill=City)) +
  geom_bar(stat = "count",position = "dodge") +
  geom_text(aes(label=after_stat(count)),stat="count",vjust=-0.5,size=2) +
  labs(x="Floor Preference",y="Count",
       title="Relationship between Month and City") +
  scale_fill_discrete(breaks=c('Apr','May','Jun','Jul')) +
  theme_bw() +
  theme(legend.position = "bottom") +
  facet_grid(factor(Month,levels=c('Apr','May','Jun','Jul'))~Tenant_Type)
```

*Figure 6.2.17: Source Code – Bar Chart to Show the Relationship between Month and City*

**Analysis Technique: Data Visualization**

Figure 6.2.17 depicts the source code used to create bar chart to show the percentage of tenants choose house based on the posted date and the area type. The number of tenants choose house based on this relationship is calculated by **after_stat(count)** and is applied to y-axis. **geom_text()** is used to add the text labels to the graph. **labs()** is used to modify the name of the axes and title. **scale_fill_discrete(breaks =)** is used to rearrange the order of legends. **theme_bw()** is a function that provide white background and black guidelines to the plot. **theme()** is used to control the non-data components such as the position of the legend to bottom. **facet grid()** produces a 2d grid of panels defined by Month and Tenant_Type which form the rows and columns.

```
>   print(house_rental_data %>% group_by(Tenant_Type,City) %>% count(),n=25)
# A tibble: 18 x 3
# Groups:   Tenant_Type, City [18]
   Tenant_Type      City          n
   <chr>            <chr>     <int>
 1 Bachelors        Bangalore   135
 2 Bachelors        Chennai     137
 3 Bachelors        Delhi       162
 4 Bachelors        Hyderabad   102
 5 Bachelors        Kolkata     122
 6 Bachelors        Mumbai      172
 7 Bachelors/Family Bangalore   694
 8 Bachelors/Family Chennai     649
 9 Bachelors/Family Delhi       432
10 Bachelors/Family Hyderabad   676
11 Bachelors/Family Kolkata     379
12 Bachelors/Family Mumbai      614
13 Family           Bangalore    57
14 Family           Chennai     105
15 Family           Delhi        11
16 Family           Hyderabad    90
17 Family           Kolkata      23
18 Family           Mumbai      186
```

*Figure 6.2.18: Source Code and Output – Number of Count Grouped by Tenant_Type and City*

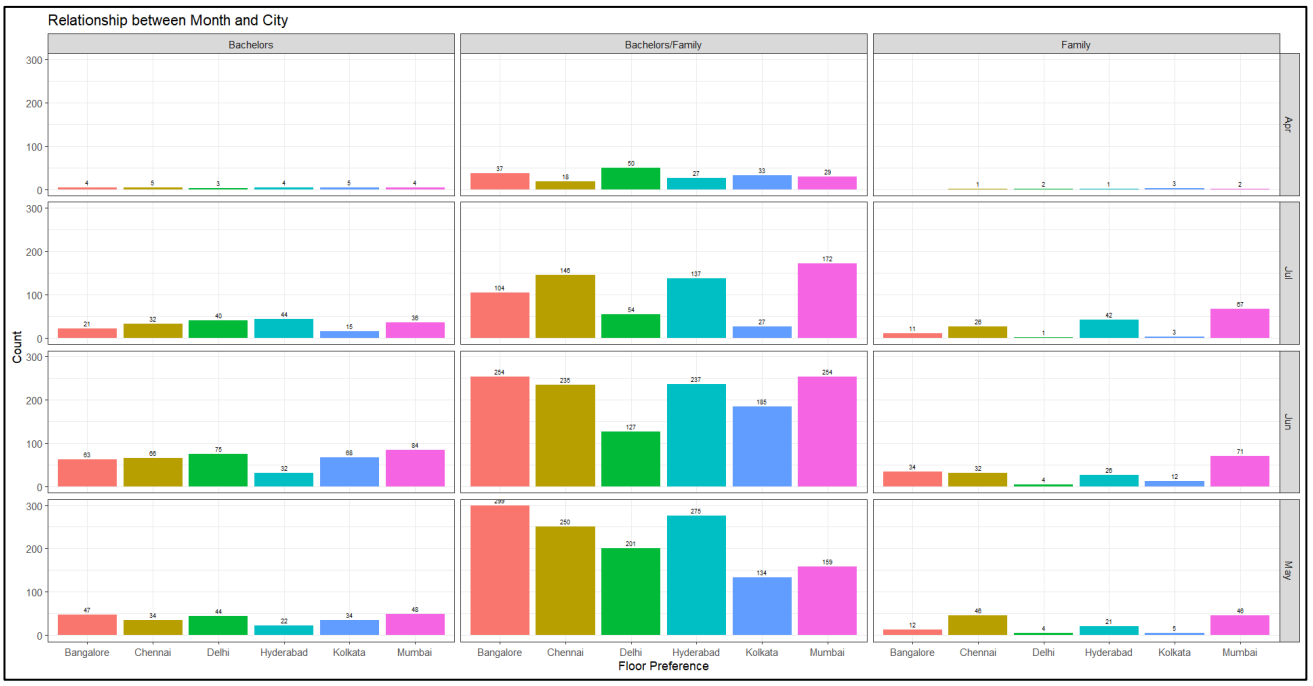



*Figure 6.2.19: Output – Bar Chart (Relationship between Month and City)*

**Explanation**

Figure 6.2.19 reveals that the city that is preferred the most by bachelors is Mumbai where the most preferred period is in June as there is 84 of them prefer to that period.

The city that bachelors/family preferred the most is Bangalore where the most preferred period is in May and June as there is 299 of them who prefer in May and 254 of them prefer in June, followed by Hyderabad with 275 of them who prefer in May and 237 in June.

The most preferred city by family is also Mumbai where the most preferred period is in June and July as there is 71 of them who prefer in June and 67 of them prefer in July, followed by Chennai with 46 of them who prefer in May and 32 in June.

**Findings**

- More tenants choose Mumbai as the house location in June
- Fewer tenants choose house in Kolkata.

## Conclusion for Question 2

1. Tenants prefer to choose 2 bedrooms, hall and kitchen in June and July
2. More tenants choose houses with 2 bathrooms in May, June, and July
3. More tenants prefer to choose house which located lower than 20 floor in May and June.
4. Tenants prefer to choose house size greater than 1000 sqft in July.
5. More Tenant prefer to choose house in July which there is high rental fee in that month
6. More tenants prefer to choose their house with semi-furnished in July.
7. More tenants prefer to choose their houses located at carpet area in July, followed by super area in May and July.
8. More tenants prefer to choose their houses located at Mumbai where the most preferred period is in June.

## Question 3: What are the Factors influencing Tenants to Choose Their Houses with respect to Bedroom_Hall_Kitchen?

### Analysis 3-1: Find the Relationship between Number of Bedroom_Hall_Kitchen and Number_of_Bathroom

This analysis is conducted to investigate how tenants choose their houses based on two variable which are the number of Bedroom_Hall_Kitchen and Number_of_Bathroom.

/**

** Following source code obtained from (Stulp, 2019)*

*/

```
# calculate Percentage Grouped by Tenant_Type, Bedroom_Hall_Kitchen and Number_of_Bathroom
group_tt_bhk_nob <- house_rental_data %>%
  group_by(Tenant_Type,Bedroom_Hall_Kitchen,Number_of_Bathroom) %>%
  summarise(number_cases=n())

# Scatter Plot
ggplot(group_tt_bhk_nob,aes(x=Bedroom_Hall_Kitchen,y=Number_of_Bathroom)) +
  geom_point(aes(size=number_cases,color=number_cases)) +
  labs(x="Number of Bedroom, Hall and Kitchen",y="Number of Bathroom") +
  ggtitle("Relationship between Bedroom_Hall_Kitchen and Number_of_Bathroom") +
  scale_x_continuous(breaks=seq(1,6,1)) +
  scale_y_continuous(breaks=seq(1,10,1)) +
  scale_color_gradient(low="#2c7fb8",high="pink") +
  facet_wrap(~Tenant_Type)
```

*Figure 6.3.1: Source Code – Scatter Plot to Show the Relationship between Number of Bedroom_Hall_Kitchen and Number_of_Bathroom*

**Analysis Technique: Data Visualization and Manipulation**

Figure 6.3.1 depicts the source code used to create the percentage of tenant choose house based on Bedroom_Hall_Kitchen and Number_of_Bathroom. The number of tenants choose house based on this relationship is calculate using **group_by()** and **count()**. Then the number is summarised and assign to variable “number_cases”.

The next source code is used to generate a scatter plot to study the distribution of tenant based on the relationship mentioned above. **scale_x_continuous()** can be used to set the breaks for x-axis label as well as **scale_y_continuous()** for y-axis. **scale_color_gradient()** can be used to modify the scale colour by providing the hex codes of the colour for low and high values. **facet_wrap()** is utilised to generate graphics tables that show the same graph for each group of tenants. **labs()** is used to modify the axes label names. Lastly, the title name is set using **ggtitle()**.

```
>   # Number_of_Bathroom for Bachelors
>   house_rental_data %>%
+     filter(Tenant_Type=="Bachelors") %>%
+     group_by(Bedroom_Hall_Kitchen,Number_of_Bathroom) %>%
+     summarise(number_cases=n())
# A tibble: 19 x 3
# Groups:   Bedroom_Hall_Kitchen [6]
   Bedroom_Hall_Kitchen Number_of_Bathroom number_cases
                  <int>             <int>        <int>
 1                    1                 1          156
 2                    1                 2           35
 3                    2                 1           80
 4                    2                 2          284
 5                    2                 3            9
 6                    3                 1            3
 7                    3                 2           70
 8                    3                 3          141
 9                    3                 4           13
10                    4                 2            2
11                    4                 3            6
12                    4                 4           14
13                    4                 5            8
14                    4                 6            3
15                    4                 7            1
16                    5                 2            1
17                    5                 3            1
18                    5                 5            2
19                    6                 5            1
```

```
>   # Calculate Number of Cases Grouped by Bedroom_Hall_Kitchen and
>   # Number_of_Bathroom for Bachelors/Family
>   print(house_rental_data %>%
+     filter(Tenant_Type=="Bachelors/Family") %>%
+     group_by(Bedroom_Hall_Kitchen,Number_of_Bathroom) %>%
+     summarise(number_cases=n()),n=30)
# A tibble: 29 x 3
# Groups:   Bedroom_Hall_Kitchen [6]
   Bedroom_Hall_Kitchen Number_of_Bathroom number_cases
                  <int>             <int>        <int>
 1                    1                 1          818
 2                    1                 2           92
 3                    1                 3            2
 4                    1                 4            1
 5                    1                10            1
 6                    2                 1          332
 7                    2                 2         1304
 8                    2                 3           41
 9                    2                 4            2
10                    3                 1           26
11                    3                 2          236
12                    3                 3          411
13                    3                 4           29
14                    3                 5            5
15                    3                 6            1
16                    4                 1            1
17                    4                 2           10
18                    4                 3           23
19                    4                 4           64
20                    4                 5           23
21                    4                 6            5
22                    5                 2            1
23                    5                 4            2
24                    5                 5            7
25                    5                 6            1
26                    6                 2            1
27                    6                 5            1
28                    6                 6            2
29                    6                 7            2
```

```
>   # Calculate Number of Cases Grouped by Bedroom_Hall_Kitchen and
>   # Number_of_Bathroom for Family
>   house_rental_data %>%
+     filter(Tenant_Type=="Family") %>%
+     group_by(Bedroom_Hall_Kitchen,Number_of_Bathroom) %>%
+     summarise(number_cases=n())
# A tibble: 17 x 3
# Groups:   Bedroom_Hall_Kitchen [6]
   Bedroom_Hall_Kitchen Number_of_Bathroom number_cases
                  <int>             <int>        <int>
 1                    1                 1           40
 2                    1                 2           22
 3                    2                 1           17
 4                    2                 2          193
 5                    2                 3            3
 6                    3                 1            1
 7                    3                 2           39
 8                    3                 3          110
 9                    3                 4           12
10                    3                 5            1
11                    4                 2            1
12                    4                 3            2
13                    4                 4           17
14                    4                 5            9
15                    5                 4            1
16                    5                 5            3
17                    6                 4            1
```

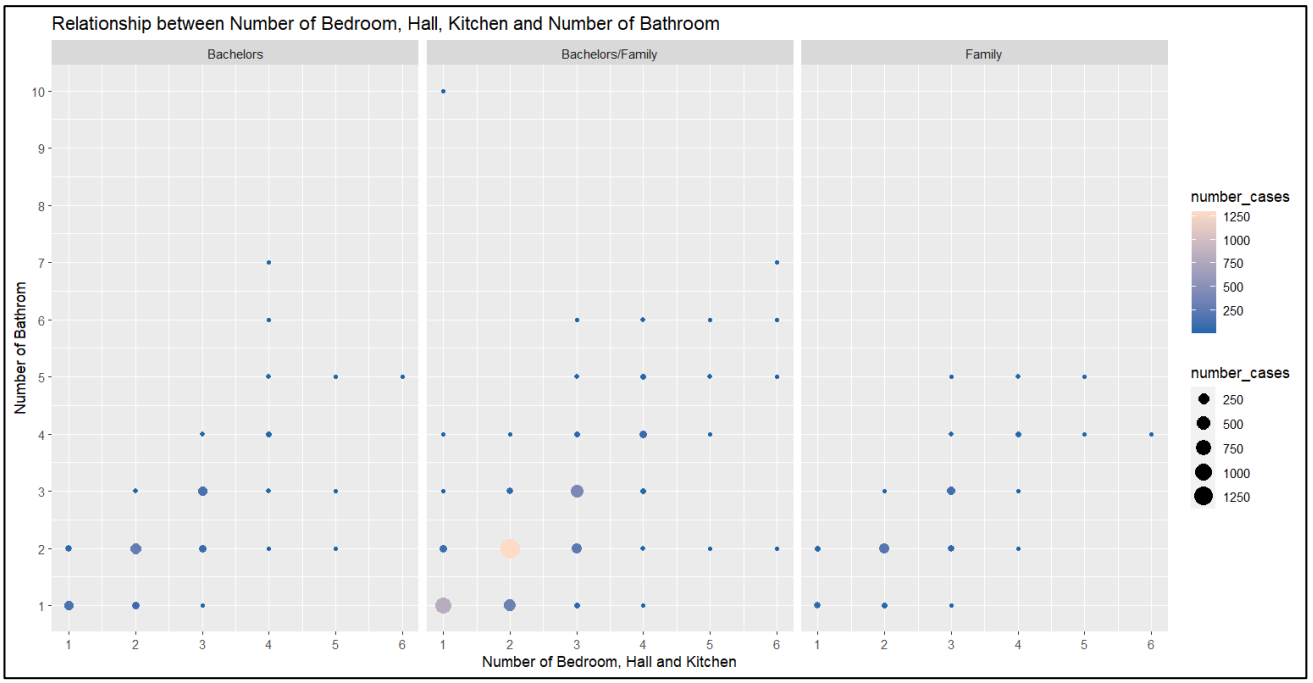



*Figure 6.3.2: Output – Scatter Plot (Relationship between Number of Bedroom_Hall_Kitchen and Number_of_Bathroom)*

**Explanation**

Figure 6.3.2 reveals that around 250 of the bachelors prefer to choose 2 bedrooms, hall and kitchen with 2 bathrooms, whereas the rest of them choose houses with different number of bathroom, number of bedrooms, hall and kitchen.

Approximately 1250 of the bachelors/family prefer to choose 2 bedrooms, hall and kitchen with 2 bathrooms as well, and around 1000 of the them prefer to choose 1 bedroom, hall and kitchen with 1 bathroom, whereas the rest of them choose houses with different number of bathroom, number of bedrooms, hall and kitchen.

Around 1000 of the family prefer to choose 2 bedrooms, hall and kitchen with 2 bathrooms, an around 500 of them prefer to choose 3 bedrooms, hall and kitchen with 3 bathrooms.

**Findings**

- Family tend to choose their preference house with more bathrooms, bedrooms, hall and kitchen
- Bachelors and bachelors/family tend to choose their preference house with 2 bathrooms, 2 bedrooms, hall and kitchen

## Analysis 3-2: Find the Relationship between Number of Bedroom_Hall_Kitchen and Floor_Preference

This analysis is conducted to investigate what number of bedrooms, hall and kitchen and which floor do tenants prefer the most.

```
# Box Plot
ggplot(house_rental_data,aes(x=Bedroom_Hall_Kitchen,y=Floor_Preference)) +
  geom_boxplot(aes(x=factor(Bedroom_Hall_Kitchen),
                   y=Floor_Preference,color=factor(Bedroom_Hall_Kitchen)))+
  labs(x= "Number of Bedroom, Hall and Kitchen",y="Floor Preferance",
       color="Number of Bedroom, Hall and Kitchen",
       title="Relationship between Number of Bedroom, Hall, Kitchen and Floor Preference") +
  scale_fill_discrete(name="Number of Bedroom, Hall and Kitchen") +
  facet_wrap(~Tenant_Type) +
  theme(legend.position = "bottom")
```

*Figure 6.3.3: Source Code – Box Plot to Show the Relationship between Number of Bedroom_Hall_Kitchen and Floor_Preference*

**Analysis Technique: Data Visualization**

Figure 6.3.3 depicts the source code used to generate a box plot to study the distribution of tenant based on Bedroom_Hall_Kitchen and Floor_Preference. **geom_boxplot()** is used to create box plot graph. **labs()** is used to modify the names of the axes label and plot title. **scale_fill_discrete()** can be used to set the name for the legends. **facet_wrap()** is utilised to generate graphics tables that show the same graph for each group of tenants. **labs()** is used to modify the axes label names. Lastly, the title name is set using **ggtitle()**.

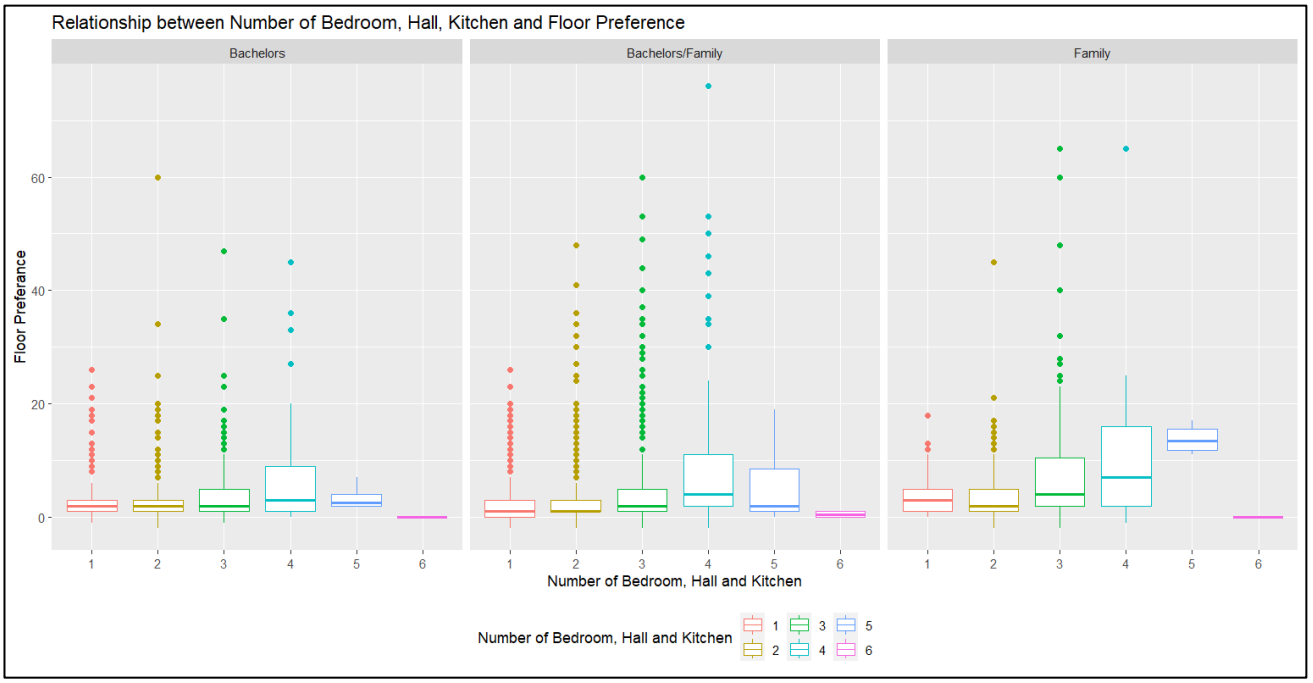



*Figure 6.3.4: Output – Box Plot (Relationship between Bedroom_Hall_Kitchen and Number_of_Bathroom)*

**Explanation**

Figure 6.3.4 suggests that more bachelors prefer 1,2, 4 and 5 number of bedrooms, hall and kitchen with floor level lower than 5, whereas more bachelors/family prefer houses with 1,2 and 6 bedrooms, hall and kitchen with floor level lower than 5. Fewer bachelors/family prefer 4 and 5 bedrooms, hall, and kitchen with floor level than 10. More family prefer their houses with 6 bedrooms, hall and kitchen located lower than 5 level.

**Findings**

- Bachelors and bachelors/family prefer houses with 1 and 2 number of bedrooms, hall, and kitchen with floor level lower than 5
- Bachelors prefer houses with 4 bedrooms, hall, and kitchen with floor lower than 5.
- Bachelors/family also prefer houses with 6 bedrooms, hall, and kitchen with lower floor.
- Family prefer houses with 6 bedrooms, hall, and kitchen with lower floor

## Analysis 3-3: Find the Relationship between Number of Bedroom_Hall_Kitchen and House_Size

This analysis is conducted to investigate how many bedrooms, hall and kitchen and how big the house do tenants prefer the most.

```
# Violin Plot with Box Plot
ggplot(house_rental_data,aes(x=House_Size,
                             y=factor(Bedroom_Hall_Kitchen),
                             color=factor(Bedroom_Hall_Kitchen))) +
  geom_violin() +
  geom_boxplot(width=0.1) +
  scale_x_continuous(breaks=seq(0,8000,500)) +
  scale_y_discrete(breaks=seq(1,6,1)) +
  facet_wrap(~Tenant_Type) +
  theme(legend.position = "bottom") +
  labs(x="House Size",y="Number of Bedroom, Hall and Kitchen",
       color="Number of Bedroom, Hall and Kitchen",
       title="Relationship between Bedroom_Hall_Kitchen and House_Size")
```

*Figure 6.3.5: Source Code – Bar Chart to Show the Relationship between Bedroom_Hall_Kitchen and House_Size*

**Analysis Technique: Data Visualization**

Figure 6.3.5 depicts the source code used to create violin plot with box plot inside to show the number of tenants choose house based on number of Bedroom_Hall_Kitchen and House_Size. **scale_x_continuous()** is the position scales for continuous data of x-axis. The x-axis label is modified so that it produce 0 to 8000 with 500 gaps between them in the graph and it is referred to breaks. **scale_y_discrete ()** is the position scales for discrete data of y-axis. The y-axis label is modified so that it produce 1 to 6 with 1 gap between them in the graph and it is referred to breaks. **facet_wrap()** is utilised to generate graphics tables that show the same graph for each group of tenants. The **theme()** is used to customize the non-data components such as the position of the legend. **labs()** is used to modify the name of the axes, name of the legend and title plot.

```
>   house_rental_data %>%
+     group_by(Tenant_Type,Bedroom_Hall_Kitchen) %>%
+     count()
# A tibble: 18 × 3
# Groups:   Tenant_Type, Bedroom_Hall_Kitchen [18]
   Tenant_Type      Bedroom_Hall_Kitchen     n
   <chr>                           <int> <int>
 1 Bachelors                           1   191
 2 Bachelors                           2   373
 3 Bachelors                           3   227
 4 Bachelors                           4    34
 5 Bachelors                           5     4
 6 Bachelors                           6     1
 7 Bachelors/Family                    1   914
 8 Bachelors/Family                    2  1679
 9 Bachelors/Family                    3   708
10 Bachelors/Family                    4   126
11 Bachelors/Family                    5    11
12 Bachelors/Family                    6     6
13 Family                              1    62
14 Family                              2   213
15 Family                              3   163
16 Family                              4    29
17 Family                              5     4
18 Family                              6     1
```

*Figure 6.3.6: Count Number of Tenants Based on Bedroom_Hall_Kitchen*

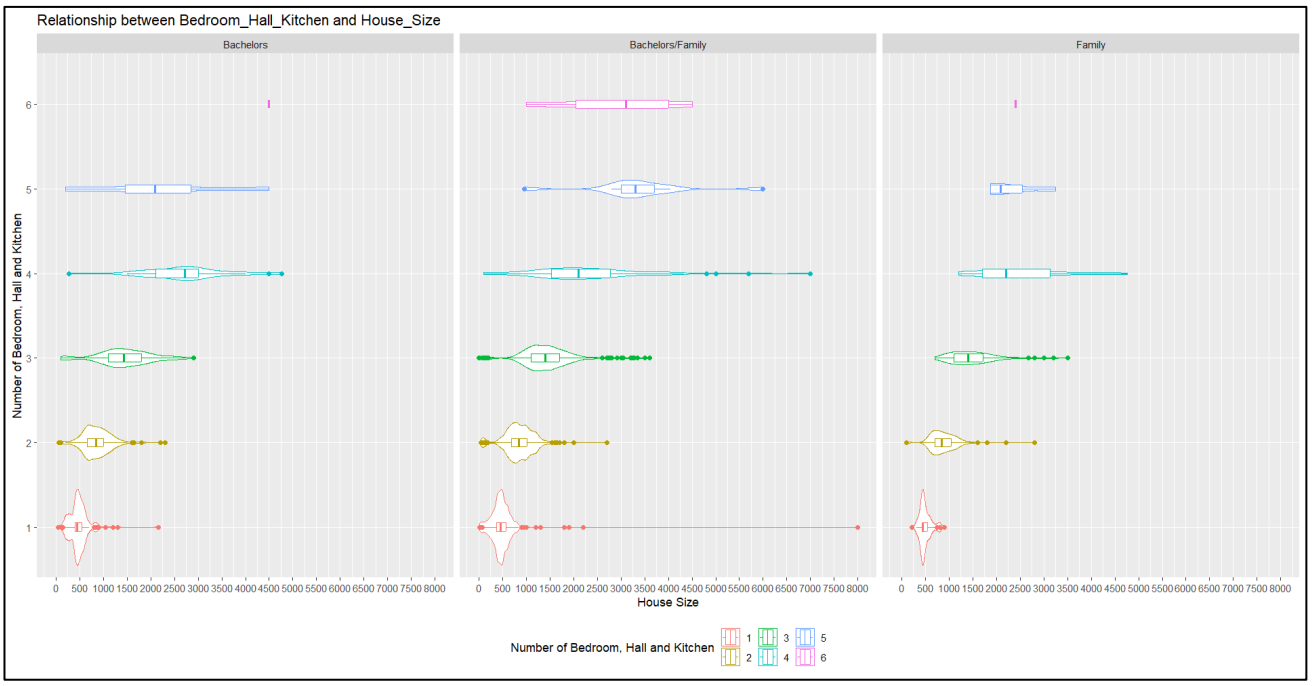



*Figure 6.3.7: Output – Bar Chart (Relationship between Bedroom_Hall_Kitchen and House_Size)*

**Explanation**

From Figure 6.3.6 and Figure 6.3.7, more bachelors prefer to choose houses with 2 bedrooms, hall and kitchen and house size between 700 sqft and 1000 sqft. More bachelors/family prefer to choose houses with 2 bedrooms, hall and kitchen and house size between 700 sqft and 1000 sqft. More family prefer to choose houses with 2 bedrooms, hall and kitchen and house size between 700 sqft and 1100 sqft.

**Findings**

- More bachelors prefer houses with 2 bedrooms, hall and kitchen and house size of 700 sqft.
- More bachelors/family prefer houses with 2 bedrooms, hall and kitchen and house size of 800 sqft.
- More family prefer houses with 2 bedrooms, hall and kitchen and house size of 700 sqft.

## Analysis 3-4: Find the Relationship between Number of Bedroom, Hall, Kitchen and Rental Fee

This analysis is conducted to find out which number of bedrooms, hall and kitchen and the amount of rental fee of it is the most preferred by the tenants.

/**

** Following source code obtained from (Priyank, 2022)*

*/

```
# Violin Plot with Box Plot
ggplot(rf_no_outliers,aes(x=Bedroom_Hall_Kitchen,
                          y=Rental_Fee,
                          color=factor(Bedroom_Hall_Kitchen))) +
  geom_violin() +
  geom_boxplot(width=0.1) +
  scale_x_continuous(breaks=seq(1,6,1)) +
  scale_y_continuous(labels=scales::comma) +
  facet_wrap(~Tenant_Type) +
  theme(legend.position = "bottom") +
  labs(x="Number of Bedroom, Hall and Kitchen",y="Rental Fee",
       color="Number of Bedroom, Hall and Kitchen",
       title="Relationship between Number of Bedroom_Hall_Kitchen and Rental_Fee")
```

*Figure 6.3.8: Source Code – Violin Plot to Show the Relationship between Bedroom_Hall_Kitchen and Rental_Fee*

**Analysis Technique: Data Visualization and Manipulation**

Figure 6.3.8 depicts the source code used to create violin plot with box plot inside it to show the number of tenants choose house based on number of Bedroom_Hall_Kitchen and Rental_Fee. **scale_x_continuous()** is the position scales for continuous data of x-axis and can be used to set the breaks from 1 to 6 with gap of 1 for the label. **scale_y_continuous()** is the position scales for continuous data of y-axis. **labels=scales::comma** is used to remove the e notation of the values using **scale_y_continuous**. **facet_wrap()** is utilised to generate graphics tables that show the same graph for each group of tenants. **theme()** is used to control the non-data components such as the position of the legend to the bottom. **labs()** is used to modify the name of the axes label, legend, and title plot.

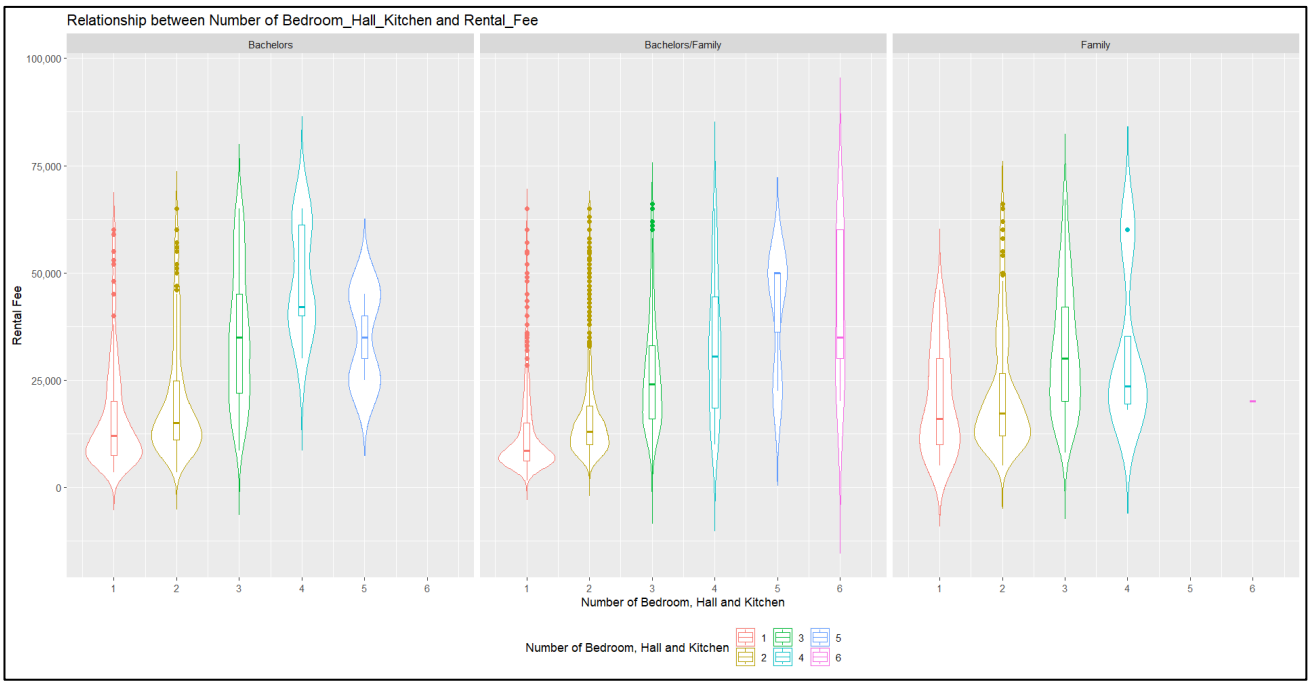



*Figure 6.3.9: Output – Violin Plot with Box Plot (Relationship between Bedroom_Hall_Kitchen and Rental_Fee)*

**Explanation**

Figure 6.3.9 reveals that more bachelors prefer 1 or 2 bedrooms, hall, and kitchen with rental fee less than RM20,000. Fewer bachelors prefer 3 bedrooms, hall, and kitchen with rental fee more than RM20,000 and fewer of them also prefer 4 and 5 bedrooms, hall and kitchen with rental fee more than RM20,000.

More bachelors/family prefer 1 or 2 bedrooms, hall, and kitchen with rental fee less than RM20,000. Fewer of theme prefer 3, 4, 5 or 6 bedrooms, hall, and kitchen within the rental fee range of RM20,000 to RM60,000.

More family prefer 2 bedrooms, hall, and kitchen with rental fee less than RM20,000 compared to 1 bedroom, hall and kitchen.

**Findings**

- More bachelors, bachelors/family and family prefer 2 bedrooms, hall, and kitchen with rental fee less than RM20,000.
- Bachelors and bachelors/family also prefer 1 bedroom, hall, and kitchen with rental fee less than RM20,000.
- Fewer bachelors, bachelors/family and family prefer more than 3 bedrooms, hall, and kitchen within the range rental fee from RM20,000 and RM60,000.

## Analysis 3-5: Find the Relationship between Number of Bedroom_Hall_Kitchen and Furnishing Status

This analysis is conducted to find out which number of bedrooms, hall and kitchen and the furnishing status of it is the most preferred by the tenants.

```
# Calculate Percentage Grouped by Tenant_Type, Bedroom_Hall_Kitchen and Furnishing_Status
group_tt_bhk_fs <- house_rental_data %>%
  group_by(Tenant_Type,Bedroom_Hall_Kitchen,Furnishing_Status) %>%
  count() %>% ungroup() %>% mutate(perc=n/sum(n)) %>% arrange(perc) %>%
  mutate(labels=scales::percent(perc))
# Bar Chart
group_tt_bhk_fs %>% ggplot(aes(Bedroom_Hall_Kitchen,perc,fill=Furnishing_Status)) +
  geom_bar(stat="identity",position="dodge") +
  facet_wrap(~Tenant_Type) +
  geom_text(aes(label=labels),size=2,vjust=-0.3,position=position_dodge(width=1)) +
  labs(x="Number of Bedroom, Hall and Kitchen",y="Percentage",
       title="Relationship between Number of Bedroom_Hall_Kitchen and Furnishing Status") +
  scale_x_continuous(breaks=seq(1,6,1)) +
  scale_y_continuous(labels=scales::percent) +
  theme(legend.position="bottom",
        legend.background = element_rect(fill="antiquewhite")) +
  scale_fill_discrete(name="Furnishing Status")
```

*Figure 6.3.10: Source Code – Bar Chart to Show the Relationship between Bedroom_Hall_Kitchen and Furnishing_Status*

**Analysis Technique: Data Visualization and Manipulation**

Figure 6.3.10 depicts the source code used to create the percentage of tenant choose house based on number of Bedroom_Hall_Kitchen and Furnishing_Status. The number of tenants choose house based on this relationship is calculated using **group_by()** and **count()**. Then the percentage is calculated by finding the total numbers using **sum()** and dividing it by the number calculated using **count()** and use **mutate()** to calculate the percentage and assigned it to variable "perc". The percentage is then arranged in ascending using **arrange()**. In order to show the percentage symbol in the graph, **scales::percent()** is used and assigned it to labels.

The next source code is used to generate a bar chart to study the distribution of tenant based on point of contact. The position with **position_dodge()** in geom_text put the bars side-by-side and the alignment of the text can be adjusted by width argument. **labs()** is used to modify the names of the axes and the title plot. The **theme()** is used to customize the non-data components such as the position of the legend to bottom and set the background of the legend to antiquewhite. **scale_fill_discrete()** is used to modify the legends in the way that the name of the legends can be modified.

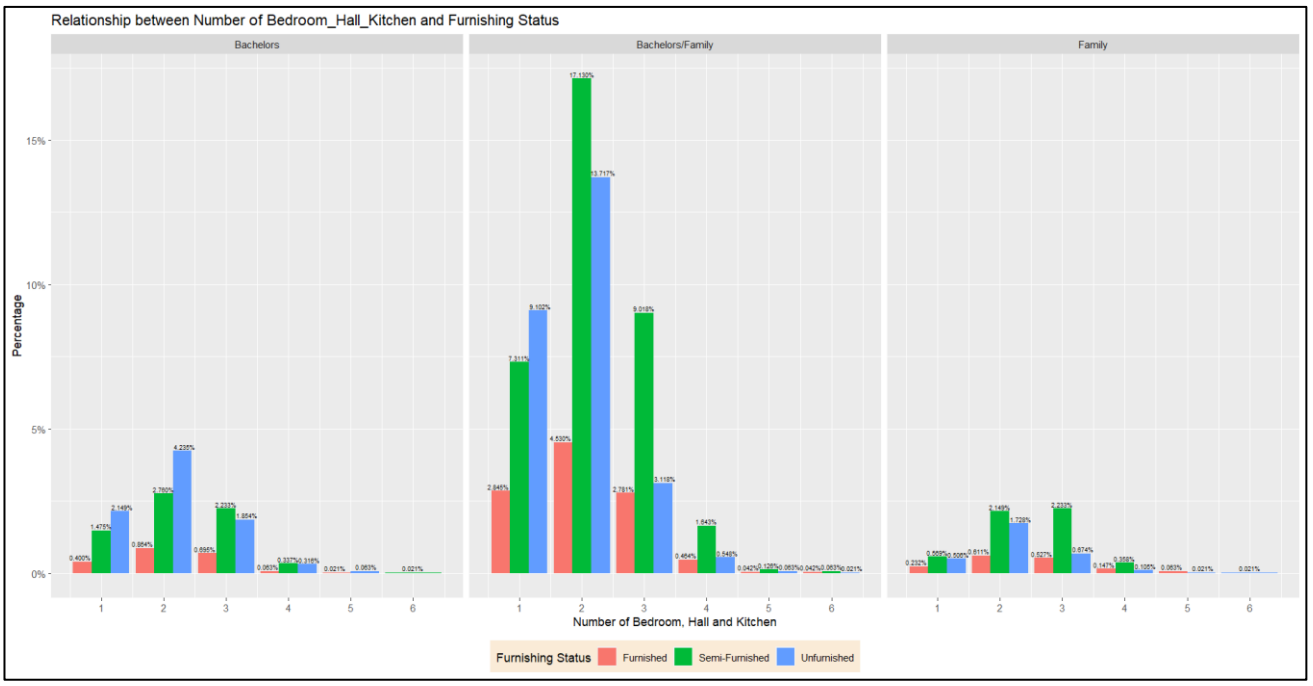



*Figure 6.3.11: Output – Bar Chart (Relationship between Bedroom_Hall_Kitchen and Furnishing_Status)*

**Explanation**

As shown in Figure 6.3.11 , it can be clearly seen that bachelors/family prefer their house to be semi-furnished instead of furnished. 17.13% of them prefer 2 bedrooms, hall and kitchen while 9.018% of them prefer 3 bedrooms, hall and kitchen. 13.717% of the bachelors/family prefer their house to be unfurnished and with 2 bedrooms, hall and kitchen, while 9.102% of them prefer 1 bedroom, hall and kitchen. Less than 5% of them prefer to choose 5 bedrooms, hall, and kitchen no matter it is unfurnished, semi-furnished or furnished.

For bachelors, they are in contrast of bachelors/family where they prefer unfurnished house with 2 bedrooms, hall, and kitchen at 4.235% but 2.76% of them prefer the house to be semi-furnished.

For family, they are similar to bachelors/family where they prefer semi-furnished house with slightly higher percentage (2.233%) on house with 3 bedrooms, hall and kitchen compared to house with 2 bedrooms, hall, and kitchen (2.149%). Fewer of them prefer unfurnished house with 2 bedrooms, hall, and kitchen too.

**Findings**

- More bachelors and bachelors/family prefer to choose house that is unfurnished with 2 bedrooms, hall, and kitchen.
- More bachelors/family prefer to choose house that is semi-furnished with 2 bedrooms, hall, and kitchen.
- Family prefer to choose both unfurnished and semi-furnished house with 2 bedrooms, hall, and kitchen.

## Analysis 3-6: Find the Relationship between Number of Bedroom, Hall, Kitchen and Area Type

This analysis is conducted to investigate how many tenants choose their house based on both number of Bedroom_Hall_Kitchen and the Area_Type.

```
# Calculate Percentage Grouped by Bedroom_Hall_Kitchen and Area_Type
group_tt_bhk_at <- house_rental_data %>%
  group_by(Tenant_Type,Bedroom_Hall_Kitchen,Area_Type) %>%
  count() %>% ungroup() %>% mutate(perc=n/sum(n)) %>%
  arrange(perc) %>% mutate(labels=scales::percent(perc))

# Bar Chart
ggplot(group_tt_bhk_at,aes(x=Bedroom_Hall_Kitchen,y=perc,fill=Area_Type)) +
  geom_bar(stat="identity",position="dodge") +
  geom_text_repel(aes(label=labels),
                  position=position_dodge(width=0.1),size=3,max.overlaps = 20) +
  labs(x="Number of Bedroom, Hall and Kitchen",y="Percentage",
       title="Relationship between Number of Bedroom_Hall_Kitchen and Area Type") +
  theme(legend.position="bottom",
        legend.background = element_rect(fill="antiquewhite")) +
  scale_fill_discrete(name="Area Type") +
  scale_x_continuous(breaks=seq(1,6,1)) +
  scale_y_continuous(labels=scales::percent) +
  facet_wrap(~Tenant_Type)
```

*Figure 6.3.12: Source Code – Bar Chart to Show the Relationship between Number of Bedroom_Hall_Kitchen and Area_Type*

**Analysis Technique: Data Visualization and Manipulation**

Figure 6.3.12 depicts the source code used to create the percentage of tenant choose house based on number of Bedroom_Hall_Kitchen and Area_Type. The number of tenants choose house based on this relationship is calculated using **group_by()** and **count()**. Then the percentage is calculated by finding the total numbers using **sum()** and dividing it by the number calculated from **count()** and use **mutate()** to calculate the percentage and assigned it to variable "perc". The percentage is then arranged in ascending using **arrange()**. In order to show the percentage symbol in the graph, **scales::percent()** is used and assigned it to labels.

The next source code is used to generate a bar chart to study the distribution of tenant based on Bedroom_Hall_Kitchen and Area_Type. **geom_text_repel()** is used to add text directly to the graph and repel overlapping text labels .The position with **position_dodge()** in geom_text_repel() put the bars side-by-side and the alignment of the text can be adjusted by width argument. **labs()** is used to modify the names of the axes and the title plot. The **theme()** is used to customize the non-data components such as the position of the legend to bottom and set the background of the legend to antiquewhite. **scale_fill_discrete()** is used to modify the legends in the way that the name of the legends can be modified. **scale_x_continuous()** is the position scales for continuous data of x-axis, while **scale_y_continuous()** is the position scales for continuous data of y-axis. The x-axis label is modified so that it produce 1 to 6 with 1 space between them in the graph and it is referred to breaks. The y-axis label is modified so that it shows percentage, and this is referred to labels. **facet_wrap()** is utilised to generate graphics tables that show the same graph for each group of tenants.

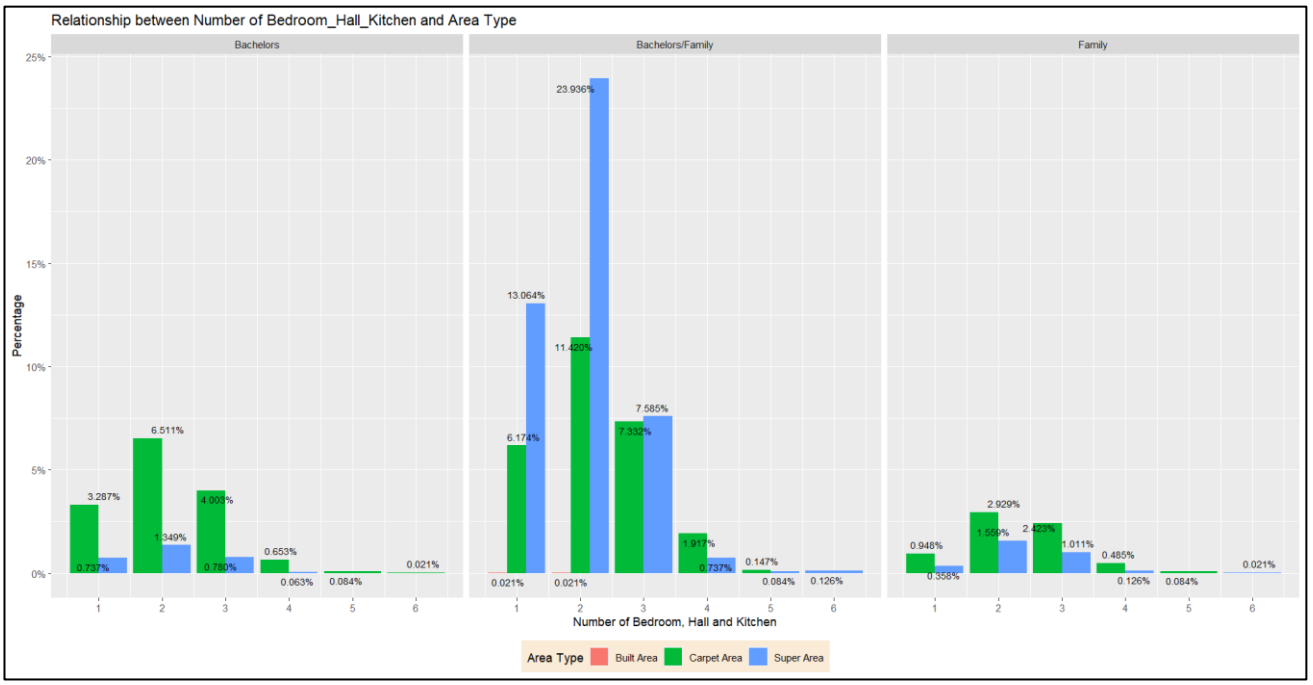



*Figure 6.3.13: Output – Bar Chart (Relationship between Bedroom_Hall_Kitchen and Area_Type)*

**Explanation**

As shown in Figure 6.3.13 , it can be clearly seen that bachelors/family prefer their house to be located at super area instead of built area. 23.936% of them prefer house with 2 bedrooms, hall, and kitchen while 13.064% of them prefer house with 1 bedroom, hall, and kitchen.

For bachelors, it is totally different from bachelors/family. Around 15% of the bachelors prefer their house to be located at carpet area. 6.511% of them prefer house with 2 bedrooms, hall, and kitchen, followed by 4.003% and 3.287% for 3 bedrooms, hall, and kitchen and 1 bedroom, hall and kitchen respectively.

For family, their choices are similar to the choice of bachelors. 2.929% of them prefer house located at carpet area with 2 bedrooms, hall, and kitchen, followed by 2.423% of them prefer house with 3 bedrooms, hall, and kitchen.

**Findings**

- More bachelors/family prefer to choose house that is located at super area with 2 bedrooms, hall, and kitchen.
- More bachelors and family prefer to choose house that is located at carpet area with 2 and 3 bedrooms, hall, and kitchen.

## Analysis 3-7: Find the Relationship between Number of Bedroom, Hall, Kitchen and City

This analysis is conducted to investigate in which city, tenants choose their house based on the number of bedrooms, hall and kitchen.

```
# Calculate Percentage Grouped by Tenant_Type, Bedroom_Hall_Kitchen and City
group_tt_bhk_c <- house_rental_data %>%
  group_by(Bedroom_Hall_Kitchen,City,Tenant_Type) %>%
  count()  %>%
  mutate(number_cases=n)

# Bar Chart
group_tt_bhk_c %>% ggplot(aes(Bedroom_Hall_Kitchen,number_cases,fill=City)) +
  geom_bar(stat="identity",position="dodge") +
  geom_text(aes(label=number_cases),size=3,position=position_dodge(width=0.5),vjust=-0.2) +
  labs(x="Number of Bedroom, Hall and Kitchen",y="Count",
       title="Relationship between Number of Bedroom_Hall_Kitchen and City") +
  scale_x_continuous(breaks=seq(1,6,1)) +
  scale_fill_discrete(name="City") +
  theme(legend.position="bottom",
        legend.background = element_rect(fill="antiquewhite")) +
  facet_grid(Tenant_Type~City)
```

*Figure 6.3.14: Source Code – Bar Chat to Show the Relationship between Number of Bedroom_Hall_Kitchen and City*

**Analysis Technique: Data Visualization and Manipulation**

Figure 6.3.14 depicts the source code used to create the percentage of tenant choose house based on number of Bedroom_Hall_Kitchen and City. The number of tenants choose house based on this relationship is calculated using **group_by()** and **count()**. Then use **mutate()** to sum up the number using **sum()** and assign to variable "number_cases"

The next source code is used to generate a bar chart to study the distribution of tenant based on Bedroom_Hall_Kitchen and Area_Type. **geom_text()** is used to add text directly to the graph. The position with **position_dodge()** in geom_text() put the bars side-by-side and the alignment of the text can be adjusted by width argument. **labs()** is used to modify the names of the axes and the title plot. The **theme()** is used to customize the non-data components such as the position of the legend to bottom and set the background of the legend to antiquewhite. **scale_fill_discrete()** is used to modify the legends in the way that the name of the legends can be modified. **scale_x_continuous()** is the position scales for continuous data of x-axis. The x-axis label is modified so that it produce 1 to 6 with 1 space between them in the graph and it is referred to breaks. **facet_grid()** generates a 2D grid of panels with the rows and columns given by Tenant_Type and City.

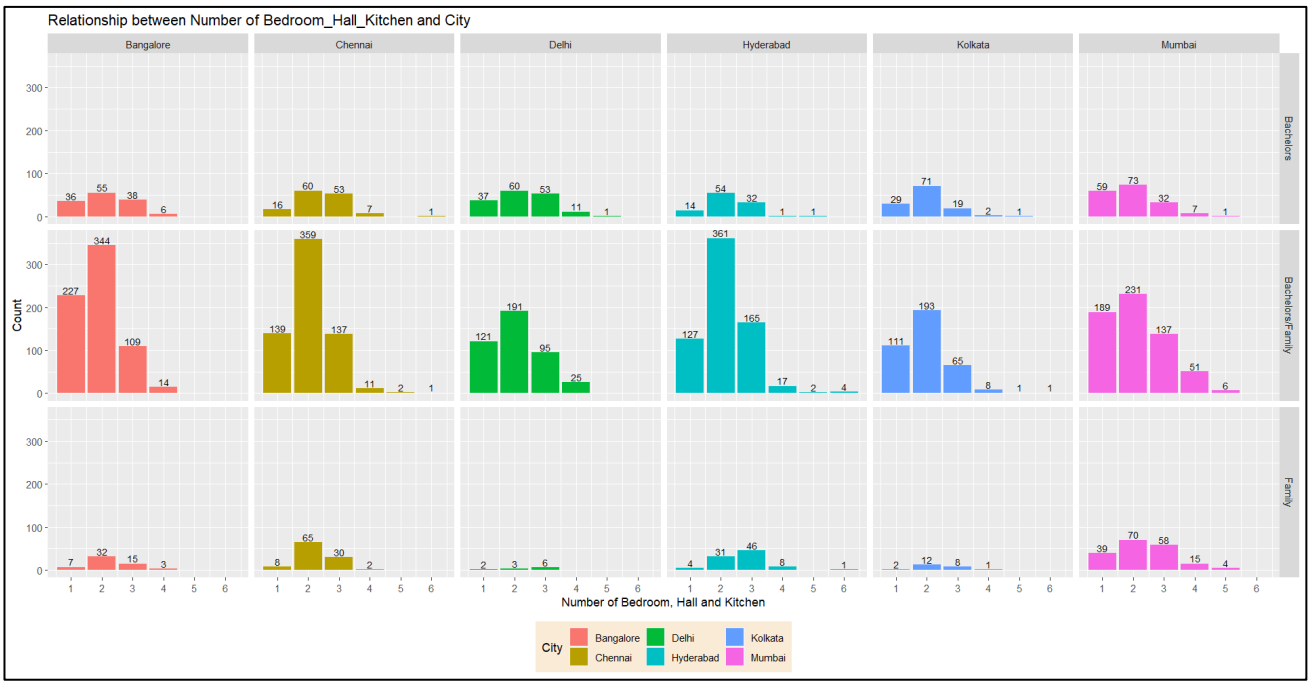



*Figure 6.3.15: Output – Bar Chart (Relationship between Bedroom_Hall_Kitchen and City)*

**Explanation**

As shown in Figure 6.3.15, it can be clearly seen that Mumbai is the top preference for bachelors and family to choose their houses there, whereas the city preference for bachelors/family is more to Bangalore.

In Bangalore, more bachelors, bachelors/family, and family prefer houses with 2 bedrooms, hall, and kitchen. In Chennai, more bachelors, bachelors/family, and family also prefer houses with 2 bedrooms, hall, and kitchen. In Delhi, more bachelors and bachelors/family prefer houses with 2 bedrooms, hall, and kitchen, whereas the majority of family choosing 3 bedrooms, hall and kitchen is fewer. In Hyderabad, more bachelors and bachelors/family choose houses with 2 bedrooms, hall, and kitchen, whereas family choose houses with 3 bedrooms, hall, and kitchen. In Kolkata, more bachelors, bachelors/family, and family prefer houses with 2 bedrooms, hall, and kitchen as well as in Mumbai.

**Findings**

- More bachelors, bachelors/family prefer houses with 2 bedrooms, hall and kitchen in Bangalore, Chennai, Kolkata, and Mumbai.
- Slightly more family prefer houses with 3 bedrooms, hall, and kitchen in Delhi and Hyderabad.

## Analysis 3-8: Find the Relationship between Number of Bedroom, Hall, Kitchen and Point of Contact

This analysis is conducted to investigate on how point of contact can influence tenants choose their house based on the number of bedrooms, hall, and kitchen.

```
# Calculate Percentage Grouped by Bedroom_Hall_Kitchen, Point_of_Contact and Tenant_Type
group_tt_bhk_poc <- house_rental_data %>%
  group_by(Tenant_Type,Bedroom_Hall_Kitchen,Point_of_Contact) %>%
  count() %>% ungroup() %>% mutate(perc=n/sum(n)) %>% arrange(perc) %>%
  mutate(labels=scales::percent(perc))

# Bar Chart
ggplot(group_tt_bhk_poc,aes(x=Bedroom_Hall_Kitchen,y=perc,fill=factor(Point_of_Contact))) +
  geom_bar(stat="identity",position="dodge") +
  geom_text(aes(label=labels),size=3,vjust=-0.3,position=position_dodge(width=1)) +
  labs(x="Number of Bedroom, Hall and Kitchen",y="Percentage",
       title="Relationship between Number of Bedroom_Hall_Kitchen and Point_of_Contact") +
  scale_y_continuous(labels=scales::percent) +
  scale_fill_discrete(name="Point of Contact") +
  theme(legend.position="bottom",
        legend.background = element_rect(fill="antiquewhite")) +
  facet_grid(Point_of_Contact~Tenant_Type)
```

*Figure 6.3.16: Source Code – Bar Chart to Show the Relationship between Bedroom_Hall_Kitchen and Point_of_Contact*

**Analysis Technique: Data Visualization and Manipulation**

Figure 6.3.16 depicts the source code used to create the percentage of tenant choose house based on number of Bedroom_Hall_Kitchen and Point_of_Contact. The number of tenants choose house based on this relationship is calculated using **group_by()** and **count()**. Then the percentage is calculated by finding the total numbers using **sum()** and use **mutate()** to calculate the percentage and assigned it to variable “perc”. The percentage is then arranged in ascending using **arrange()**. In order to show the percentage symbol in the graph, **scales::percent()** is used and assigned it to labels.

The next source code is used to generate a bar chart to study the distribution of tenant based on Bedroom_Hall_Kitchen and Area_Type. **geom_text()** is used to add text directly to the graph. The position with **position_dodge()** in geom_text() put the bars side-by-side and the alignment of the text can be adjusted by width argument. **labs()** is used to modify the names of the axes and the title plot. The **theme()** is used to customize the non-data components such as the position of the legend to bottom and set the background of the legend to antiquewhite. **scale_fill_discrete()** is used to modify the legends in the way that the name of the legends can be modified. **scale_y_continuous()** is the position scales for continuous data of y-axis. labels in y-axis can be modified to percentage using **scale_y_continuous(labels=scales::percent)**. **facet_grid()** generates a 2D grid of panels with the rows and columns given by Point_of_Contact and Tenant_Type.

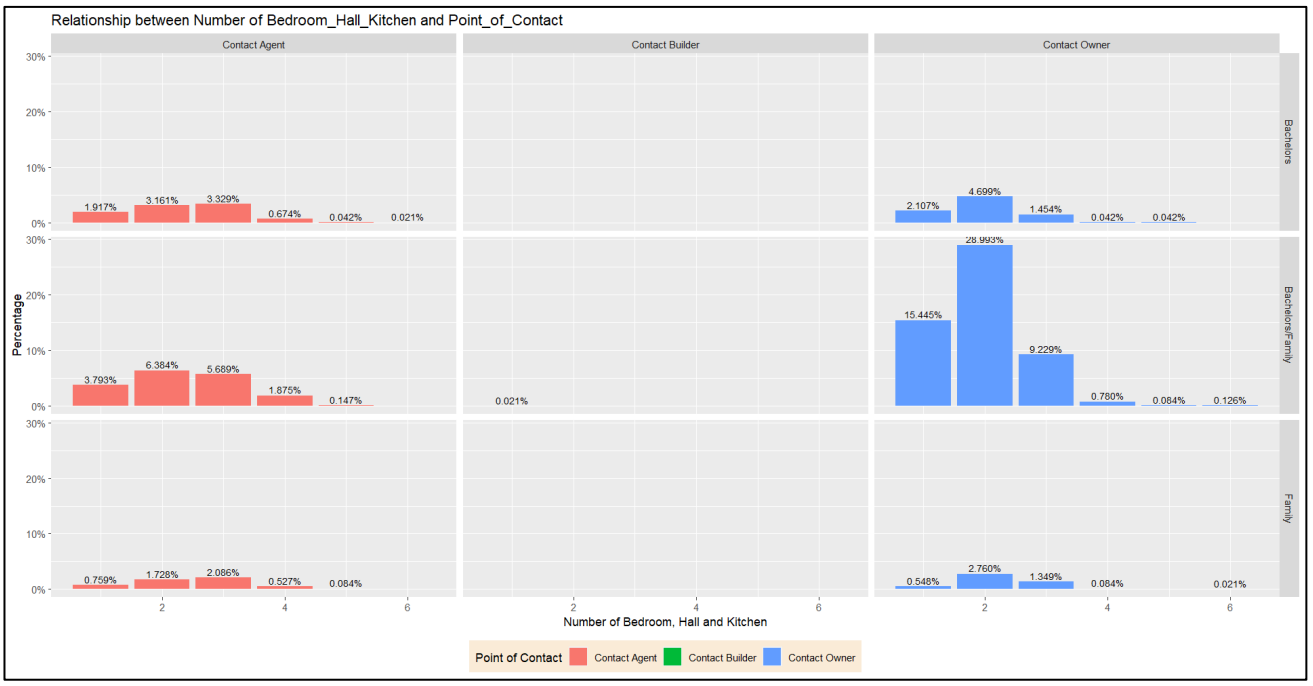



*Figure 6.3.17: Output – Bar Chart (Relationship between Bedroom_Hall_Kitchen and Point_of_Contact)*

**Explanation**

As shown in Figure 6.3.17, it can be clearly seen that more than 50% of bachelors/family prefer to choose their house by contacting owner. 28.993% of the bachelors/family prefer 2 bedrooms, hall and kitchen, followed by 15.445% of them prefer 1 bedroom, hall, and kitchen and the rest prefer to choose their house by contacting agent and builder. Bachelors and family do not prefer to contact with builder so there is no data about it.

4.699% of the bachelors prefer to choose house with 2 bedrooms, hall, and kitchen by contacting owner and 3.328% of them prefer to choose house with 3 bedrooms, hall, and kitchen by contacting agent.

2.76% of the family prefer to choose house with 2 bedrooms, hall, and kitchen by contacting owner and 2.066% of them prefer to choose house with 3 bedrooms, hall, and kitchen.

**Findings**

- More bachelors, bachelors/family and family prefer houses with 2 bedrooms, hall, and kitchen by contacting owner.
- Slightly more family prefer houses with 3 bedrooms, hall, and kitchen in by contacting agent.

## Analysis 3-9: Find the Relationship between Bedroom_Hall_Kitchen, Number_of_Bathroom and City

This analysis is conducted to investigate how will tenant choose their house based on Bedroom_Hall_Kitchen, Number_of_Bathroom and City.

```
# Calculate Percentage Grouped by Tenant_Type, Bedroom_Hall_Kitchen, Number_of_Bathroom and City
group_tt_bhk_nob_c <- house_rental_data %>%
  group_by(Tenant_Type,Bedroom_Hall_Kitchen,Number_of_Bathroom,City) %>%
  summarise(number_cases=n())

# Bubble Plot
ggplot(group_tt_bhk_nob_c,aes(x=Bedroom_Hall_Kitchen,y=Number_of_Bathroom,color=number_cases)) +
  geom_point(aes(size=number_cases),alpha=0.5) +
  labs(x="Number of Bedroom, Hall and Kitchen",y="Number of Bathroom",
       title="Relationship between Number of Bedroom, Hall, Kitchen, Number of Bathroom and City") +
  scale_x_continuous(breaks=seq(1,6,1)) +
  scale_y_continuous(breaks=seq(1,10,1)) +
  facet_grid(Tenant_Type~City) +
  theme(legend.position = "bottom")
```

*Figure 6.3.18: Source Code – Bubble Plot to Show the Relationship between Bedroom_Hall_Kitchen, Number_of_Bathroom and City*

**Analysis Technique: Data Visualization and Manipulation**

Figure 6.3.18 depicts the source code used to create the percentage of tenant choose house based on number of Bedroom_Hall_Kitchen, Number_of_Bathroom and City. The number of tenants choose house based on this relationship is calculated using **group_by()** and **summarise()**. Then use **mutate()** to sum up the number using **sum()** and assign to variable "number_cases"

The next source code is used to generate a bar chart to study the distribution of tenant based on Bedroom_Hall_Kitchen and Area_Type. **geom_text()** is used to add text directly to the graph. The position with **position_dodge()** in geom_text() put the bars side-by-side and the alignment of the text can be adjusted by width argument. **labs()** is used to modify the names of the axes and the title plot. **scale_x_continuous()** is the position scales for continuous data of x-axis. The x-axis label is modified so that it produce 1 to 6 with 1 space between them in the graph and it is referred to breaks. **facet_wrap()** is utilised to generate graphics tables that show the same graph for each group of city. The **theme()** is used to customize the non-data components such as the position of the legend to bottom and set the background of the legend to antiquewhite.

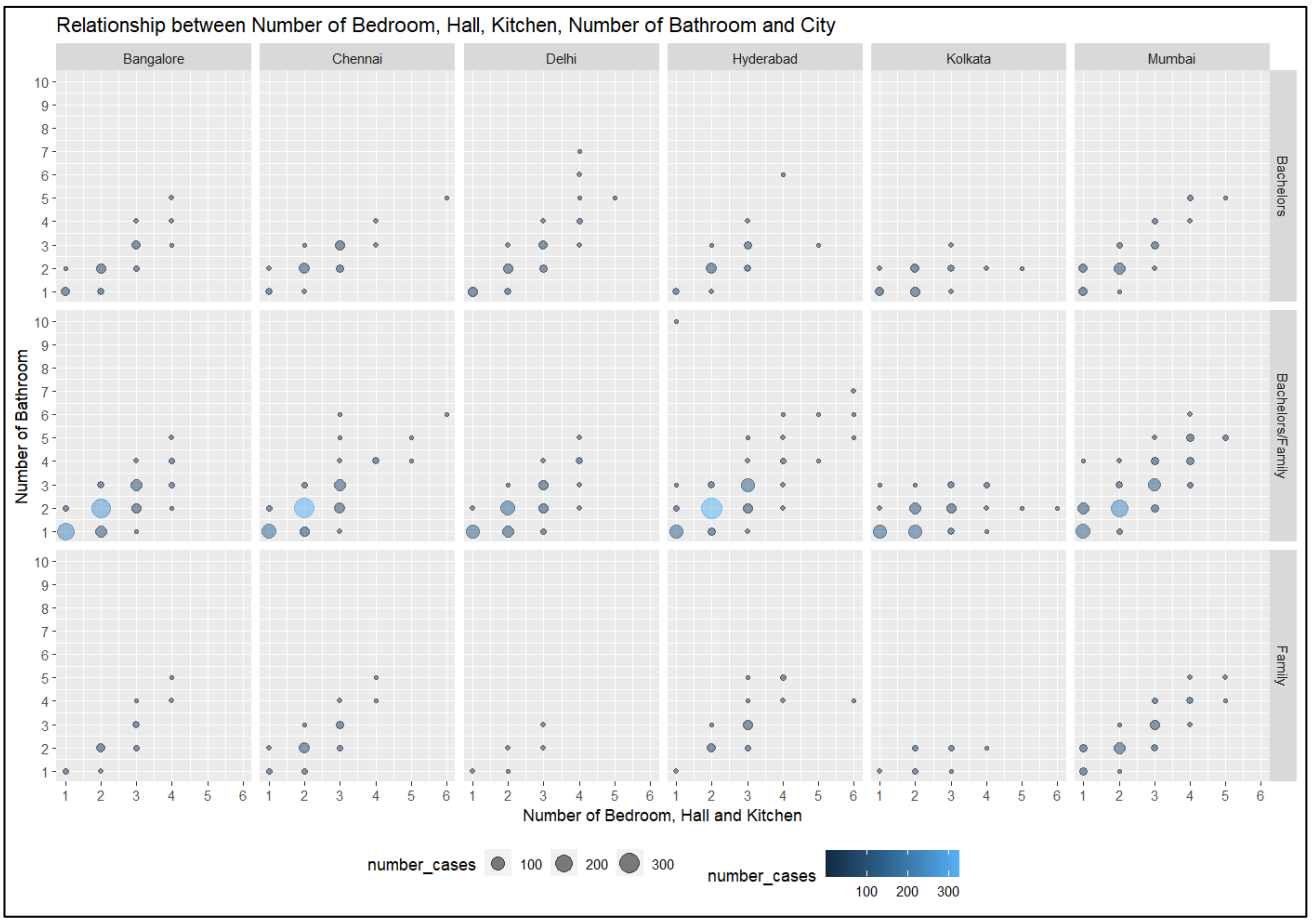



*Figure 6.3.19: Output – Bubble Plot (Show the Relationship between Bedroom_Hall_Kitchen, Number_of_Bathroom and City)*

**Explanation**

The most obvious point from Figure 6.3.19 is that bachelors/family mostly choose 2 bathrooms and 2 bedrooms, hall, and kitchen in the cities Bangalore, Chennai, Hyderabad and Mumbai, whereas some of them also prefer to choose 1 bathroom and 1 bedroom, hall, and kitchen.

Bachelors also prefer to choose 2 bathrooms and 2 bedrooms, hall, and kitchen according to the size of the point in Figure 6.3.19 whereas family prefer to choose either 2 bedrooms, hall, and kitchen with 2 bathrooms or 3 bedrooms, hall, and kitchen with 3 bathrooms in Hyderabad and Mumbai

250 of the bachelors prefer to choose 2 bedrooms, hall, and kitchen with 2 bathrooms, whereas the rest of them choose houses with different number of bathrooms, number of bedrooms, hall, and kitchen.

**Findings**

- Family tend to choose their preference house with more bathrooms, bedrooms, hall, and kitchen in Mumbai

- Bachelors/family tend to choose their preference house with 2 bathrooms, 2 bedrooms, hall, and kitchen in Bangalore, Chennai, Hyderabad and Mumbai.

**Conclusion for Question 3**

1. More tenants prefer to choose houses with 2 bathrooms, 2 bedrooms, hall, and kitchen.
2. More tenants prefer 1 and 2 bedrooms, hall, and kitchen which located at floor level than 5.
3. More tenants choose houses with house size of more than 2000 sqft and 4, 5 and 6 bedrooms, hall, and kitchen.
4. Within the similar range of RM20,000, more tenants prefer to choose house with 2 bedrooms, hall, and kitchen.
5. More tenants prefer to choose houses that are semi-furnished with 2 bedrooms, hall, and kitchen.
6. More tenants prefer to choose houses that are located at carpet area and super area with 2 bedrooms, hall, and kitchen.
7. More tenants prefer to choose house that are located at Bangalore, Chennai and Hyderabad with 2 bedrooms, hall, and kitchen.
8. More tenants prefer to contact owner to choose their houses with 2 bedrooms, hall, and kitchen.
9. More tenants prefer to choose house with 2 bathrooms, 2 bedrooms, hall, and kitchen in Mumbai.

## Question 4: What are the Factors influencing Tenants to Choose Their Houses with respect to Number of Bathroom?

### Analysis 4-1: Find the Relationship between Number_of_Bathroom and Floor_Preference

This analysis is conducted to investigate which number of bathroom and which floor will have more tenants prefer to choose.

```
# Box Plot
ggplot(house_rental_data,aes(x=Number_of_Bathroom,y=Floor_Preference,color=factor(Number_of_Bathroom))) +
  geom_boxplot(aes(x=Number_of_Bathroom,y=Floor_Preference,group=Number_of_Bathroom)) +
  labs(x="Number of Bathroom",y="Floor Preference") +
  scale_x_continuous(breaks=seq(1,10,1)) +
  scale_y_continuous(breaks=seq(-2,80,2)) +
  guides(color=guide_legend("Number of Bathroom",nrow=1)) +
  ggtitle("Relationship between Number of Bathroom and Floor Preference") +
  theme(legend.position = "bottom") +
  facet_wrap(~Tenant_Type)
```

*Figure 6.4.1: Source Code – Box Plot to Show the Relationship between Number_of_Bathroom and Floor_Preference*

**Analysis Technique: Data Visualization**

Figure 6.4.1 depicts the source code used to generate a box plot to study the distribution of tenant based on Number_of_Bathroom and Floor_Preference. **geom_boxplot()** is used to create box plot graph. **labs()** is used to modify the names of the axes label and plot title. **labs()** is used to modify the axes label names. **scale_x_continuous()** is the position scales for continuous data of x-axis. The x-axis label is modified so that it produce 1 to 10 with 1 space between them in the graph and it is referred to breaks. The **guides()** is used to provide guides for each scale and the **guide_legend(nrow=1)** is used to customize the legend to show in one horizontal row. The title name is set using **ggtitle()**.The **theme()** is used to customize the non-data components such as the position of the legend. **facet_wrap()** is utilised to generate graphics tables that show the same graph for each group of tenants.

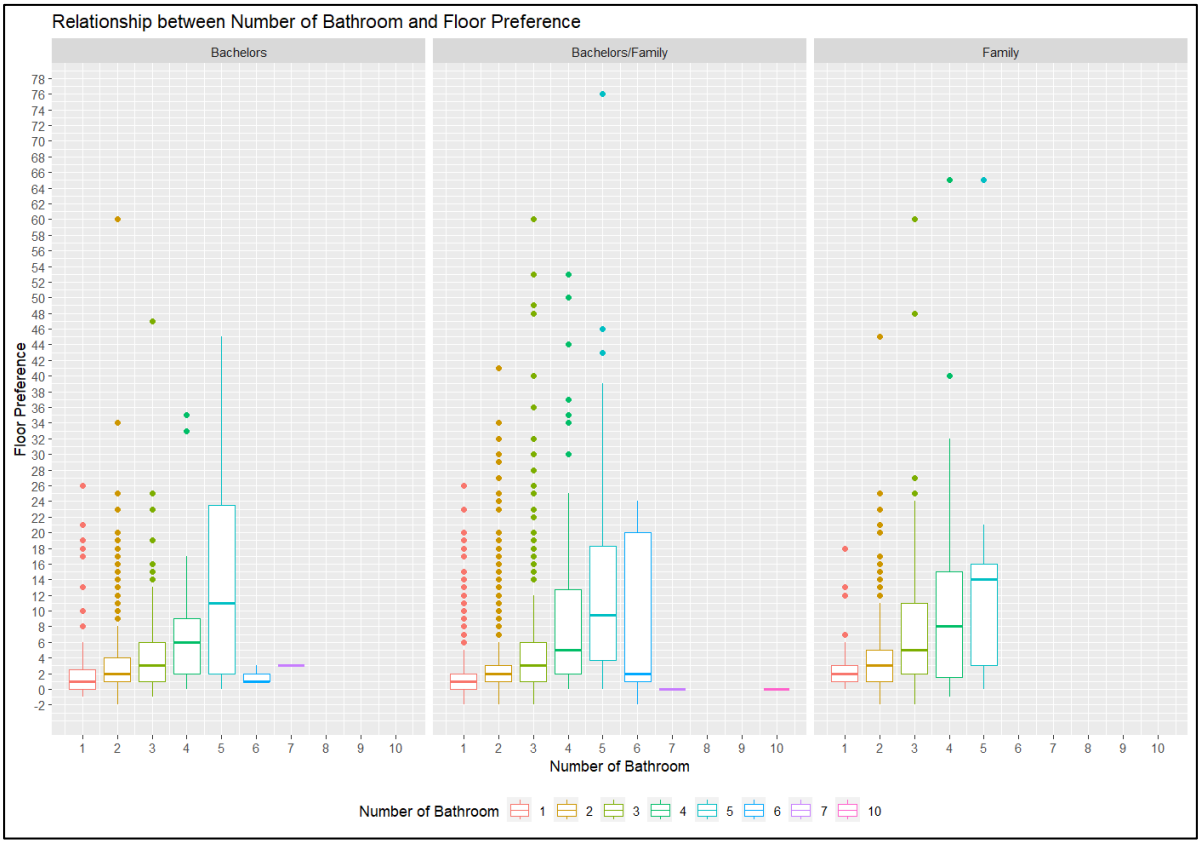



*Figure 6.4.2: Output – Box Plot (Relationship between Number_of_Bathroom and Floor_Preference)*

**Explanation**

Figure 6.4.2 suggests that the floor that bachelor choose house with 1 bathroom is within the upper basement and floor 3. Bachelor choose house with 2 bathroom is within lower basement and floor 4. Bachelor choose house with 3 bathroom is within upper basement and floor 6. If they want to choose house with 4 bathrooms, then the house is located between ground floor and floor 9. House with 5 bathrooms is quite common as it can be found in floor between ground floor and floor 23. However, house with greater than 5 bathrooms is not included in the preference of bachelors.

Bachelors/family choose house with 1 bathroom is within the lower basement and floor 2 and house with 2 bathrooms is within lower basement and floor 4. House with 3 bathrooms is available between lower basement and floor 7 and house with 4 bathrooms is available between first floor and floor 13. House with 5 bathrooms and 6 bathrooms is quite common for bachelors/family to choose between the ground floor and floor 18 and between lower basement and floor 20. Similar to bachelors, house with greater than 6 bathrooms is not included in the preference of bachelors/family.

The floor which family prefer to have house with 1 bathroom is between ground floor and floor 3. House with 2 bathrooms and 3 bathrooms is located between lower basement and floor 5 and between lower basement and floor 11. House with 4 bathrooms can be found between upper basement and floor 15 whereas house with 5 bathrooms is available between ground floor and floor 15.

**Findings**

- More bachelors prefer to rent house located at the lower floor with 5 bathrooms
- More bachelors/family prefer to rent house located at the middle floor with either 4 or 5 or 6 bathrooms.
- More family prefer to rent house located at the lower floor with 4 or 5 bathrooms

## Analysis 4-2: Find the Relationship between Number_of_Bathroom and House_Size

This analysis is conducted to find out how tenant choose their house based on number of bathroom and the size of the house..

/**

** Following source code obtained from (Datavizpyr, 2020)*

*/

```
# Calculate Mean of House_Size Grouped by Tenant_Type and Number_of_Bathroom
group_tt_nob_hs <- house_rental_data %>% group_by(Tenant_Type,Number_of_Bathroom) %>%
  summarise(avg_house_size = mean(House_Size))

# Lollipop Graph
ggplot(group_tt_nob_hs,aes(x=Number_of_Bathroom,y=avg_house_size)) +
  geom_point(size=3,colour="black") +
  geom_segment(aes(x=Number_of_Bathroom,xend=Number_of_Bathroom,y=0,yend=avg_house_size)) +
  geom_text(aes(Number_of_Bathroom,avg_house_size,label=signif(avg_house_size,2),vjust=-0.6)) +
  scale_x_continuous(breaks=seq(1,10,1)) +
  labs(x="Number of Bathroom",y="Size of House",
       title="Relationship between Number of Bathroom and House Size") +
  facet_wrap(~Tenant_Type)
```

*Figure 6.4.3: Source Code – Lollipop Graph to show the Relationship between Number_of_Bathroom and House_Size*

**Analysis Technique: Data Visualization and Manipulation**

Figure 6.4.3 depicts the source code used to create the graph of tenant choose house based on Number_of_Bathroom and House_Size. The number of tenants choose house based on this relationship is calculate using **group_by()** and **count()**. Then calculate the mean of house size, assign it to variable named "avg_house_size" and summarise it using **summarise()**.

The next source code is used to generate a lollipop graph to study how number of bathrooms varies the house size. **geom_point()** is used to create point graph. **geom_segment()** is used to draw a straight line between two points. **labs()** is used to modify the names of the axes label and plot title. **scale_x_continuous()** is the position scales for continuous data of x-axis. The x-axis label is modified so that it produce 1 to 10 with 1 space between them in the graph and it is referred to breaks. **facet_wrap()** is utilised to generate graphics tables that show the same graph for each group of tenants.

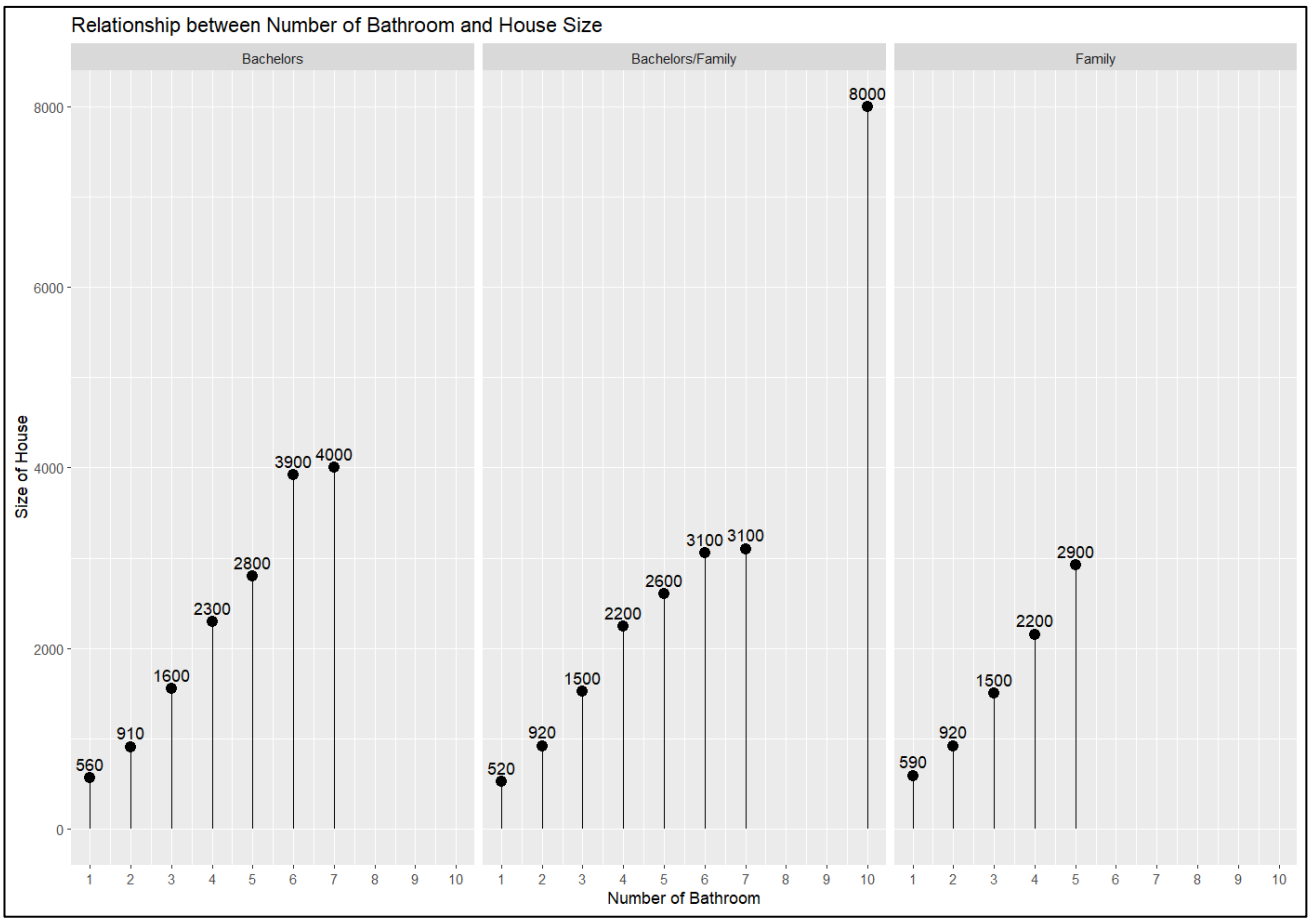



*Figure 6.4.4: Output – Lollipop Graph (Relationship between Number_of_Bathroom and House_Size)*

**Explanation**

Figure 6.4.4 shows an increase trend. When the number of bathroom increase, the house size also increase. This is also said that the number of bathrooms vary the house size.

**Findings**

- The minimum average house size that bachelors can rent a house with 1 bathroom is 560 sqft and maximum average house size that they can rent a house with 7 bathrooms is 4000 sqft
- The minimum average house size that bachelors/family can rent a house with 1 bathroom is 520 sqft and maximum average house size that they can rent house with 7 bathrooms is 3100 sqft.
- The minimum average house size that family can rent a house with 1 bathroom is 880 and the maximum average house size that they can rent house with 5 bathrooms is 2800.
- None of the family choose number of bathrooms greater than 5.

## Analysis 4-3: Find the Relationship between Number_of_Bathroom and Rental_Fee

```
# Box Plot
ggplot(rf_no_outliers,aes(x=Number_of_Bathroom,y=Rental_Fee)) +
  geom_boxplot(aes(x=factor(Number_of_Bathroom),y=Rental_Fee,color=factor(Number_of_Bathroom))) +
  facet_wrap(~Tenant_Type) +
  labs(x= "Number of Bathroom",y="Rental Fee",
       title="Relationship between Number of Bathroom and Rental Fee") +
  scale_y_continuous(labels=scales::comma) +
  theme(legend.position="bottom") +
  scale_color_discrete(name="Number of Bathroom")
```

*Figure 6.4.5: Source Code – Box Plot to Show the Relationship between Number_of_Bathroom and Rental_Fee*

**Analysis Technique: Data Visualization**

Figure 6.4.5 depicts the source code used to generate a box plot to study the distribution of tenant based on Number_of_Bathroom and Floor_Preference. **geom_boxplot()** is used to create box plot graph. **facet_wrap()** is utilised to generate graphics tables that show the same graph for each group of tenants. **labs()** is used to modify the names of the axes label, the name of the legend and plot title. **labs()** is used to modify the axes label names. **scale_y_continuous()** is the position scales for continuous data of y-axis. The y-axis label is modified so that it do not produce e notation in the y-axis label and it is referred to labels. The **theme()** is used to customize the non-data components such as the position of the legend. **scale_color_discrete()** is used to modify the legends in the way that the name of the legends can be modified.

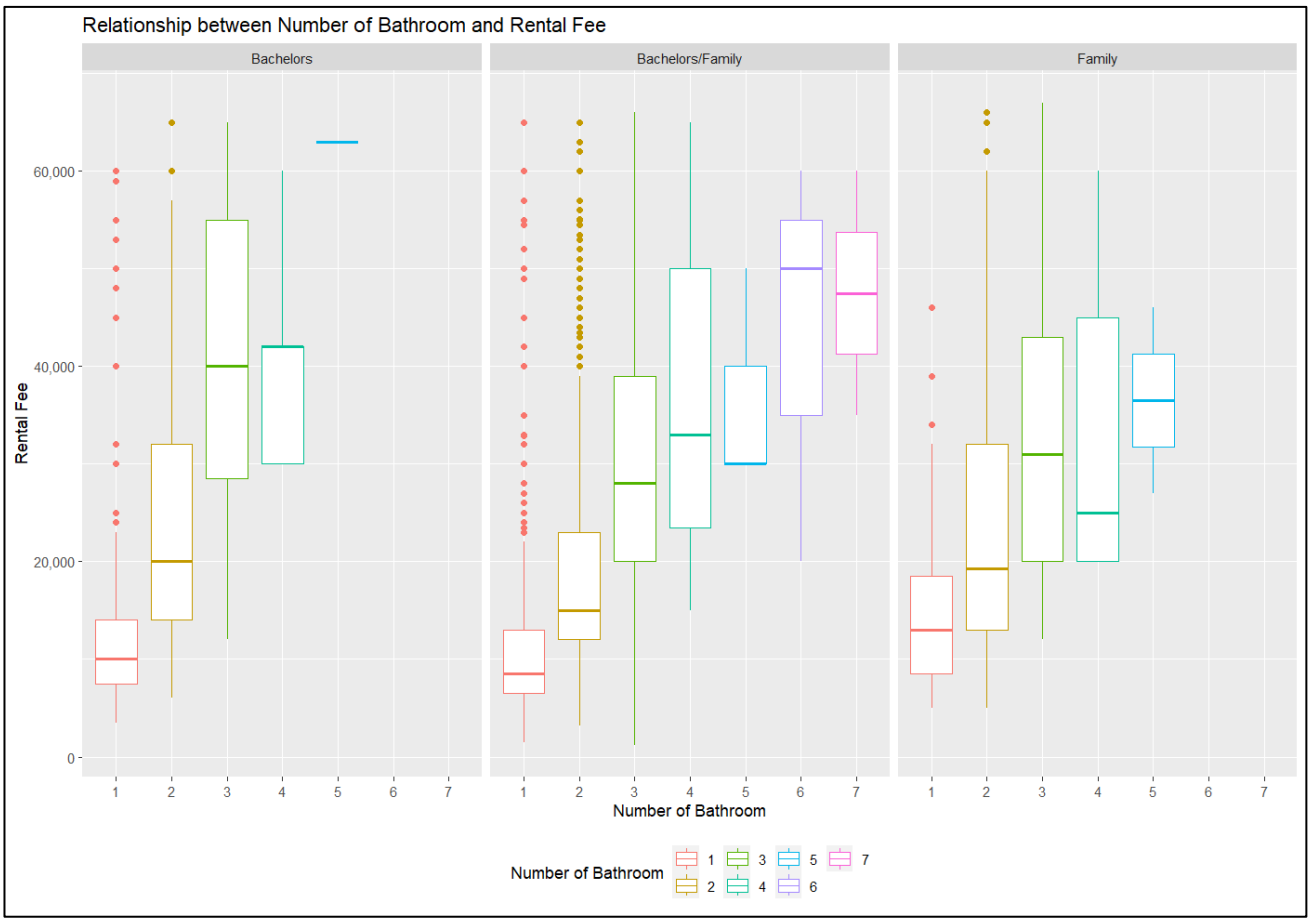



*Figure 6.4.6: Output – Box Plot (Relationship between Number_of_Bathroom and Rental Fee)*

**Explanation**

Figure 6.4.6 shows an increasing trend of rental fee when number of bathrooms is increasing. Bachelors, bachelors/family, and family prefer to rent a house with 1 bathroom below total rental fee of RM20,000. Bachelors and family rent house with 2 bathrooms below RM30,000, bachelors/family rent house with 2 bathrooms below around RM22,000. The rest prefer to rent the house with more bathrooms at higher rental fee.

**Findings**

- Bachelors and bachelors/family prefer to rent houses with 2 bathrooms at a lower total rental fee.
- Some bachelors/family prefer to rent houses with 6 and 7 bathrooms at higher total rental fee.
- Family prefer houses with 6 bedrooms, hall, and kitchen with lower floor

## Analysis 4-4: Find the Relationship between Number_of_Bathroom and Furnishing_Status

```
# Calculate Percentage Grouped By Tenant_Type, Number_of_Bathroom and Furnishing_Status
group_tt_nob_fs <- house_rental_data %>%
  group_by(Tenant_Type,Number_of_Bathroom,Furnishing_Status) %>%
  count() %>% ungroup() %>% mutate(perc=n/sum(n)) %>% arrange(perc) %>%
  mutate(labels=scales::percent(perc))

# Bar Chart
ggplot(group_tt_nob_fs,aes(factor(Number_of_Bathroom),perc,fill=factor(Number_of_Bathroom))) +
  geom_bar(stat="identity",position="dodge") +
  geom_text(aes(label=labels),size=3,vjust=-0.2,
                position=position_dodge(width=0.9))+
  labs(x="Number of Bathroom",y="Percentage",
       title="Relationship between Number of Bathroom and Furnishing Status") +
  scale_y_continuous(labels=scales::percent) +
  scale_fill_discrete(name="Number of Bathroom") +
  theme(legend.position="bottom",
        legend.background = element_rect(fill="antiquewhite")) +
  facet_grid(Tenant_Type~Furnishing_Status)
```

*Figure 6.4.7: Source Code – Bar Chart to Show the Relationship between Number_of_Bathroom and Furnishing_Status*

**Analysis Technique: Data Visualization and Manipulation**

Figure 6.4.7 depicts the source code used to create the percentage of tenant choose house based on Number_of_Bathroom and Furnishing_Status. The number of tenants choose house based on this relationship is calculated using **group_by()** and **count()**. Then the percentage is calculated by finding the total numbers using **sum()** and use **mutate()** to calculate the percentage and assigned it to variable "perc". The percentage is then arranged in ascending using **arrange()**. In order to show the percentage symbol in the graph, **scales::percent()** is used and assigned it to labels.

The next source code is used to generate a bar chart to study the distribution of tenant based on Number_of_Bathroom and Furnishing_Status. **geom_text()** is used to add text directly to the graph. The position with **position_dodge()** in geom_text() put the bars side-by-side and the alignment of the text can be adjusted by width argument. **labs()** is used to modify the names of the axes and the title plot. **scale_y_continuous()** is the position scales for continuous data of y-axis. labels in y-axis can be modified to percentage using **scale_y_continuous(labels=scales::percent)**. **scale_fill_discrete()** is used to modify the legends in the way that the name of the legends can be modified. The **theme()** is used to customize the non-data components such as the position of the legend to bottom and set the background of the legend to antiquewhite. **facet_grid()** generates a 2D grid of panels with the rows and columns given by Tenant_Type and Furnishing_Status**.**

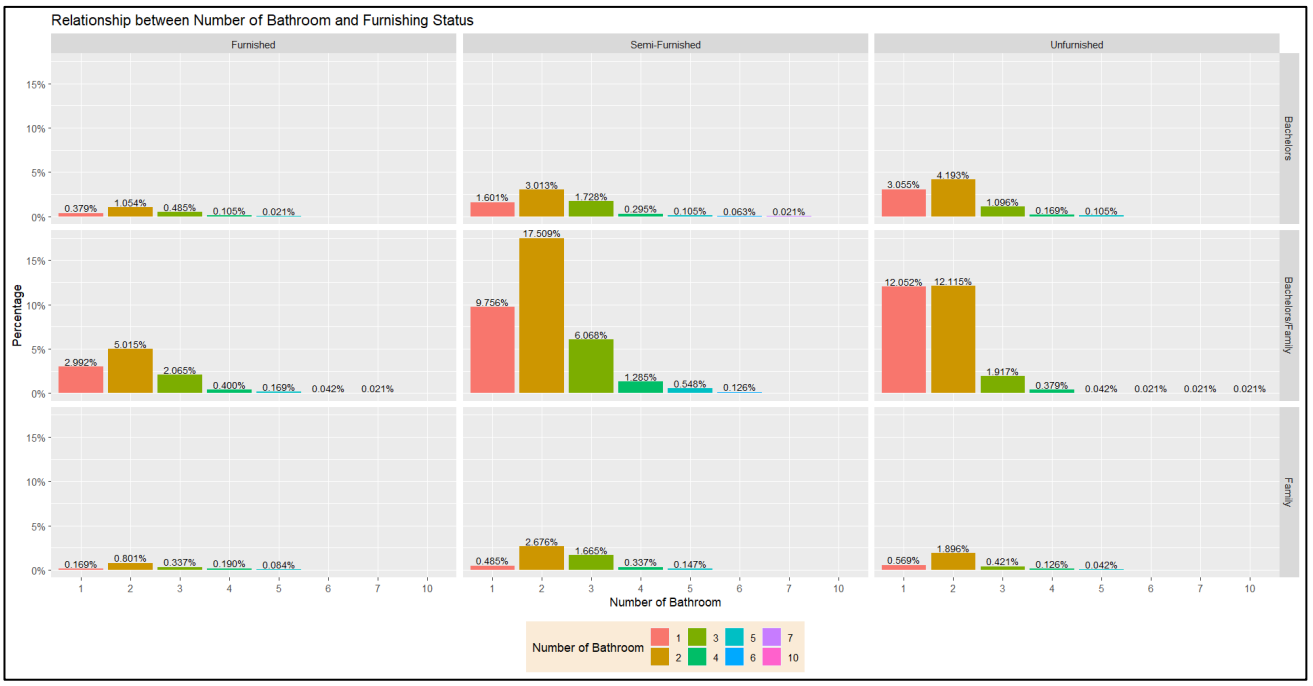



*Figure 6.4.8: Output – Bar Chart (Relationship between Number_of_Bathroom and Furnishing_Status)*

**Explanation**

Figure 6.4.8 illustrate that most of the tenants prefer to choose houses with 2 bathrooms but with different furnishing status

4.193% of the bachelors prefer unfurnished houses, 3.013% of them prefer semi-furnished houses whereas 1.054% of them prefer furnished house.

For bachelors/family site, 17.509% of them prefer semi-furnished houses, 12.115% of them prefer unfurnished houses and 5.015% of them prefer furnished houses.

2.676% of the family prefer semi-furnished houses, 1.896% of them prefer unfurnished and 0.801% prefer furnished houses.

**Findings**

- More bachelors prefer unfurnished houses with 2 bathrooms
- More bachelors/family and family prefer semi-furnished houses with 2 bathrooms

## Analysis 4-5: Find the Relationship between Number_of_Bathroom and Area_Type

This analysis is conducted to find out tenants prefer houses with how many bathrooms in each area type.

```
# Calculate Percentage Grouped by Tenant_Type, Number_of_Bathroom and Area_Type
group_tt_nob_at <- house_rental_data %>%
  group_by(Tenant_Type,Number_of_Bathroom,Area_Type) %>%
  count() %>% ungroup() %>% mutate(perc=n/sum(n)) %>% arrange(perc) %>%
  mutate(labels=scales::percent(perc))

# Bar Chart
ggplot(group_tt_nob_at,aes(Number_of_Bathroom,perc,fill=Area_Type)) +
  geom_bar(stat="identity",position="dodge") +
  geom_text_repel(aes(label=labels,group=Area_Type),
                  size=3,position=position_dodge(width=0.9),
                  max.overlaps = 30) +
  labs(x="Number of Bathroom",y="Percentage",
       title="Relationship between Number of Bathroom and Area Type") +
  scale_x_continuous(breaks=seq(1,10,1)) +
  scale_y_continuous(labels=scales::percent) +
  scale_fill_discrete(name="Area Type") +
  theme(legend.position="bottom",
        legend.background = element_rect(fill="antiquewhite")) +
  facet_wrap(~Tenant_Type)
```

*Figure 6.4.9: Source Code – Bar Chart to Show the Relationship between Number_of_Bathroom and Area_Type*

**Analysis Technique: Data Visualization and Manipulation**

Figure 6.4.9 depicts the source code used to create the percentage of tenant choose house based on Number_of_Bathroom and Area_Type. The number of tenants choose house based on this relationship is calculated using **group_by()** and **count()**. Then the percentage is calculated by finding the total numbers using **sum()**, dividing it with the counted number and use **mutate()** to calculate the percentage and assigned it to variable "perc". The percentage is then arranged in ascending using **arrange()**. In order to show the percentage symbol in the graph, **scales::percent()** is used and assigned it to labels.

The next source code is used to generate a bar chart to study the distribution of tenant based on Number_of_Bathroom and Area_Type. **geom_text()** is used to add text directly to the graph. The position with **position_dodge()** in geom_text() put the bars side-by-side and the alignment of the text can be adjusted by width argument. **labs()** is used to modify the names of the axes and the title plot. **scale_x_continuous()** is the position scales for continuous data of x-axis, while **scale_y_continuous()** is the position scales for continuous data of y-axis. The x-axis label is modified so that it produce 1 to 10 with 1 space between them in the graph and it is referred to breaks. labels in y-axis can be modified to percentage using **scale_y_continuous(labels=scales::percent)**. **scale_fill_discrete()** is used to modify the legends in the way that the name of the legends can be modified. The **theme()** is used to customize the non-data components such as the position of the legend to bottom and set the background of the legend to antiquewhite. **facet_wrap()** is utilised to generate graphics tables that show the same

graph for each group of tenants.

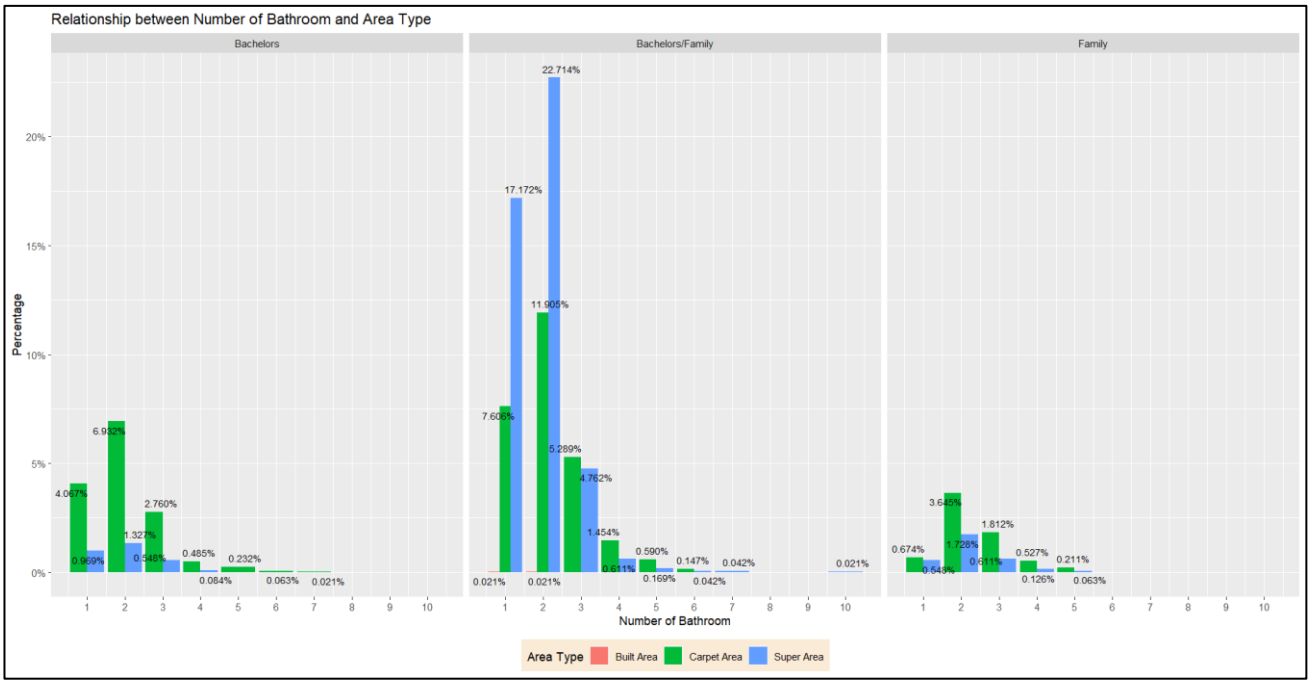



*Figure 6.4.10: Output – Bar Chart (Relationship between Number_of_Bathroom and Area_Type)*

**Explanation**

The most obvious that can be seen clearly in Figure 6.4.10 is that bachelors/family prefer houses with 2 bathrooms to be located in the super area. There is 22.714% of them, followed by 17.172% with 1 bathroom. 11.905% of them prefer to choose houses with 2 bathrooms located in the carpet area.

Both bachelors and family more prefer to choose houses that is located in the carpet area. There is 6.912% of the bachelors who prefer to choose houses with 2 bathrooms, followed by 2.067% with 1 bathroom and 2.76% with 3 bathrooms, whereas there is 3.645% of the family prefer to choose house with 2 bathrooms, followed by 1.812% with 3 bathrooms. 1.728% of them prefer houses with 2 bathrooms located in super area. None of them prefer houses that is located in the built area.

**Findings**

- More bachelors and family prefer houses with 2 bathrooms located in carpet area.
- More bachelors/family prefer houses with 2 bathrooms located in super area

## Analysis 4-6: Find the Relationship between Number_of_Bathroom and City

```
# Calculate Percentage Grouped by Tenant_Type, Number_of_Bathroom and City
group_tt_nob_c <- house_rental_data %>%
  group_by(Tenant_Type,Number_of_Bathroom,City) %>%
  count() %>% ungroup() %>% mutate(perc=n/sum(n)) %>% arrange(perc) %>%
  mutate(labels=scales::percent(perc))

# Bar Chart
ggplot(group_tt_nob_c,aes(Number_of_Bathroom,perc,fill=Tenant_Type)) +
  geom_bar(stat="identity",position="dodge") +
  geom_text_repel(aes(label=labels,group=City),
                  size=3, position=position_dodge(width=0.9),
                  max.overlaps=30) +
  labs(x="Number of Bathroom",y="Percentage",
       title="Relationship between Number of Bathroom and City") +
  scale_x_continuous(breaks=seq(1,10,1)) +
  scale_y_continuous(labels=scales::percent) +
  scale_fill_discrete(name="Tenant Type") +
  theme(legend.position="bottom",
        legend.background = element_rect(fill="antiquewhite")) +
  facet_wrap(~City)
```

*Figure 6.4.11:Source Code – Bar Chart to Show the Relationship between Number_of_Bathroom and City*

**Analysis Technique: Data Visualization and Manipulation**

Figure 6.4.11 depicts the source code used to create the percentage of tenant choose house based on Number_of_Bathroom and City. The number of tenants choose house based on this relationship is calculated using **group_by()** and **count()**. Then the percentage is calculated by finding the total numbers using **sum()**, dividing it with the counted number and use **mutate()** to calculate the percentage and assigned it to variable “perc”. The percentage is then arranged in ascending using **arrange()**. In order to show the percentage symbol in the graph, **scales::percent()** is used and assigned it to labels.

The next source code is used to generate a bar chart to study the distribution of tenant based on Number_of_Bathroom and City. **geom_text_repel()** is used to add text directly to the graph and avoid overlapping labels. The position with **position_dodge()** in geom_text_repel() put the bars side-by-side and the alignment of the text can be adjusted by width argument. The **max.overlaps** in geom_text_repel is used to set the maximum overlapping that the label can overlap. **labs()** is used to modify the names of the axes and the title plot. **scale_x_continuous()** is the position scales for continuous data of x-axis, while **scale_y_continuous()** is the position scales for continuous data of y-axis. The x-axis label is modified so that it produce 1 to 10 with 1 space between them in the graph and it is referred to breaks. labels in y-axis can be modified to percentage using **scale_y_continuous(labels=scales::percent)**. **scale_fill_discrete()** is used to modify the legends in the way that the name of the legends can be modified. The **theme()** is used to customize the non-data components such as the position of the legend to bottom and set the background of the legend to antiquewhite. **facet_wrap()** is utilised to generate graphics tables that show the same graph for each group of cities.

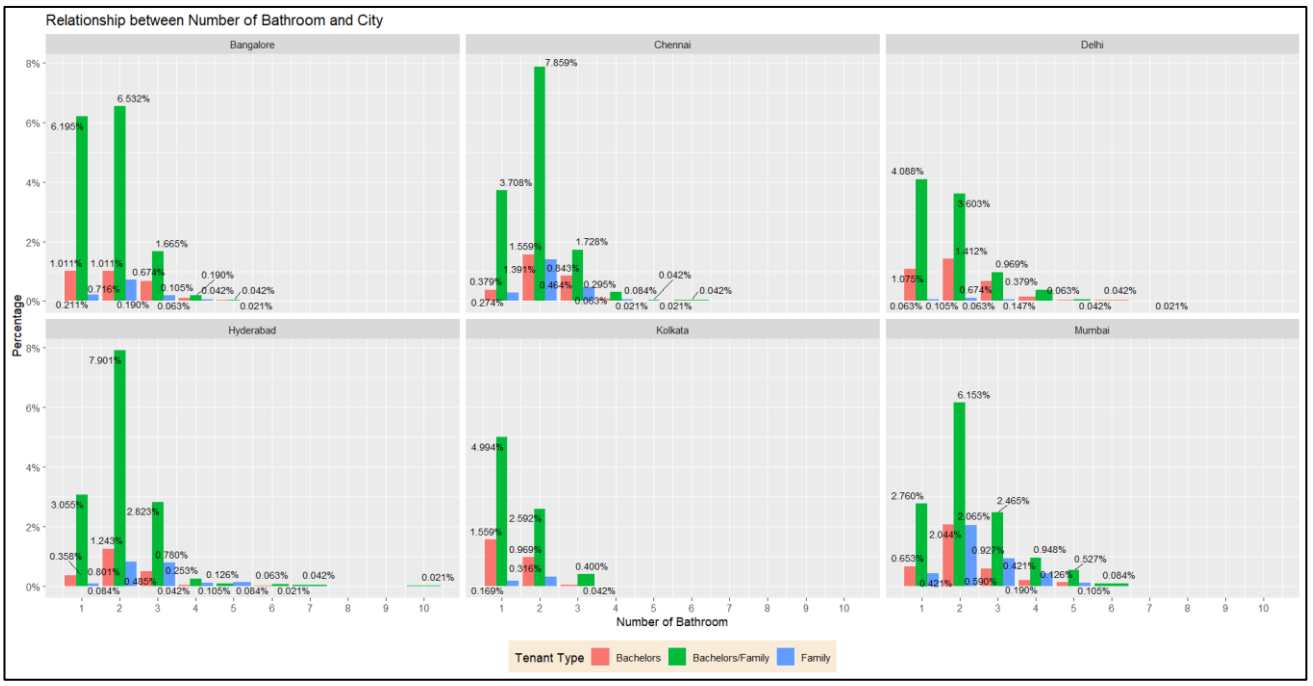



*Figure 6.4.12: Output – Bar Chart (Relationship between Number_of_Bathroom and City)*

**Explanation**

The most obvious that can be seen clearly in Figure 6.4.12 is that bachelors/family prefer houses with 2 bathrooms to be located in Chennai and Hyderabad. There is 7.859% of them in Chennai and 7.901% of them in Hyderabad.

2.067% of the bachelors prefer houses with 2 bathrooms in Mumbai, followed by 2.760% of them prefer houses with 1 bathroom. 1.559% of the bachelors prefer houses with 1 bathroom in Kolkata and houses with 2 bathrooms in Chennai.

2.044% of the family prefer houses with 2 bathrooms in Mumbai, followed by 1.391% of them prefer houses with 2 bathrooms in Chennai.

**Findings**

- More bachelors and family prefer houses with 2 bathrooms located in Mumbai.
- Slightly more bachelors/family prefer houses with 2 bathrooms located in Hyderabad than in Chennai

## Analysis 4-7: Find the Relationship between Number_of_Bathroom and Point_of_Contact

This analysis is conducted to investigate how many percent of the tenants prefer to contact to with the number of bathrooms.

```
# calculate Percentage Grouped By Tenant_Type, Number_of_Bathroom and Point_of_Contact
group_tt_nob_poc <- house_rental_data %>%
  group_by(Tenant_Type,Number_of_Bathroom,Point_of_Contact) %>%
  count() %>% ungroup() %>% mutate(perc=n/sum(n)) %>% arrange(perc) %>%
  mutate(labels=scales::percent(perc))
# Bar Chart
ggplot(group_tt_nob_poc,aes(x=Number_of_Bathroom,y=perc,fill=Point_of_Contact)) +
  geom_bar(stat="identity",position="dodge") +
  geom_text(aes(label=labels),size=3,vjust=-0.3,position=position_dodge(width=0.9)) +
  labs(x="Number of Bathroom",y="Percentage",
       title="Relationship between Number of Bathroom and Point_of_Contact") +
  scale_x_continuous(breaks=seq(1,10,1)) +
  scale_y_continuous(labels=scales::percent) +
  scale_fill_discrete(name="Point of Contact") +
  theme(legend.position="bottom",
        legend.background = element_rect(fill="antiquewhite")) +
  facet_grid(Point_of_Contact~Tenant_Type)
```

*Figure 6.4.13: Source Code – Bar Chart to Show the Relationship between Number_of_Bathroom and Point_of_Contact*

**Analysis Technique: Data Visualization and Manipulation**

Figure 6.4.13 depicts the source code used to create the percentage of tenant choose house based on Number_of_Bathroom and Point_of_Contact. The number of tenants choose house based on this relationship is calculated using **group_by()** and **count()**. Then the percentage is calculated by finding the total numbers using **sum()**, dividing it with the counted number and use **mutate()** to calculate the percentage and assigned it to variable "perc". The percentage is then arranged in ascending using **arrange()**. In order to show the percentage symbol in the graph, **scales::percent()** is used and assigned it to labels.

The next source code is used to generate a bar chart to study the distribution of tenant based on Number_of_Bathroom and Point_of_Contact. **geom_text()** is used to add text directly to the graph. The position with **position_dodge()** in geom_text_repel() put the bars side-by-side and the alignment of the text can be adjusted by width argument. **labs()** is used to modify the names of the axes and the title plot. **scale_x_continuous()** is the position scales for continuous data of x-axis, while **scale_y_continuous()** is the position scales for continuous data of y-axis. The x-axis label is modified so that it produce 1 to 10 with 1 space between them in the graph and it is referred to breaks. labels in y-axis can be modified to percentage using **scale_y_continuous(labels=scales::percent)**. **scale_fill_discrete()** is used to modify the legends in the way that the name of the legends can be modified. The **theme()** is used to customize the non-data components such as the position of the legend to bottom and set the background of the legend to antiquewhite. **facet_grid()** generates a 2D grid of panels with the rows and columns given by Point_of_Contact and Tenant_Type.

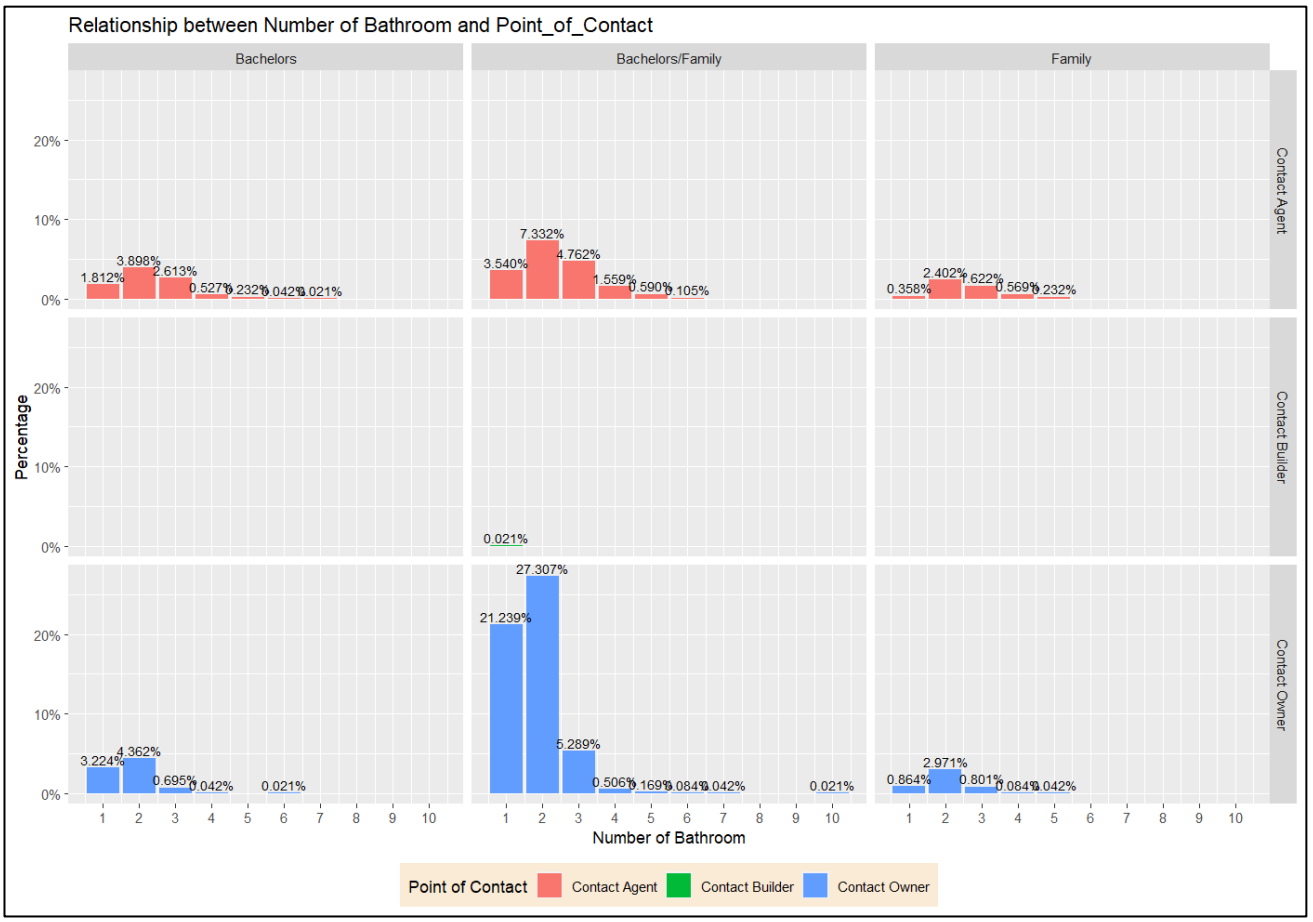



*Figure 6.4.14: Output – Bar Chart (Relationship between Number_of_Bathroom and Point_of_Contact)*

**Explanation**

The most obvious that can be seen clearly in Figure 6.4.14 is that 27.207% of the bachelors/family prefer to choose their houses with 2 bathrooms via contacting owner, followed by 21.239% of them contact owner for houses with 1 bathroom. 7.332% of them prefer to contact agent for houses with 2 bathrooms.

4.362% of the bachelors also prefer contact owner for houses with 2 bathrooms, followed by3.224% of them contact owner for houses with 1 bathroom, whereas 3.8998% of them prefer to contact agent for houses with 2 bathrooms.

2.971% of the family prefer contacting owner for houses with 2 bathrooms, whereas 2.402% of them prefer contacting agent for houses with 2 bathrooms.

**Findings**

- More bachelors, bachelors/family and family prefer houses with 2 bathrooms by contacting owner.

## Analysis 4-8: Find the Relationship between Number_of_Bathroom, Floor_Preference and City

This analysis is conducted to observe tenants prefer houses with how many bathrooms, which floor and in which city.

```
# Box Plot
ggplot(house_rental_data,aes(x=factor(Number_of_Bathroom),y=Floor_Preference,
                             color=factor(Number_of_Bathroom))) +
  geom_boxplot(aes(x=factor(Number_of_Bathroom),y=Floor_Preference)) +
  labs(x="Number of Bathroom",y="Floor Preference", color="Number of Bathroom",
       title="Relationship between Number_of_Bathroom, Floor_Preference and City") +
  facet_grid(Tenant_Type~City)
```

*Figure 6.4.15: Source Code – Box Plot to Show the Relationship between Number_of_Bathroom, Floor_Preference and City*

**Analysis Technique: Data Visualization**

Figure 6.4.15 depicts the source code used to generate a box plot to study the distribution of tenant based on Number_of_Bathroom, Floor_Preference and City. **geom_boxplot()** is used to create box plot graph. **labs()** is used to modify the names of the axes label and plot title. **scale_fill_discrete()** can be used to set the name for the legends. **facet_wrap()** is utilised to generate graphics tables that show the same graph for each group of tenants. **labs()** is used to modify the axes label names. Lastly, the title name is set using **ggtitle()**.

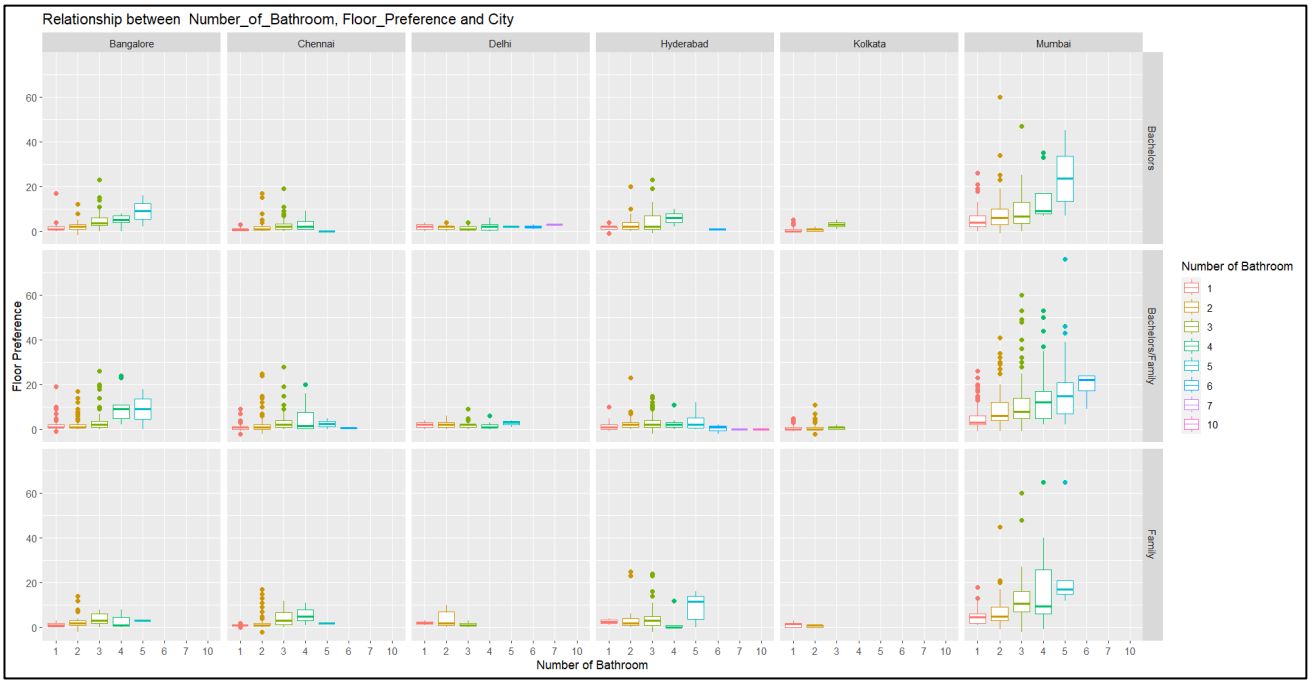



*Figure 6.4.16: Output - Bar Chart (Relationship between Number_of_Bathroom, Floor_Preference and City)*

**Explanation**

From Figure 6.4.16, bachelors prefer houses with lower floor level in each city except houses with 6 bathrooms in Mumbai. Houses with 6 bathrooms in Mumbai are available between about floor 8 and floor 23.

Bachelors/family and family prefer houses with 1 or 2 or 3 bathrooms at lower floor level .

**Findings**

- More bachelors, bachelors/family and family prefer houses with 2 bathrooms with lower floor level in Mumbai

## Conclusion for Question 4

1. More tenants prefer to rent house with 5 bathrooms at the lower and middle floor.
2. The minimum house size that tenant choose house with 1 bathroom is 520 sqft and the maximum house size that tenant choose house with maximum 7 bathrooms is 3100 sqft.
3. More tenants prefer to rent house with 2, 3 and 4 bathrooms at lower total rental fee
4. More tenants prefer semi-furnished houses with 2 bathrooms compared to other furnishing status.
5. More tenants prefer houses with 2 bathrooms located in super area and carpet
6. More tenants prefer houses with 2 bathrooms located in Mumbai and Hyderabad.
7. More tenants prefer to contact owner for houses with 2 bathrooms.
8. More tenants prefer to choose houses with 2 bathrooms in Mumbai.

## Question 5: What are the Factors influencing Tenants to Choose Their Houses with respect to Floor_Preference?

### Analysis 5-1: Find the Relationship between Floor_Preference and House_Size

This analysis is conducted to investigate houses in which floor and what house size are being provided to each tenant.

```
# Bar Chart
ggplot(house_rental_data,aes(x=Floor_Preference,y=House_Size,fill=Floor_Preference)) +
  geom_bar(stat="summary") +
  labs(x="Floor Preference",y="Average House Size (sqft)",color="Floor Preference") +
  ggtitle("Relationship between Floor Preference and House Size") +
  scale_x_continuous(breaks = seq(-2,80,4)) +
  scale_y_continuous(labels=scales::comma) +
  facet_wrap(~Tenant_Type)
```

*Figure 6.5.1: Source Code – Bar Chart to Show the Relationship between Floor_Preference and House_Size*

**Analysis Technique: Data Visualization**

Figure 6.5.1 depicts the source code used to show the number of tenants choose house based on floor preference and house size. The number of tenants choose house based on this relationship is calculated by find the mean using stat = "summary". **labs()** is used to modify the name of the axes and title. **scale_x_continuous()** is the position scales for continuous data of x-axis. The x-axis label is modified so that it produce -2 to 80 with 2 gaps between them in the graph and it is referred to breaks. scale_y_continuous() is the position scales for continuous data of y-axis. The y-axis label is modified so that it do not show the e notation in the y-axis, and this is referred to labels. **facet_wrap()** is utilised to generate graphics tables that show the same graph for each group of tenants.

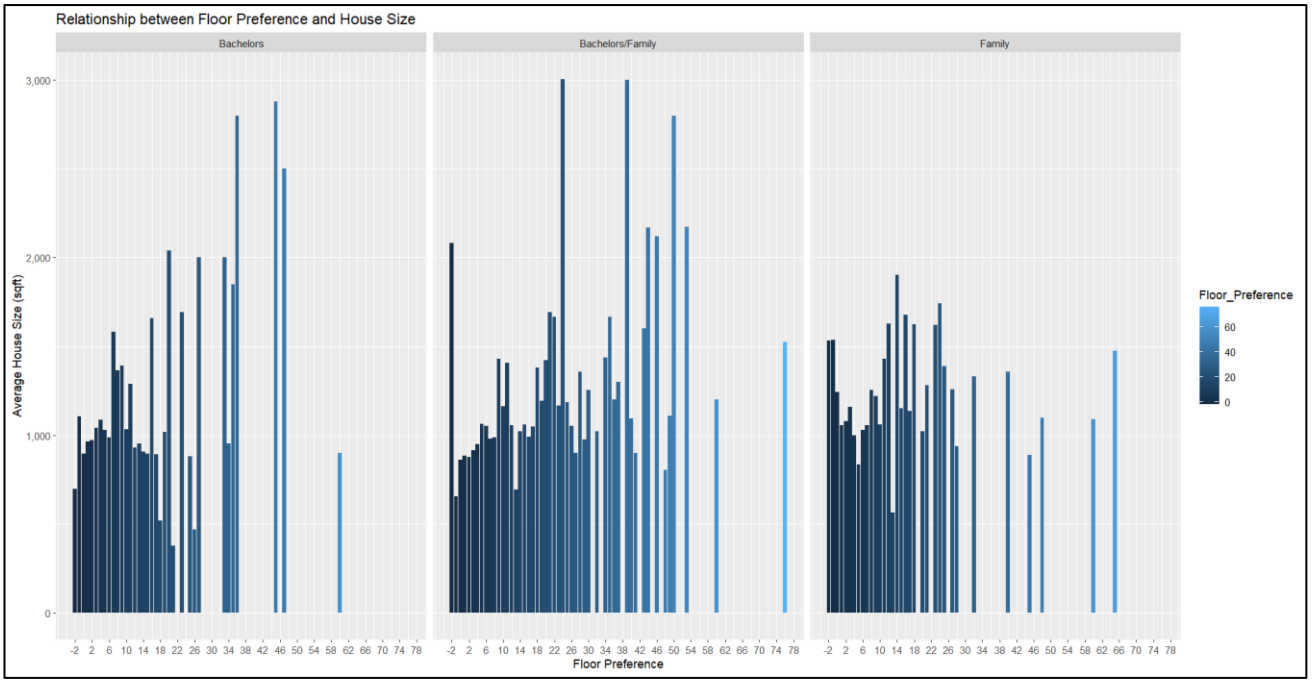



*Figure 6.5.2: Output – Bar Chart (Relationship between Floor_Preference and House_Size)*

**Explanation**

From Figure 6.5.2, the average house size at lower floor level is smaller compared to those at higher floor level. However, when it is higher than 50 floors, then the house size becomes smaller. The largest houses which bachelors can choose is around 2800 sqft and it is located at around floor 46. The largest houses which bachelors/family can choose is 3000 sqft and it is available at around floor 24 and floor 39, whereas the maximum house size that family can choose is around 1800 sqft which is located at around floor 17 or floor 18.

**Findings**

- Houses larger than 2000sqft are found located at middle floor

## Analysis 5-2: Find the Relationship between Floor_Preference and Rental_Fee

This analysis is conducted to investigate whether the floor preference can affect the rental fee or not.

```
# Combination of Scatter and Line Graph
ggplot(rf_no_outliers,aes(x=Floor_Preference,y=Rental_Fee,color=Tenant_Type)) +
  geom_line(stat="summary") +
  geom_point(stat="summary") +
  facet_wrap(~Tenant_Type) +
  scale_x_continuous(breaks=seq(-2,80,2)) +
  scale_y_continuous(labels = scales::comma) +
  labs(x="Floor Preference",y="Average Rental Fee (RM)") +
  ggtitle("Relationship between Floor_Preference and Rental_Fee")
```

*Figure 6.5.3: Source Code – Point and Line Graph to Show the Relationship between Floor_Preference and Rental_Fee*

**Analysis Technique: Data Visualization and Manipulation**

Figure 6.5.3 depicts the source code used to show how changing floor preference influence rental fee The outliers in rental fee has been removed. The number of tenants choose house based on this relationship is calculated by find the mean using stat = "summary". This graph is a combination of point and line graph. **facet_wrap()** is utilised to generate graphics tables that show the same graph for each group of tenants. **scale_y_continuous()** is the position scales for continuous data of y-axis. The y-axis label is modified so that it do not show the e notation in the y-axis, and this is referred to labels. **labs()** is used to modify the name of the axes and title.

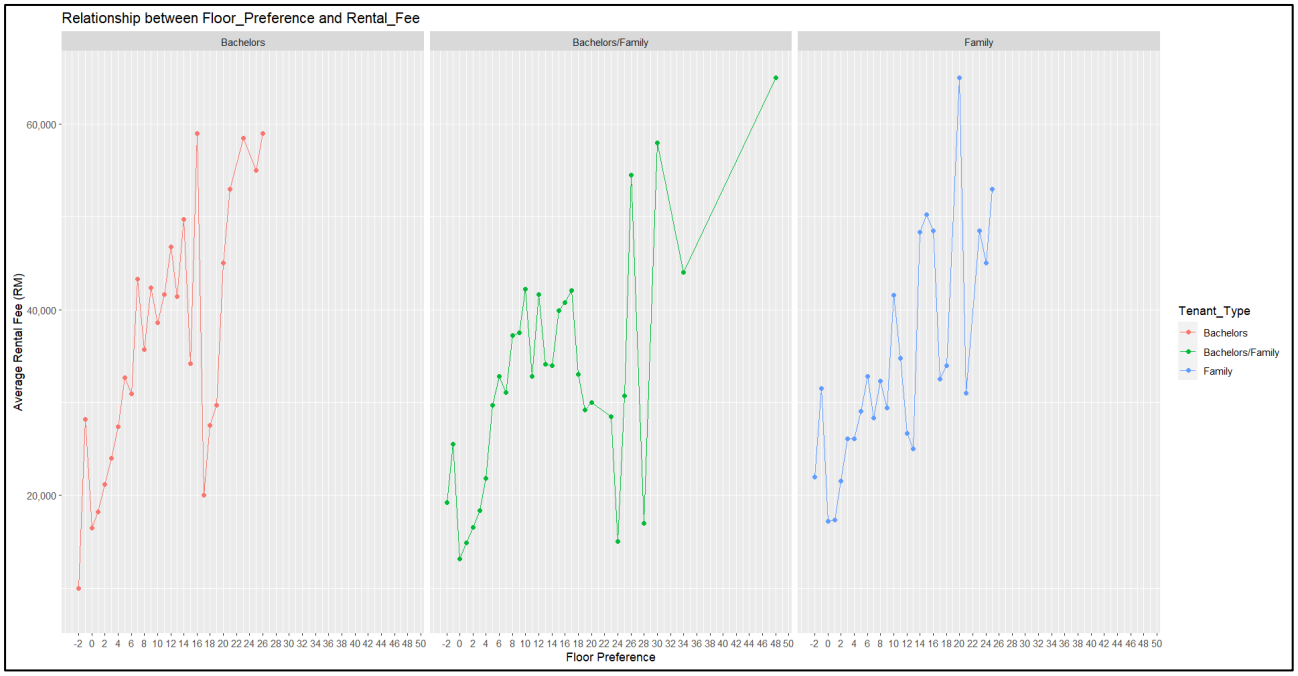



*Figure 6.5.4: Output – Point and Line Graph (Relationship between Floor_Preference and Rental_Fee)*

**Explanation**

From Figure 6.5.4, the average rental fee is having an increase trend when the houses are located at higher floor levels. However, there is a steep decrease between floor 16 and floor 18. The highest average rental fee in family site is above RM60,000 in floor 20.

**Findings**

- The higher the house located, the higher the rental fee is.

## Analysis 5-3: Find the Relationship between Floor_Preference and Furnishing_Status

```
# Box Plot
ggplot(house_rental_data,aes(x=Floor_Preference,y=Furnishing_Status)) +
  geom_boxplot(aes(x=Floor_Preference,y=Furnishing_Status,color=Furnishing_Status)) +
  labs(x= "Floor Preference",y="Furnishing Status",
       color="Furnishing Status",
       title="Relationship between Floor Preference and Furnishing Status") +
  scale_x_continuous(breaks=seq(-2,80,4)) +
  theme(legend.position = "bottom") +
  facet_wrap(~Tenant_Type)
```

*Figure 6.5.5: Source Code – Box Plot to Show the Relationship between Floor_Preference and Furnishing_Status*

**Analysis Technique: Data Visualization**

Figure 6.5.5 depicts the source code used to show how changing floor preference influence rental fee The outliers in rental fee has been removed. The number of tenants choose house based on this relationship is calculated by find the mean using stat = "summary". This graph is a combination of point and line graph. **facet_wrap()** is utilised to generate graphics tables that show the same graph for each group of tenants. **scale_y_continuous()** is the position scales for continuous data of y-axis. The y-axis label is modified so that it do not show the e notation in the y-axis, and this is referred to labels. **labs()** is used to modify the name of the axes and title.

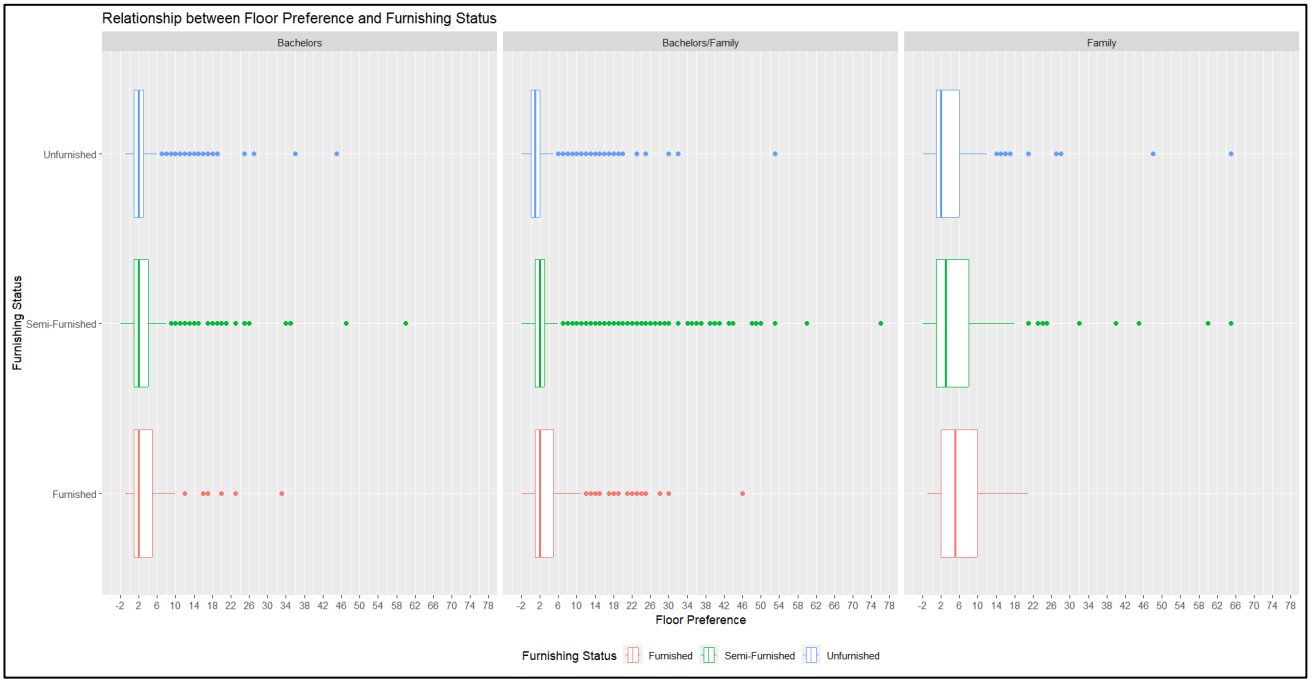



*Figure 6.5.6: Output – Box Plot (Relationship between Floor_Preference and Furnishing_Status)*

**Explanation**

From Figure 6.5.6, the unfurnished houses which bachelors prefer to choose is located between 1st floor and 3rd floor. The semi-furnished houses which bachelors prefer to choose is located between 1st floor and the 4th floor. The unfurnished houses which bachelors prefer to choose is located between 1st floor and the 5th floor.

Bachelors/family prefer to choose unfurnished houses which is located between ground floor and 2nd floor. The semi-furnished houses which they prefer to choose is located between 1st floor and the 3rd floor, whereas the unfurnished houses which they prefer to choose is located between 1st floor and 5th floor.

Family prefer to choose unfurnished houses which is located between 1st floor and 6th floor. The semi-furnished houses which they prefer to choose is located between first 1st floor and 8th floor and the unfurnished houses which they prefer to choose is located between 2nd floor and 10th floor.

**Findings**

- More bachelors and bachelors/family prefer semi-furnished houses located at 2nd floor.
- More family prefer semi-furnished houses located at 3rd floor.

## Analysis 5-4: Find the Relationship between Floor_Preference and Area_Type

This analysis is conducted to investigate which floor in each area type do tenants prefer to choose their houses.

```
# Box Plot
ggplot(house_rental_data,aes(x=Floor_Preference,y=Area_Type)) +
  geom_boxplot(aes(x=Floor_Preference,y=Area_Type,color=Area_Type))+
  labs(x= "Floor Preference",y="Area Type",
       color="Area Type",
       title="Relationship between Floor Preference and Area Type") +
  scale_x_continuous(breaks=seq(-2,80,4)) +
  theme(legend.position = "bottom") +
  facet_wrap(~Tenant_Type)
```

*Figure 6.5.7: Source Code – Box Plot to Show the Relationship between Floor_Preference and Area_Type*

**Analysis Technique: Data Visualization**

Figure 6.5.7 depicts the source code used to show which floor in each area do tenants prefer to choose. The relationship is represented using box plot. **labs()** is used to modify the name of the axes, name of the legend and title plot. **scale_x_continuous()** is the position scales for continuous data of x-axis. The x-axis label is modified so that it produce -2 to 80 with 4 gaps between them in the graph and it is referred to breaks. The **theme()** is used to customize the non-data components such as the position of the legend. **facet_wrap()** is utilised to generate graphics tables that show the same graph for each group of tenants.

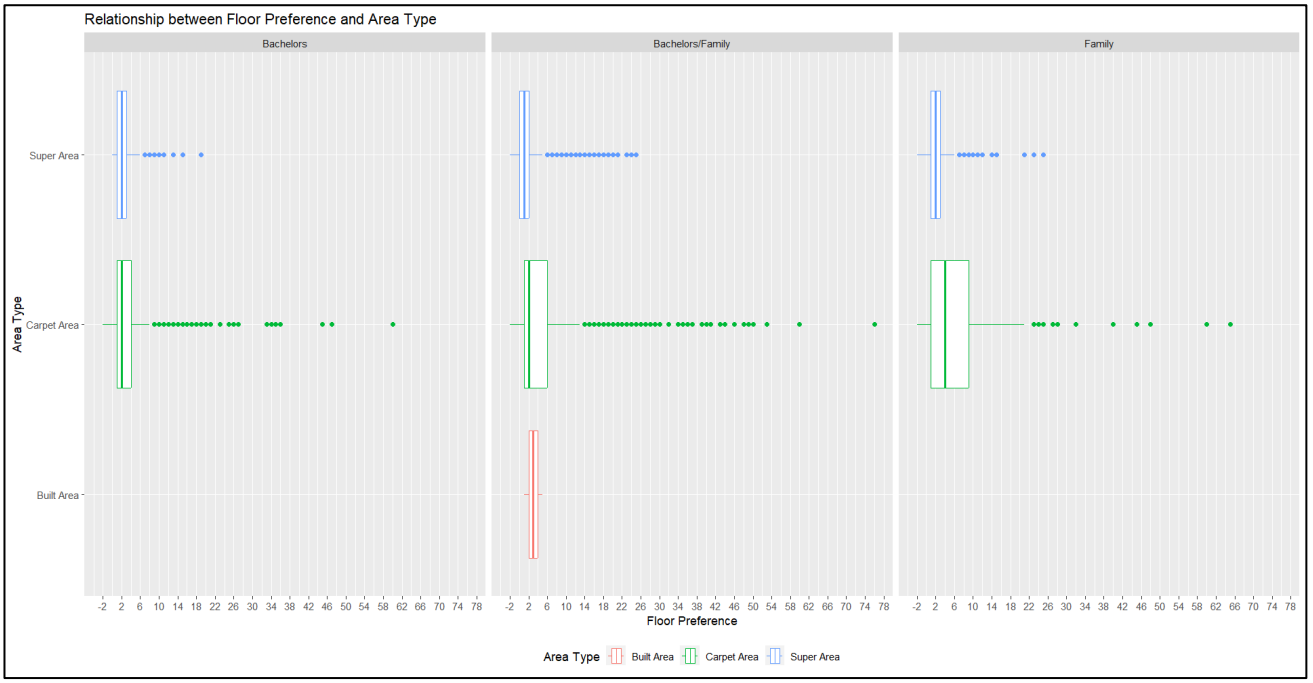



*Figure 6.5.8: Output – Box Plot (Relationship between Floor_Preference and Area_Type)*

**Explanation**

From Figure 6.5.8, bachelors prefer to choose their houses in super area which located between 1st floor and 3rd floor. Some of them prefer to choose their houses in carpet area which located between

1st floor and 4th floor. None of them prefer to choose houses in built area.

Bachelors/family prefer to choose houses in super area which located either at 1st floor or 2nd floor. They prefer to choose houses in carpet area which located between 1st floor and 6th floor. Fewer of them prefer to choose houses in built area which located 2nd floor and 4th floor.

Family prefer to choose houses in super area which located between 1st floor and 3rd floor. They prefer to choose houses in carpet area which located between 1st floor and 9th floor. None of them prefer to choose houses in built area.

**Findings**

- Fewer bachelors/family choose built area which located between 2nd floor and 4th floor.
- More bachelors prefer to choose houses in carpet area which located between 1st floor and 4th floor
- More bachelors/family prefer to choose houses in carpet area which located between 1st floor and 6th floor.
- More family prefer to choose houses in carpet area which located between 1st floor and 9th floor

## Analysis 5-5: Find the Relationship between Floor_Preference and City

This analysis is conducted to investigate which floor in each city do tenants prefer to choose their houses.

```
# Violin Plot with Box Plot
ggplot(house_rental_data,aes(x=Floor_Preference,y=City)) +
  geom_violin(aes(x=Floor_Preference,y=City,color=City))+
  geom_boxplot(width=0.05) +
  labs(x= "Floor Preference",y="City",
       color="City",
       title="Relationship between Floor Preference and City") +
  scale_x_continuous(breaks=seq(-2,80,4)) +
  theme(legend.position = "bottom") +
  facet_wrap(~Tenant_Type)
```

*Figure 6.5.9: Source Code – Violin Plot with Box Plot to Show the Relationship between Floor_Preference and City*

**Analysis Technique: Data Visualization**

Figure 6.5.9 depicts the source code used to show which floor in each city do tenants prefer to choose. The relationship is represented using violin plot with box plot inside it. **labs()** is used to modify the name of the axes, name of the legend and title plot. **scale_x_continuous()** is the position scales for continuous data of x-axis. The x-axis label is modified so that it produce -2 to 80 with 4 gaps between them in the graph and it is referred to breaks. **facet_wrap()** is utilised to generate graphics tables that show the same graph for each group of tenants. The **theme()** is used to customize the non-data components such as the position of the legend.

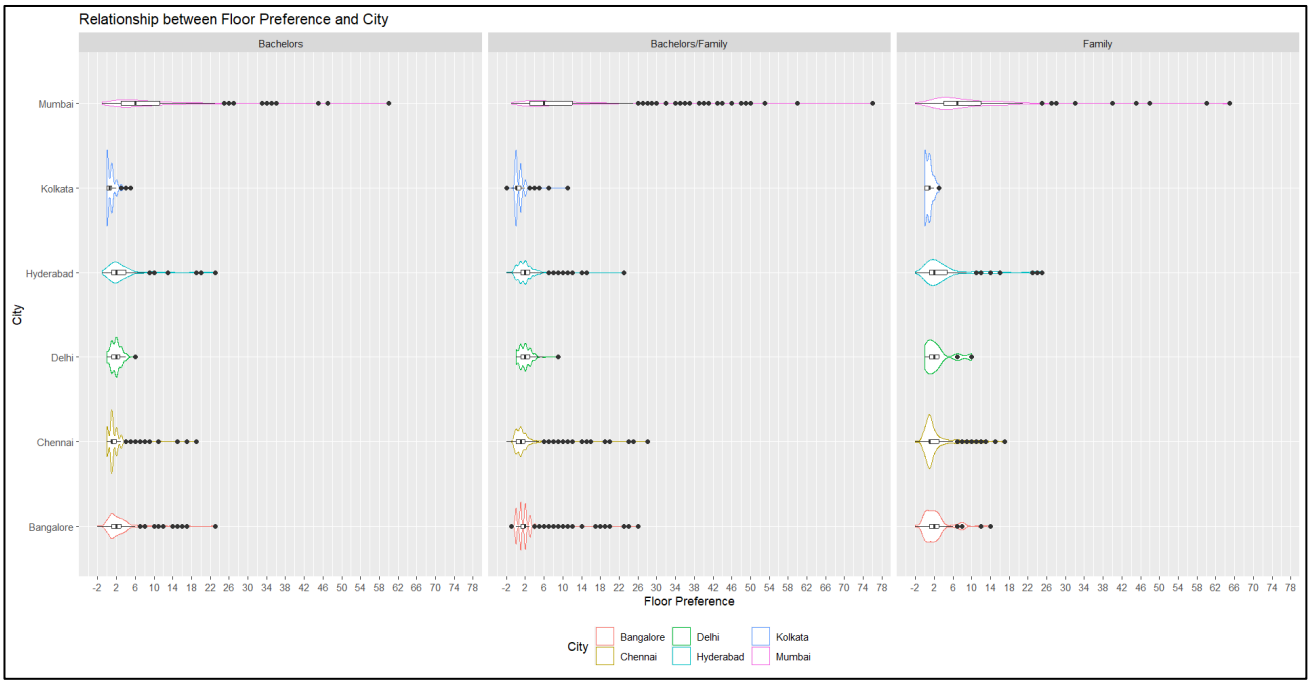



*Figure 6.5.10: Output – Violin Plot (Relationship between Floor_Preference and City)*

```
>   house_rental_data %>%
+     group_by(Tenant_Type,City) %>%
+     count()
# A tibble: 18 × 3
# Groups:   Tenant_Type, City [18]
   Tenant_Type      City          n
   <chr>            <chr>     <int>
 1 Bachelors        Bangalore   135
 2 Bachelors        Chennai     137
 3 Bachelors        Delhi       162
 4 Bachelors        Hyderabad   102
 5 Bachelors        Kolkata     122
 6 Bachelors        Mumbai      172
 7 Bachelors/Family Bangalore   694
 8 Bachelors/Family Chennai     649
 9 Bachelors/Family Delhi       432
10 Bachelors/Family Hyderabad   676
11 Bachelors/Family Kolkata     379
12 Bachelors/Family Mumbai      614
13 Family           Bangalore    57
14 Family           Chennai     105
15 Family           Delhi        11
16 Family           Hyderabad    90
17 Family           Kolkata      23
18 Family           Mumbai      186
```

*Figure 6.5.11: Number of Tenants in Each City*

**Explanation**

From Figure 6.5.10 and Figure 6.5.11, 135 of bachelors prefer to choose house in Bangalore located between 1st floor and 3rd floor. In Chennai, there is 137 of the bachelors prefer to choose house located at 1st floor and 2nd floor. 162 of them prefer to choose house in Delhi located between the 1st floor and 3rd floor. In Hyderabad, 102 of them prefer to choose house located between 1st floor and 4th floor. 122 of them prefer to choose house in Kolkata located at ground floor and 1st floor. In Mumbai, 172 of the bachelors prefer to choose house located between 3rd floor and 11th floor.

694 of bachelors/family prefer to choose house in Bangalore located at 1st floor and 2nd floor. In Chennai, there is 649 of the bachelors/family prefer to choose house located between ground floor and 2nd floor. 432 of them prefer to choose house in Delhi located between 1st floor and 3rd floor. In Hyderabad, 676 of them prefer to choose house located between 1st floor and 3rd floor. 379 of them prefer to choose house in Kolkata located at ground floor and 1st floor. In Mumbai, 172 of the bachelors/family prefer to choose house located between 3rd floor and 12th floor.

57 of family prefer to choose house in Bangalore located between 1st floor and 3rd floor. In Chennai, there is 105 of the family prefer to choose house located between 1st floor and 3rd floor. 11 of them prefer to choose house in Delhi located between 1st floor and 3rd floor. In Hyderabad, 90 of them prefer to choose house located between 1st floor and 5th floor. 23 of them prefer to choose house in Kolkata located at ground floor and 1st floor. In Mumbai, 186 of the family prefer to choose house located between 4th floor and 12th floor.

**Findings**

- More bachelors and bachelors/family prefer to choose house located at 3rd floor in Mumbai
- More family prefer to choose house located at 4th floor in Mumbai.

## Analysis 5-6: Find the Relationship between Floor_Preference and Point_of_Contact

This analysis is conducted to find out tenants prefer to contact who and what are the preferred floor level.

```
# Violin Plot with Box Plot
ggplot(house_rental_data,aes(x=Floor_Preference,y=Point_of_Contact)) +
  geom_violin(aes(x=Floor_Preference,y=Point_of_Contact,color=Point_of_Contact)) +
  geom_boxplot(width=0.05) +
  labs(x= "Floor Preference",y="Point of Contact",
       color="Point of Contact",
       title="Relationship between Floor Preference and Point of Contact") +
  scale_x_continuous(breaks=seq(-2,80,4)) +
  theme(legend.position = "bottom") +
  facet_wrap(~Tenant_Type)
```

*Figure 6.5.12: Source Code – Violin Plot with Box Plot to Show the Relationship between Floor_Preference and Point_of_Contact*

**Analysis Technique: Data Visualization**

Figure 6.5.12 depicts the source code used to show which floor in each city do tenants prefer to choose. The relationship is represented using violin plot with box plot inside it. **labs()** is used to modify the name of the axes, name of the legend and title plot. **scale_x_continuous()** is the position scales for continuous data of x-axis. The x-axis label is modified so that it produce -2 to 80 with 4 gaps between them in the graph and it is referred to breaks. **facet_wrap()** is utilised to generate graphics tables that show the same graph for each group of tenants. The **theme()** is used to customize the non-data components such as the position of the legend.

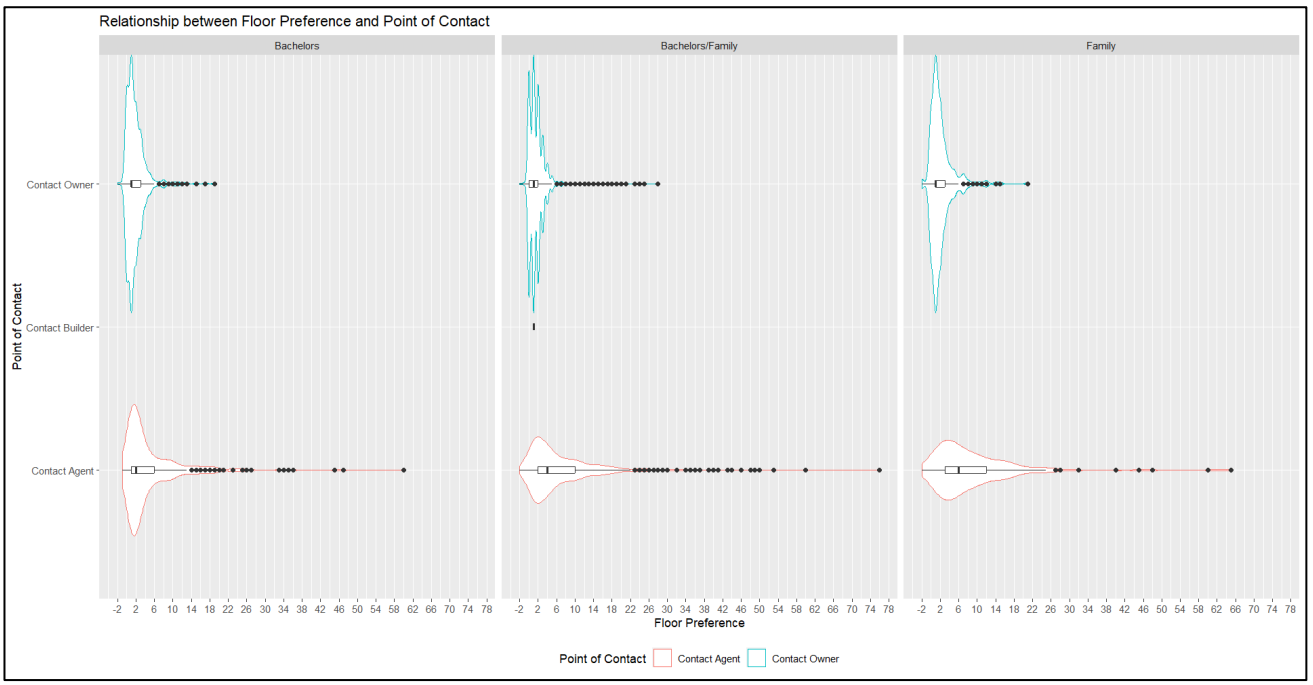



*Figure 6.5.13: Output – Violin Plot with Box Plot (Relationship between Floor Preference and Point_of_Contact)*

```
>   house_rental_data %>%
+     group_by(Tenant_Type,Point_of_Contact) %>%
+     count()
# A tibble: 7 x 3
# Groups:   Tenant_Type, Point_of_Contact [7]
  Tenant_Type      Point_of_Contact     n
  <chr>            <chr>            <int>
1 Bachelors        Contact Agent      434
2 Bachelors        Contact Owner      396
3 Bachelors/Family Contact Agent      849
4 Bachelors/Family Contact Builder      1
5 Bachelors/Family Contact Owner     2594
6 Family           Contact Agent      246
7 Family           Contact Owner      226
```

*Figure 6.5.14: Number of Tenants who Contact Agent, Owner and Builder*

**Explanation**

From Figure 6.5.13 and Figure 6.5.14, 434 of bachelors prefer to choose house located between 1st floor and 6th floor by contacting agent. 396 of them prefer to choose house located between 1st floor and 3rd floor. None of them prefer to contact builder to choose their houses.

849 of the bachelors/family prefer to contact agent to choose house located between 2nd floor and 10th floor. Only 1 of them prefer to contact builder which the house is located at 1st floor. 2594 of them prefer to contact owner which the house is located between ground floor and 2nd floor.

246 of the family prefer to contact agent to choose their house which is located between 3rd floor and 12th floor. 226 of them prefer to contact agent to choose their house which is located between 1st floor and 3rd floor.

**Findings**

- More bachelors prefer to contact agent to choose their house which is located between 1st and 3rd floor.
- More than 50% of the bachelors/family prefer to contact owner to choose their house which is located between ground floor and 1st floor.
- Slightly more family prefer to contact agent to choose their houses between 3rd floor and 12th floor.

## Analysis 5-7: Find the Relationship between Floor_Preference, Bedroom_Hall_Kitchen and Furnishing_Status

This analysis is conducted to investigate house available for each tenant type based on Floor_Preference, Bedroom_Hall_Kitchen and Furnishing_Status.

```
# Box Plot
ggplot(house_rental_data,aes(x=Floor_Preference,y=Bedroom_Hall_Kitchen)) +
  geom_boxplot(aes(x=Floor_Preference,y=factor(Bedroom_Hall_Kitchen),color=Furnishing_Status))+
  labs(x= "Floor Preferance",y="Number of Bedroom, Hall and Kitchen",
       color="Number of Bedroom, Hall and Kitchen",
       title="Relationship between Floor_Preference, Number of Bedroom_Hall_Kitchen and Furnishing_Status") +
  scale_x_continuous(breaks=seq(-2,80,4)) +
  scale_fill_discrete(name="Number of Bedroom, Hall and Kitchen") +
  theme(legend.position = "bottom") +
  facet_grid(Tenant_Type~Furnishing_Status)
```

*Figure 6.5.15: Source Code – Box Plot to Show the Relationship between Floor_Preference, Bedroom_Hall_Kitchen and Furnishing_Status*

**Analysis Technique: Data Visualization**

Figure 6.5.15 depicts the source code used to show how tenants choose their house based on Floor_Preference, Bedroom_Hall_Kitchen and Furnishing_Status. The relationship is represented using box plot. **labs()** is used to modify the name of the axes, name of the legend and title plot. **scale_x_continuous()** is the position scales for continuous data of x-axis. The x-axis label is modified so that it produce -2 to 80 with 4 gaps between them in the graph and it is referred to breaks. **scale_fill_discrete()** can be used to set the name for the legends. The **theme()** is used to customize the non-data components such as the position of the legend. **facet_grid()** generates a 2D grid of panels with the rows and columns given by Tenant_Type and Furnishing_Status.

```
>   # Count Number of Tenants choose house based on Bedroom_Hall_Kitchen and Furnishing_Status
>   # By Bachelors
>   house_rental_data %>% filter(Tenant_Type=="Bachelors") %>%
+     group_by(Tenant_Type,Bedroom_Hall_Kitchen,Furnishing_Status) %>%
+     count()
# A tibble: 15 x 4
# Groups:   Tenant_Type, Bedroom_Hall_Kitchen, Furnishing_Status [15]
   Tenant_Type Bedroom_Hall_Kitchen Furnishing_Status     n
   <chr>                      <int> <chr>            <int>
 1 Bachelors                      1 Furnished           19
 2 Bachelors                      1 Semi-Furnished      70
 3 Bachelors                      1 Unfurnished        102
 4 Bachelors                      2 Furnished           41
 5 Bachelors                      2 Semi-Furnished     131
 6 Bachelors                      2 Unfurnished        201
 7 Bachelors                      3 Furnished           33
 8 Bachelors                      3 Semi-Furnished     106
 9 Bachelors                      3 Unfurnished         88
10 Bachelors                      4 Furnished            3
11 Bachelors                      4 Semi-Furnished      16
12 Bachelors                      4 Unfurnished         15
13 Bachelors                      5 Furnished            1
14 Bachelors                      5 Unfurnished          3
15 Bachelors                      6 Semi-Furnished       1
```

*Figure 6.5.16 : Source Code and Output – Count Number of Bachelors Based on Bedroom_Hall_Kitchen and Furnishing_Status*

```
>   # By Bachelors/Family
>   house_rental_data %>% filter(Tenant_Type=="Bachelors/Family") %>%
+     group_by(Tenant_Type,Bedroom_Hall_Kitchen,Furnishing_Status) %>%
+     count()
# A tibble: 18 x 4
# Groups:   Tenant_Type, Bedroom_Hall_Kitchen, Furnishing_Status [18]
   Tenant_Type      Bedroom_Hall_Kitchen Furnishing_Status     n
   <chr>                          <int> <chr>            <int>
 1 Bachelors/Family                   1 Furnished          135
 2 Bachelors/Family                   1 Semi-Furnished     347
 3 Bachelors/Family                   1 Unfurnished        432
 4 Bachelors/Family                   2 Furnished          215
 5 Bachelors/Family                   2 Semi-Furnished     813
 6 Bachelors/Family                   2 Unfurnished        651
 7 Bachelors/Family                   3 Furnished          132
 8 Bachelors/Family                   3 Semi-Furnished     428
 9 Bachelors/Family                   3 Unfurnished        148
10 Bachelors/Family                   4 Furnished           22
11 Bachelors/Family                   4 Semi-Furnished      78
12 Bachelors/Family                   4 Unfurnished         26
13 Bachelors/Family                   5 Furnished            2
14 Bachelors/Family                   5 Semi-Furnished       6
15 Bachelors/Family                   5 Unfurnished          3
16 Bachelors/Family                   6 Furnished            2
17 Bachelors/Family                   6 Semi-Furnished       3
18 Bachelors/Family                   6 Unfurnished          1
```

*Figure 6.5.17: Source Code and Output – Count Number of Bachelors/Family Based on Bedroom_Hall_Kitchen and Furnishing_Status*

```
>   house_rental_data %>% filter(Tenant_Type=="Family") %>%
+     group_by(Tenant_Type,Bedroom_Hall_Kitchen,Furnishing_Status) %>%
+     count()
# A tibble: 15 x 4
# Groups:   Tenant_Type, Bedroom_Hall_Kitchen, Furnishing_Status [15]
   Tenant_Type Bedroom_Hall_Kitchen Furnishing_Status     n
   <chr>                      <int> <chr>            <int>
 1 Family                         1 Furnished           11
 2 Family                         1 Semi-Furnished      27
 3 Family                         1 Unfurnished         24
 4 Family                         2 Furnished           29
 5 Family                         2 Semi-Furnished     102
 6 Family                         2 Unfurnished         82
 7 Family                         3 Furnished           25
 8 Family                         3 Semi-Furnished     106
 9 Family                         3 Unfurnished         32
10 Family                         4 Furnished            7
11 Family                         4 Semi-Furnished      17
12 Family                         4 Unfurnished          5
13 Family                         5 Furnished            3
14 Family                         5 Unfurnished          1
15 Family                         6 Unfurnished          1
```

*Figure 6.5.18: Source Code and Output – Count Number of Family Based on Bedroom_Hall_Kitchen and Furnishing_Status*

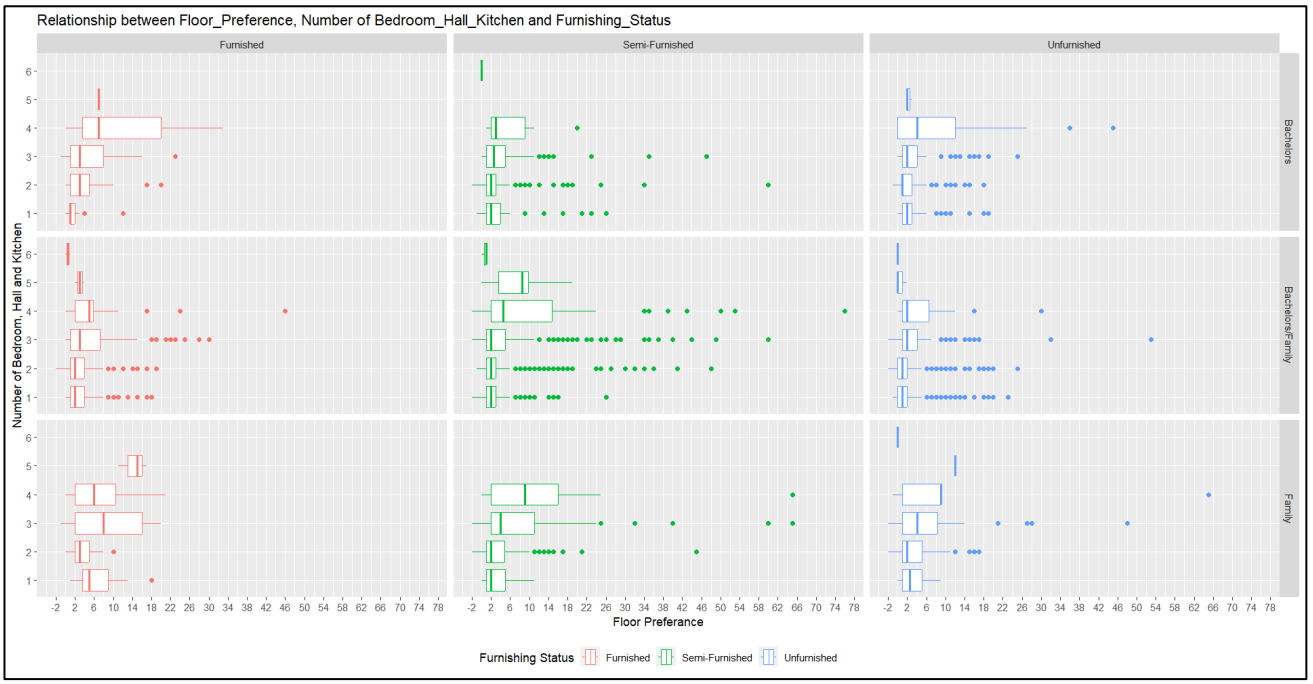



*Figure 6.5.19: Output – Box Plot (Relationship between Floor_Preference, Bedroom_Hall_Kitchen and Furnishing_Status)*

**Explanation**

From Figure 6.5.16 and Figure 6.5.19, more bachelors prefer unfurnished houses with 2 bedrooms, hall, and kitchen which located between 1st floor and 3rd floor.

From Figure 6.5.17 and Figure 6.5.19, more bachelors/family prefer semi-furnished houses with 2 bedrooms, hall, and kitchen which located between 1st floor and 3rd floor.

From Figure 6.5.18 and Figure 6.5.19, more family prefer semi-furnished houses with 3 bedrooms, hall, and kitchen which located between 2nd floor and 11th floor.

**Findings**

- More bachelors prefer unfurnished houses with 2 bedrooms, hall, and kitchen located at 1st floor.

- More bachelors/family prefer semi-furnished houses with 2 bedrooms, hall, and kitchen located at 1st floor.
- More family prefer semi-furnished houses with 2 bedrooms, hall, and kitchen located at 1st floor.

## Analysis 5-8: Find the Relationship between Floor_Preference, Number_of_Bathroom and Area_Type

This analysis is conducted to investigate how tenant choose based on floor preference, number of bathrooms in each area.

```
# Box Plot
ggplot(house_rental_data,aes(x=Floor_Preference,y=Number_of_Bathroom)) +
  geom_boxplot(aes(x=Floor_Preference,y=factor(Number_of_Bathroom),color=Area_Type))+
  labs(x= "Floor Preferance",y="Number of Bathroom",
       color="Area Type",
       title="Relationship between Floor_Preference, Number_of_Bathroom and Area_Type") +
  scale_x_continuous(breaks=seq(-2,80,4)) +
  scale_y_discrete(breaks=seq(1,10,1)) +
  theme(legend.position = "bottom") +
  facet_grid(Tenant_Type~Area_Type)
```

*Figure 6.5.20: Source Code – Box Plot to Show the relationship between Floor_Preference, Number_of_Bathroom and Area_Type*

**Analysis Technique: Data Visualization**

Figure 6.5.20 depicts the source code used to show how tenants choose their house based on Floor_Preference, Number_of_Bathroom and Area_Type. The relationship is represented using box plot. **labs()** is used to modify the name of the axes, name of the legend and title plot. **scale_x_discrete()** is the position scales for discrete data of x-axis. The x-axis label is modified so that it produce 1 to 10 with 1 gap between them in the graph and it is referred to breaks. **scale_y_continuous()** is the position scales for discrete data of y-axis. The y-axis label is modified so that it produce -2 to 80 with 4 gaps between them in the graph and it is referred to breaks. The **theme()** is used to customize the non-data components such as the position of the legend. **facet_wrap()** is utilised to generate graphics tables that show the same graph for each group of tenants.

```
>   # Count Number of Tenants choose house based on Number_of_Bathroom and Area_Type
>   # By Bachelors
>   house_rental_data %>% filter(Tenant_Type=="Bachelors") %>%
+     group_by(Tenant_Type,Number_of_Bathroom,Area_Type) %>%
+     count()
# A tibble: 11 x 4
# Groups:   Tenant_Type, Number_of_Bathroom, Area_Type [11]
   Tenant_Type Number_of_Bathroom Area_Type       n
   <chr>                   <int> <chr>       <int>
 1 Bachelors                   1 Carpet Area   193
 2 Bachelors                   1 Super Area     46
 3 Bachelors                   2 Carpet Area   329
 4 Bachelors                   2 Super Area     63
 5 Bachelors                   3 Carpet Area   131
 6 Bachelors                   3 Super Area     26
 7 Bachelors                   4 Carpet Area    23
 8 Bachelors                   4 Super Area      4
 9 Bachelors                   5 Carpet Area    11
10 Bachelors                   6 Carpet Area     3
11 Bachelors                   7 Carpet Area     1
```

*Figure 6.5.21: Source Code and Output – Count Number of Bachelors Based on Number_of_Bathroom and Area_Type*

```
>   # By Bachelors/Family
>   house_rental_data %>% filter(Tenant_Type=="Bachelors/Family") %>%
+     group_by(Tenant_Type,Number_of_Bathroom,Area_Type) %>%
+     count()
# A tibble: 16 x 4
# Groups:   Tenant_Type, Number_of_Bathroom, Area_Type [16]
   Tenant_Type      Number_of_Bathroom Area_Type      n
   <chr>                         <int> <chr>       <int>
 1 Bachelors/Family                  1 Built Area      1
 2 Bachelors/Family                  1 Carpet Area   361
 3 Bachelors/Family                  1 Super Area    815
 4 Bachelors/Family                  2 Built Area      1
 5 Bachelors/Family                  2 Carpet Area   565
 6 Bachelors/Family                  2 Super Area   1078
 7 Bachelors/Family                  3 Carpet Area   251
 8 Bachelors/Family                  3 Super Area    226
 9 Bachelors/Family                  4 Carpet Area    69
10 Bachelors/Family                  4 Super Area     29
11 Bachelors/Family                  5 Carpet Area    28
12 Bachelors/Family                  5 Super Area      8
13 Bachelors/Family                  6 Carpet Area     7
14 Bachelors/Family                  6 Super Area      2
15 Bachelors/Family                  7 Super Area      2
16 Bachelors/Family                 10 Super Area      1
```

*Figure 6.5.22: Source Code and Output - Count Number of Bachelors/Family Based on Number_of_Bathroom and Area_Type*

```
>   # By Family
>   house_rental_data %>% filter(Tenant_Type=="Family") %>%
+     group_by(Tenant_Type,Number_of_Bathroom,Area_Type) %>%
+     count()
# A tibble: 10 x 4
# Groups:   Tenant_Type, Number_of_Bathroom, Area_Type [10]
   Tenant_Type Number_of_Bathroom Area_Type      n
   <chr>                   <int> <chr>       <int>
 1 Family                      1 Carpet Area    32
 2 Family                      1 Super Area     26
 3 Family                      2 Carpet Area   173
 4 Family                      2 Super Area     82
 5 Family                      3 Carpet Area    86
 6 Family                      3 Super Area     29
 7 Family                      4 Carpet Area    25
 8 Family                      4 Super Area      6
 9 Family                      5 Carpet Area    10
10 Family                      5 Super Area      3
```

*Figure 6.5.23: Source Code and Output - Count Number of Family Based on Number_of_Bathroom and Area_Type*

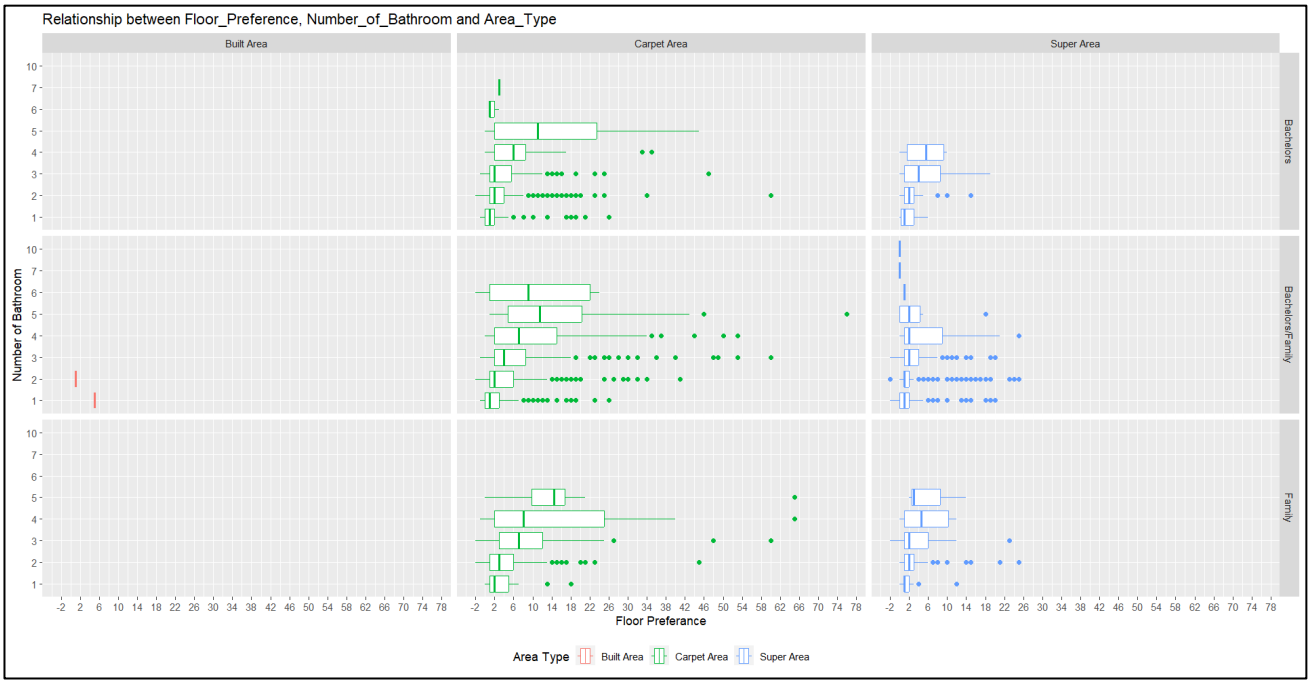



*Figure 6.5.24: Output – Box Plot (Relationship between Floor_Preference, Number_of_Bathroom and Area_Type)*

**Explanation**

From Figure 6.5.21 and Figure 6.5.24, more bachelors prefer houses with 2 bathrooms in carpet area located between 1st floor and 4th floor.

From Figure 6.5.22 and Figure 6.5.24, more bachelors/family prefer houses with 2 bathrooms in super area located at 1st floor and 2nd floor.

From Figure 6.5.23 and Figure 6.5.24, more family prefer houses with 2 bathrooms in carpet area located between 1st floor and 6th floor.

**Findings**

- More bachelors prefer to contact agent to choose their house which is located between 1st and 3rd floor.
- More than 50% of the bachelors/family prefer to contact owner to choose their house which is located between ground floor and 1st floor.
- Slightly more family prefer to contact agent to choose their houses between 3rd floor and 12th floor.

## Conclusion For Question 5

1. More tenants prefer to choose houses that is lower than 20 with house size is between 1000sqft and 2000sqft.
2. Higher floor has higher rental fee.
3. More tenants prefer to choose unfurnished, semi-furnished and furnished houses that is located between ground floor and 2th floor
4. More tenants prefer to choose houses in carpet area which mostly located between 1st and 4th floor.
5. More tenants prefer to choose their house in Mumbai located at 3rd and 4th floors.
6. More tenants prefer to contact owner to choose their house which is located at 3rd floor in common.
7. More tenants prefer semi-furnished houses with 4 bedrooms, hall, and kitchen located at 3rd floor.

## Question 6: What are the Factors influencing Tenants to Choose Their Houses with respect to House Size?

### Analysis 6-1: Find the Relationship between House_Size and Rental Fee

This analysis is used to investigate how the house size increase varies the rental fee and how tenant choose their house based on house size and rental fee.

```
# Point Graph
ggplot(rf_no_outliers,aes(x=House_Size,y=Rental_Fee)) +
  geom_point(stat="summary",fun="mean") +
  geom_smooth(method="lm") +
  facet_wrap(~Tenant_Type) +
  scale_x_continuous(labels = scales::comma) +
  scale_y_continuous(labels = scales::comma) +
  labs(x="Size of House",y="Rental Fee (RM)") +
  ggtitle("Relationship between House_Size and Rental_Fee")
```

*Figure 6.6.1: Source Code – Line Graph to Show the Relationship between House_Size and Rental_Fee*

**Analysis Technique: Data Visualization and Manipulation**

Figure 6.6.1 depicts the source code used to show the number of tenants choose house based on house size and rental fee. The outliers in the rental fee have been removed. The number of tenants choose house based on this relationship is calculated by find the mean using stat = "summary". **geom_smooth()** is used to draw the best fit line of the points. **facet_wrap()** is utilised to generate graphics tables that show the same graph for each group of tenants. **scale_x_continuous()** is the position scales for continuous data of x-axis, while **scale_y_continuous()** is the position scales for continuous data of y-axis. The x-axis label and y-axis label are modified so that it do not show e notation on the graph, and this is referred to labels. **Labs()** is used to modify the name of the axes and title. Lastly, the title name is set using **ggtitle()**.

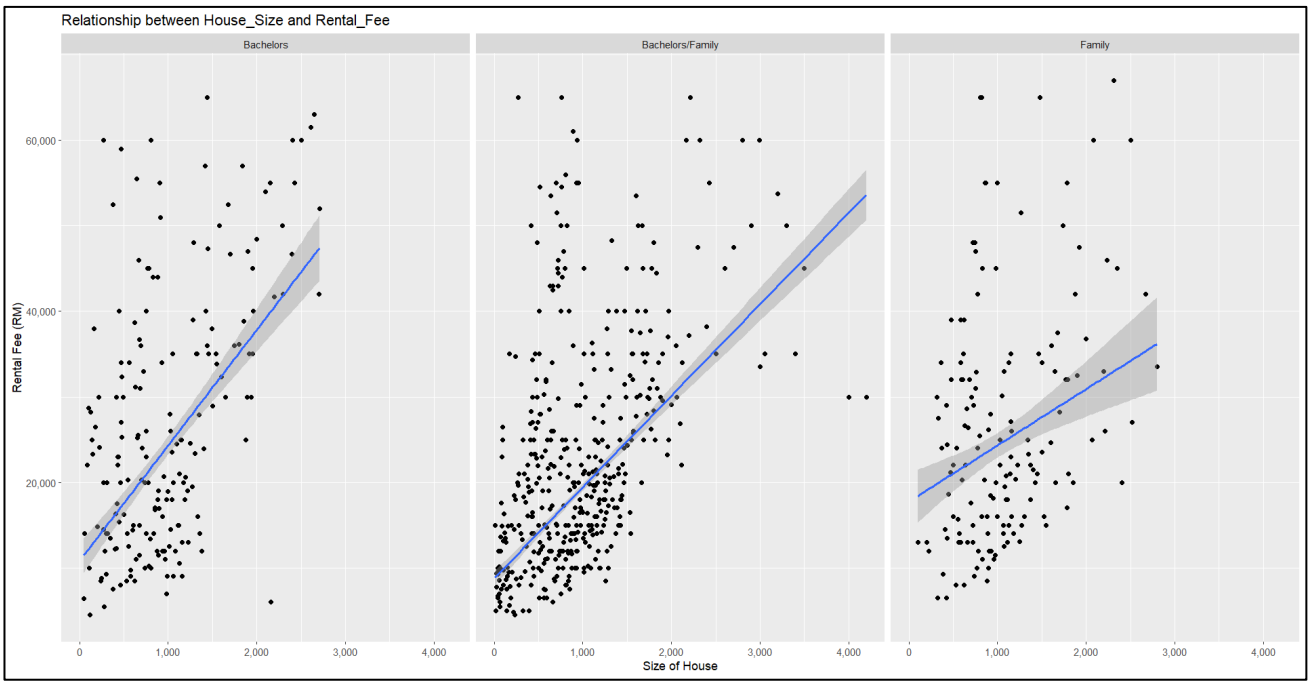



*Figure 6.6.2: Output – Line Graph (Relationship between House_Size and Rental_Fee)*

**Explanation**

From Figure 6.6.2, it can be clearly seen that the house size and rental fee has a positive gradient. The gradient in bachelors' site is the steepest one and it show that it is having a strong relationship, followed by the gradient in bachelors/family site.

**Findings**

- More bachelors prefer to choose house that the house size is smaller than 1500 sqft and average rental fee lower than RM40,000.
- More bachelors/family prefer to choose house that the house size is smaller than 2000 sqft and the rental fee is lower than RM30,000.
- More family prefer to choose house that the house size is smaller than 2000 sqft and the rental fee is lower than RM40,000

## Analysis 6-2: Find the Relationship between House_Size and Furnishing Status

This analysis is used to investigate how tenant choose their house based on house size and furnishing status.

```
# Box Plot
ggplot(house_rental_data,aes(x=House_Size,y=Furnishing_Status)) +
  geom_violin(aes(x=House_Size,y=Furnishing_Status,color=Furnishing_Status)) +
  geom_boxplot(width=0.05,aes(color=Furnishing_Status)) +
  labs(x= "House Size",y="Furnishing Status",
       color="Furnishing Status",
       title="Relationship between House Size and Furnishing Status") +
  scale_x_continuous(breaks=seq(0,8000,500))+
  facet_wrap(~Tenant_Type) +
  theme(legend.position = "bottom")
```

*Figure 6.6.3: Source Code – Violin Plot with Box Plot to Show the Relationship between House_Size and Furnishing_Status*

**Analysis Technique: Data Visualization and Manipulation**

Figure 6.6.3 depicts the source code used to show how tenants choose their house based on house size and furnishing status. The relationship is represented using box plot inside violin plot. **labs()** is used to modify the name of the axes, name of the legend and title plot. **scale_x_**continuous**()** is the position scales for continuous data of x-axis. The x-axis label is modified so that it produce 0 to 8000 with 500 gaps between them in the graph and it is referred to breaks. **facet_wrap()** is utilised to generate graphics tables that show the same graph for each group of tenants. The **theme()** is used to customize the non-data components such as the position of the legend.

```
>   house_rental_data %>%
+     group_by(Tenant_Type,Furnishing_Status) %>% count()
# A tibble: 9 x 3
# Groups:   Tenant_Type, Furnishing_Status [9]
  Tenant_Type      Furnishing_Status     n
  <chr>            <chr>             <int>
1 Bachelors        Furnished            97
2 Bachelors        Semi-Furnished      324
3 Bachelors        Unfurnished         409
4 Bachelors/Family Furnished           508
5 Bachelors/Family Semi-Furnished     1675
6 Bachelors/Family Unfurnished        1261
7 Family           Furnished            75
8 Family           Semi-Furnished      252
9 Family           Unfurnished         145
```

*Figure 6.6.4: Source Code and Output – Count Number of Tenants by Tenant_Type Based on Furnishing_Status*

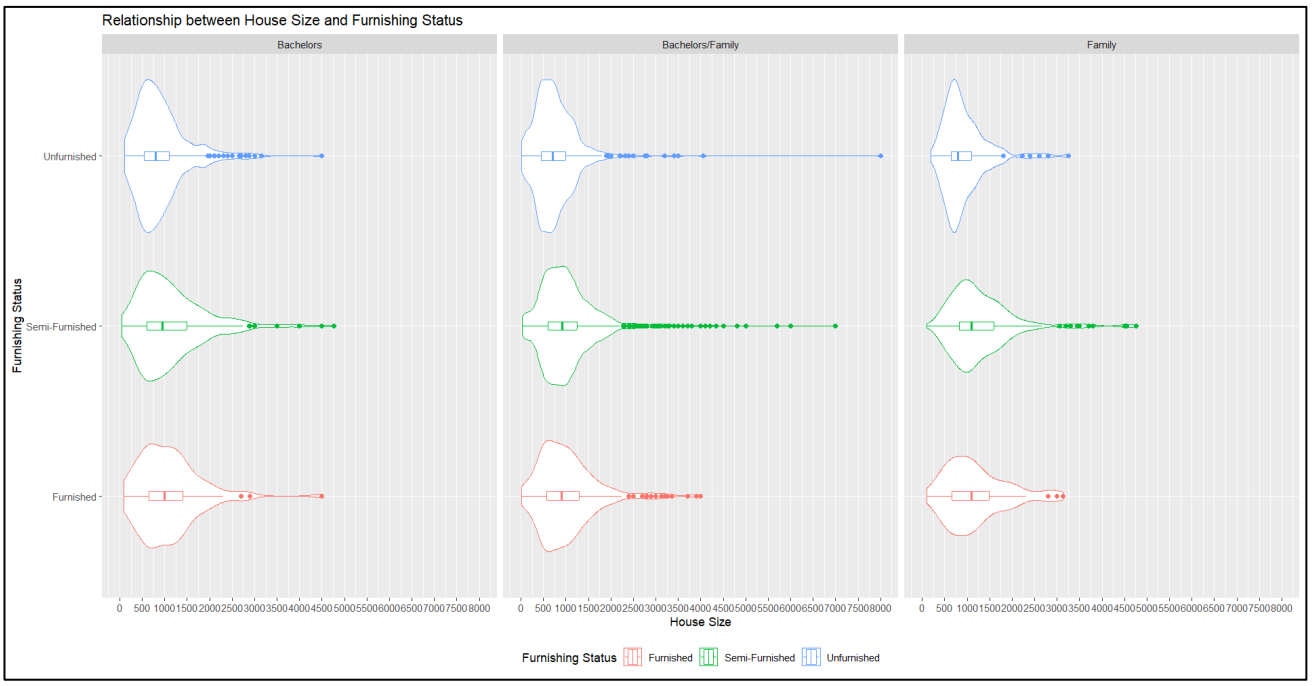



*Figure 6.6.5: Output – Violin Plot with Box Plot (Relationship between House_Size and Furnishing_Status)*

**Explanation**

From Figure 6.6.4 and Figure 6.6.5, more bachelors prefer unfurnished houses with average house size between 600 sqft and 1400 sqft. More bachelors/family prefer semi-furnished houses with average house size between 625 sqft and 1625 sqft. More family prefer semi-furnished houses with average house size between 550 sqft and 1300 sqft.

**Findings**

- More bachelors prefer to choose unfurnished house with house size of 700 sqft.
- More bachelors/family prefer to choose semi-furnished houses with house size of 1000 sqft.
- More family prefer to choose semi-furnished houses with house size of 1000 sqft.

## Analysis 6-3: Find the Relationship between House_Size and Area Type

This analysis is conducted to investigate how tenants choose their house based on house size and area type.

```
# Box Plot
ggplot(house_rental_data,aes(x=House_Size,y=Area_Type)) +
  geom_boxplot(aes(x=House_Size,y=Area_Type,color=Area_Type)) +
  labs(x= "House Size",y="Area Type",
       color="Area Type",
       title="Relationship between House Size and Area Type") +
  scale_x_continuous(breaks=seq(0,8000,500))+
  facet_wrap(~Tenant_Type) +
  theme(legend.position = "bottom")
```

*Figure 6.6.6: Source Code – Box Plot to Show the Relationship between House_Size and Area_Type*

**Analysis Technique: Data Visualization and Manipulation**

Figure 6.6.6 depicts the source code used to show how tenants choose their house based on house size and furnishing status. The relationship is represented using box plot. **labs()** is used to modify the name of the axes, name of the legend and title plot. **scale_x_continuous()** is the position scales for continuous data of x-axis. The x-axis label is modified so that it produce 0 to 8000 with 500 gaps between them in the graph and it is referred to breaks. **facet_wrap()** is utilised to generate graphics tables that show the same graph for each group of tenants. The **theme()** is used to customize the non-data components such as the position of the legend.

```
>   # Calculate Number of Tenants Choose Houses Based on Area_Type
>   house_rental_data %>%
+     group_by(Tenant_Type,Area_Type) %>% count()
# A tibble: 7 x 3
# Groups:   Tenant_Type, Area_Type [7]
  Tenant_Type      Area_Type       n
  <chr>            <chr>       <int>
1 Bachelors        Carpet Area   691
2 Bachelors        Super Area    139
3 Bachelors/Family Built Area      2
4 Bachelors/Family Carpet Area  1281
5 Bachelors/Family Super Area   2161
6 Family           Carpet Area   326
7 Family           Super Area    146
```

*Figure 6.6.7: Source Code and Output – Count of Tenants by Tenant_Type Choose House Based on Area_Type*

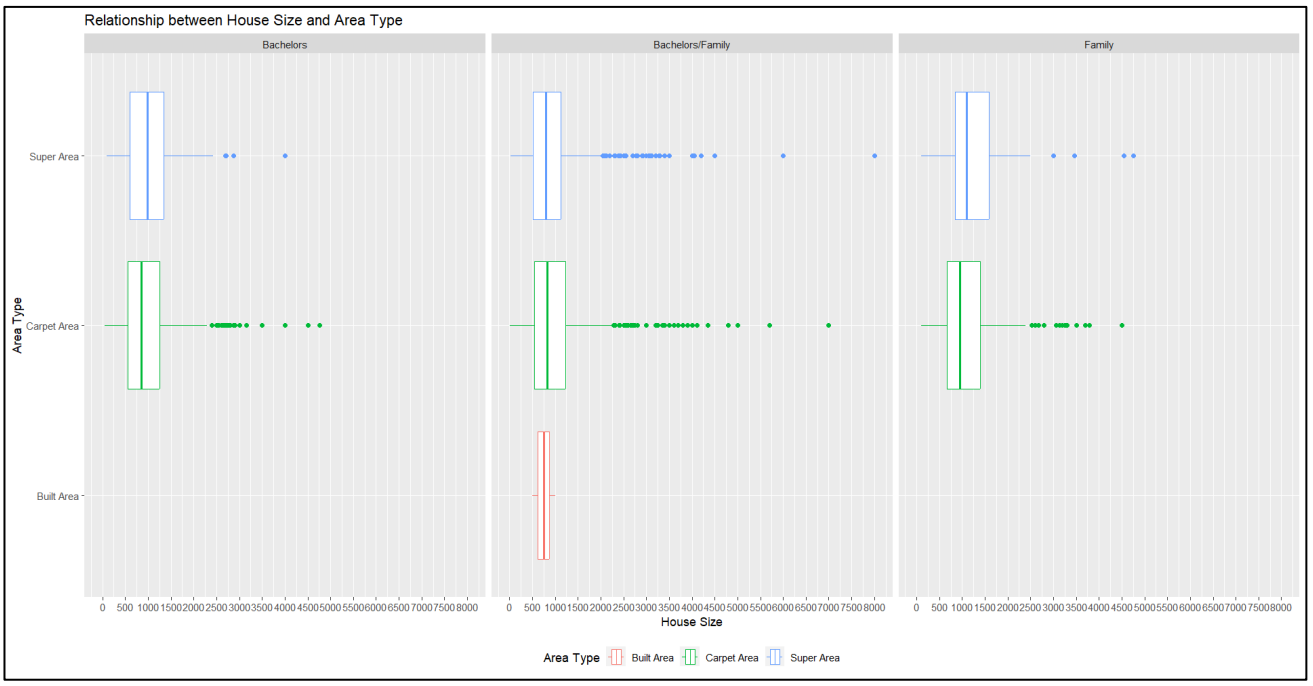



*Figure 6.6.8: Output – Box Plot (Relationship between House_Size and Area_Type*

**Explanation**

From Figure 6.6.7 and Figure 6.6.8, more bachelors prefer to choose house in carpet area with house size between 600 sqft and 1250 sqft. More bachelors/family prefer to choose house in super area with house size between 550 sqft to 1150 sqft. More family prefer to choose house in carpet area with house size between 700 sqft and 1400 sqft.

**Findings**

- More bachelors prefer to choose house in carpet area with house size of 600 sqft.
- More bachelors/family prefer to choose houses in super area with house size of 1000 sqft.
- More family prefer to choose houses in carpet area with house size of 1000 sqft.

## Analysis 6-4: Find the Relationship between House_Size and City

This analysis is conducted to investigate how tenants choose their house based on house size and city.

```
# Violin Plot with Box Plot
ggplot(house_rental_data,aes(x=House_Size,y=City,color=City)) +
  geom_violin() +
  geom_boxplot(width=0.05) +
  labs(x= "House Size",y="City",
       color="City",
       title="Relationship between House Size and City") +
  scale_x_continuous(breaks=seq(0,8000,500)) +
  theme(legend.position = "bottom") +
  facet_wrap(~Tenant_Type)
```

*Figure 6.6.9: Source Code – Violin Plot to Show the Relationship between House_Size and City*

**Analysis Technique: Data Visualization**

Figure 6.6.9 depicts the source code used to create violin plot with box plot inside it to show the number of tenants choose house based on House_Size and City. **labs()** is used to modify the name of the axes label, legend and title plot. **scale_x_continuous()** is the position scales for continuous data of x-axis and can be used to set the breaks from 0 to 8000 with gap of 500 for the label. **Theme()** is used to control the non-data components such as the position of the legend to the bottom. **facet_wrap()** is utilised to generate graphics tables that show the same graph for each group of tenants.

```
>   # Calculate Number of Tenants Choose Houses Based on City
>   house_rental_data %>%
+     group_by(Tenant_Type,City) %>% count()
# A tibble: 18 x 3
# Groups:   Tenant_Type, City [18]
   Tenant_Type      City          n
   <chr>            <chr>     <int>
 1 Bachelors        Bangalore   135
 2 Bachelors        Chennai     137
 3 Bachelors        Delhi       162
 4 Bachelors        Hyderabad   102
 5 Bachelors        Kolkata     122
 6 Bachelors        Mumbai      172
 7 Bachelors/Family Bangalore   694
 8 Bachelors/Family Chennai     649
 9 Bachelors/Family Delhi       432
10 Bachelors/Family Hyderabad   676
11 Bachelors/Family Kolkata     379
12 Bachelors/Family Mumbai      614
13 Family           Bangalore    57
14 Family           Chennai     105
15 Family           Delhi        11
16 Family           Hyderabad    90
17 Family           Kolkata      23
18 Family           Mumbai      186
```

*Figure 6.6.10: Source Code and Output – Count Number of Tenants by Tenant_Type Choose House Based on City*

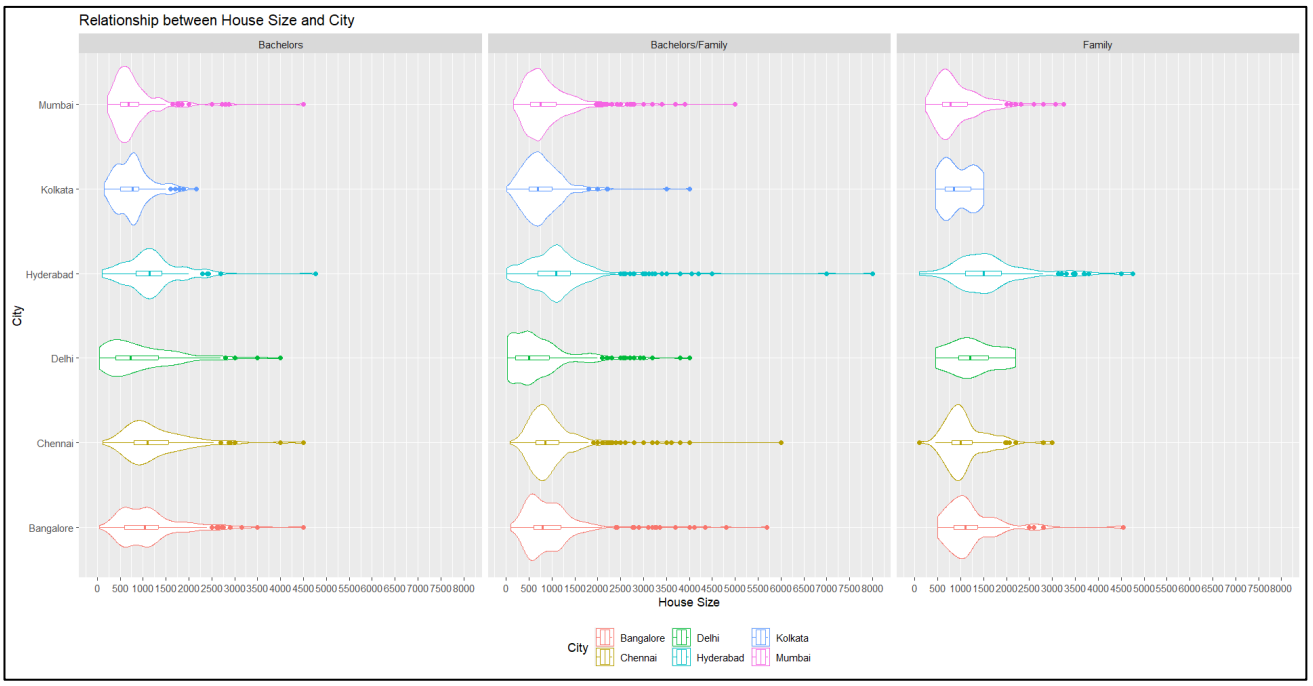



*Figure 6.6.11 – Violin Plot with Box Plot (Relationship between House_Size and City)*

**Explanation**

From Figure 6.6.10 and Figure 6.6.11, more bachelors prefer to choose house in Mumbai with house size between 500 sqft and 900 sqft. More bachelors/family prefer to choose house in Bangalore with house size between 650 sqft and 1200 sqft. More family prefer to choose house in Mumbai with house size between 500 sqft and 1100 sqft.

**Findings**

- More bachelors prefer to choose house in Mumbai with house size of 700 sqft.
- More bachelors/family prefer to choose houses in Bangalore with house size of 600 sqft.
- More family prefer to choose houses in Mumbai with house size of 650 sqft.

## Analysis 6-5: Find the Relationship between House_Size and Point_of_Contact

This analysis is conducted to investigate how tenant choose house based on house size and point of contact.

```
# Box Plot
ggplot(house_rental_data,aes(x=House_Size,y=Point_of_Contact,color=Point_of_Contact)) +
  geom_violin() +
  geom_boxplot(width=0.05)+
  labs(x= "Average House Size",y="Point of Contact",
       color="City",
       title="Relationship between House Size and Point of Contact") +
  scale_x_continuous(breaks=seq(0,8000,500)) +
  facet_wrap(~Tenant_Type) +
  theme(legend.position = "bottom")
```

*Figure 6.6.12: Source Code - Violin Plot with Box Plot to Show the Relationship between House_Size and Point_of_Contact*

**Analysis Technique: Data Visualization**

Figure 6.6.12 depicts the source code used to create violin plot with box plot inside it to show the number of tenants choose house based on House_Size\ and Point_of_Contact. **labs()** is used to modify the name of the axes label, legend, and title plot. **scale_x_continuous()** is the position scales for continuous data of x-axis and can be used to set the breaks from 0 to 8000 with gap of 500 for the label. **facet_wrap()** is utilised to generate graphics tables that show the same graph for each group of tenants. **theme()** is used to control the non-data components such as the position of the legend to the bottom.

```
>   # Count Number of Tenants Grouped by Point_Of_Contact
>   house_rental_data %>%
+     group_by(Tenant_Type,Point_of_Contact) %>%
+     count()
# A tibble: 7 x 3
# Groups:   Tenant_Type, Point_of_Contact [7]
  Tenant_Type      Point_of_Contact     n
  <chr>            <chr>            <int>
1 Bachelors        Contact Agent      434
2 Bachelors        Contact Owner      396
3 Bachelors/Family Contact Agent      849
4 Bachelors/Family Contact Builder      1
5 Bachelors/Family Contact Owner     2594
6 Family           Contact Agent      246
7 Family           Contact Owner      226
```

*Figure 6.6.13: Source Code and Output – Count Number of Tenants Grouped by Tenant_Type and Point_of_Contact*

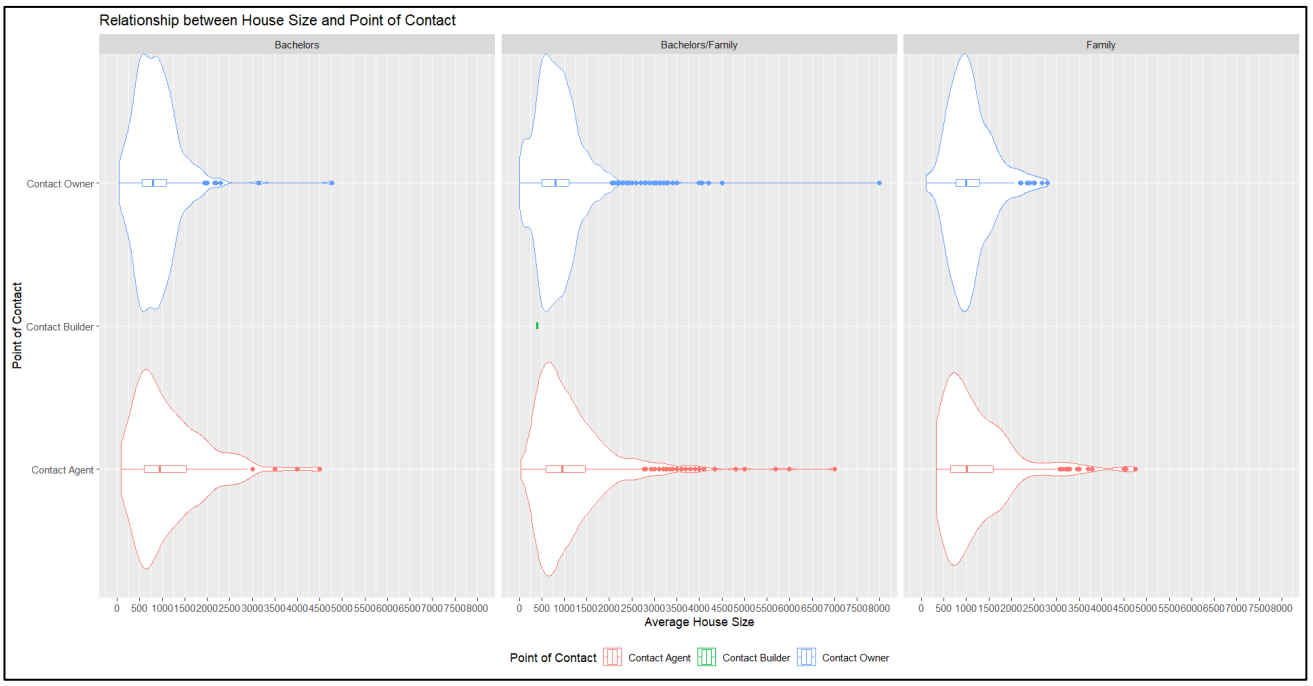



*Figure 6.6.14: Output – Violin Plot with Box Plot (Relationship between House_Size and Point_of_Contact)*

**Explanation**

From Figure 6.6.13 and Figure 6.6.14, more bachelors prefer to contact agent to choose house with house size between 600 sqft and 1600 sqft. More bachelors/family prefer to contact agent to choose house with house size between 600 sqft and 1500 sqft. More family prefer to contact owner to choose house with house size between 775 sqft and 1300 sqft.

**Findings**

- More bachelors prefer to contact agent to choose house with house size of 600 sqft.
- More bachelors/family prefer to contact agent to choose house with house size of 650 sqft.
- More family prefer to contact owner to choose house with house size of 1000 sqft.

## Analysis 6-6: Find the Relationship between House_Size, Bedroom_Hall_Kitchen and Furnishing_Status

This analysis is conducted to investigate how tenant choose their house based on the size of the house, the number of bedrooms, hall, and kitchen and the furnishing status of the house.

```
# Violin Plot with Box Plot
ggplot(house_rental_data,aes(x=House_Size,y=factor(Bedroom_Hall_Kitchen),
                             color=factor(Bedroom_Hall_Kitchen))) +
  geom_violin() +
  geom_boxplot(width=0.05) +
  labs(x= "House Size",y="Number of Bedroom, Hall and Kitchen",
       color="Number of Bedroom, Hall and Kitchen",
       title="Relationship between House_Size, Number of Bedroom_Hall_Kitchen and Furnishing_Status") +
  scale_x_continuous(breaks=seq(0,8000,500)) +
  theme(legend.position = "bottom") +
  facet_grid(Tenant_Type~Furnishing_Status)
```

*Figure 6.6.15: Source Code – Bar Chart to Show the Relationship between House_Size, Bedroom_Hall_Kitchen and Furnishing_Status*

**Analysis Technique: Data Visualization**

Figure 6.6.15 depicts the source code used to show the number of tenants choose house based on House_Size, Bedroom_Hall_Kitchen and Furnishing_Status. This relationship is represented using violin plot with box plot inside it. **Labs()** is used to modify the name of the axes, title, and the name of legend. **Scale_x_continuous()** is the position scales for continuous data of x-axis. The x-axis label is modified so that it produce 0 to 8000 with 500 gaps between them in the graph and it is referred to breaks. **theme()** is used to control the non-data components such as the position of the legend to the bottom. **facet_grid()** generates a 2D grid of panels with the rows and columns given by Tenant_Type and Furnishing_Status.

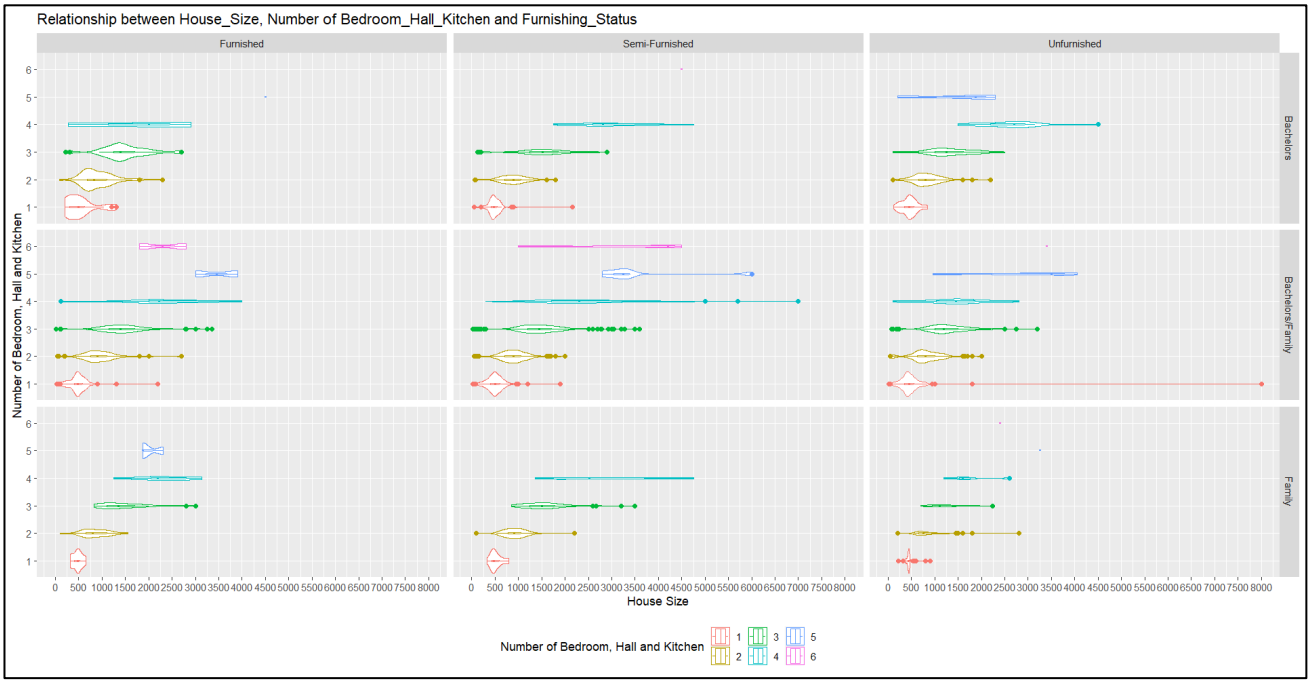



*Figure 6.6.16: Output – Violin Plot with Box Plot (Relationship between House_Size, Bedroom_Hall_Kitchen and Furnishing_Status)*

**Explanation**

From previous Analysis 3-3, Analysis 3-5, Analysis 6-2 and Figure 6.6.16, more bachelors prefer unfurnished houses with 2 bedrooms, hall, and kitchen and house size between 600sqft and 1000 sqft. More bachelors/family prefer semi-furnished houses with 2 bedrooms, hall, and kitchen and house size between 700 sqft and 1100 sqft. More family prefer semi-furnished houses with 2 bedrooms, hall, and kitchen and house size between 750 sqft and 1100 sqft.

**Findings**

- More bachelors prefer unfurnished houses with 2 bedrooms, hall, and kitchen and house size of 700 sqft.
- More bachelors/family prefer semi-furnished houses with 2 bedrooms, hall, and kitchen and house size of 1000 sqft.
- More family prefer semi-furnished houses with 2 bedrooms, hall, and kitchen and house size of 1000 sqft

## Analysis 6-7: Find the Relationship between House_Size, Number of Bedroom_Hall_Kitchen and Area_Type

This analysis is conducted to investigate how tenant choose their house based on the size of the house, the number of bedrooms, hall, and kitchen and the furnishing status of the house.

```
# Violin Plot with Box Plot
ggplot(house_rental_data,aes(x=House_Size,y=factor(Bedroom_Hall_Kitchen),
                             color=factor(Bedroom_Hall_Kitchen))) +
  geom_violin() +
  geom_boxplot(width=0.05) +
  labs(x= "House Size",y="Number of Bedroom, Hall and Kitchen",
       color="City",
       title="Relationship between House_Size, Number of Bedroom_Hall_Kitchen and Area_Type") +
  scale_x_continuous(breaks=seq(0,8000,500)) +
  theme(legend.position = "bottom") +
  facet_grid(Tenant_Type~Area_Type)
```

*Figure 6.6.17: Source Code – Bar Chart to Show the Relationship between House_Size, Bedroom_Hall_Kitchen and Area_Type*

**Analysis Technique: Data Visualization**

Figure 6.6.17 depicts the source code used to show the number of tenants choose house based on House_Size, Bedroom_Hall_Kitchen and Area_Type. This relationship is represented using violin plot with box plot inside it. **Labs()** is used to modify the name of the axes, title and the name of legend. **Scale_x_continuous()** is the position scales for continuous data of x-axis. The x-axis label is modified so that it produce 0 to 8000 with 500 gaps between them in the graph and it is referred to breaks. **theme()** is used to control the non-data components such as the position of the legend to the bottom. **facet_grid()** generates a 2D grid of panels with the rows and columns given by Tenant_Type and Area_Type.

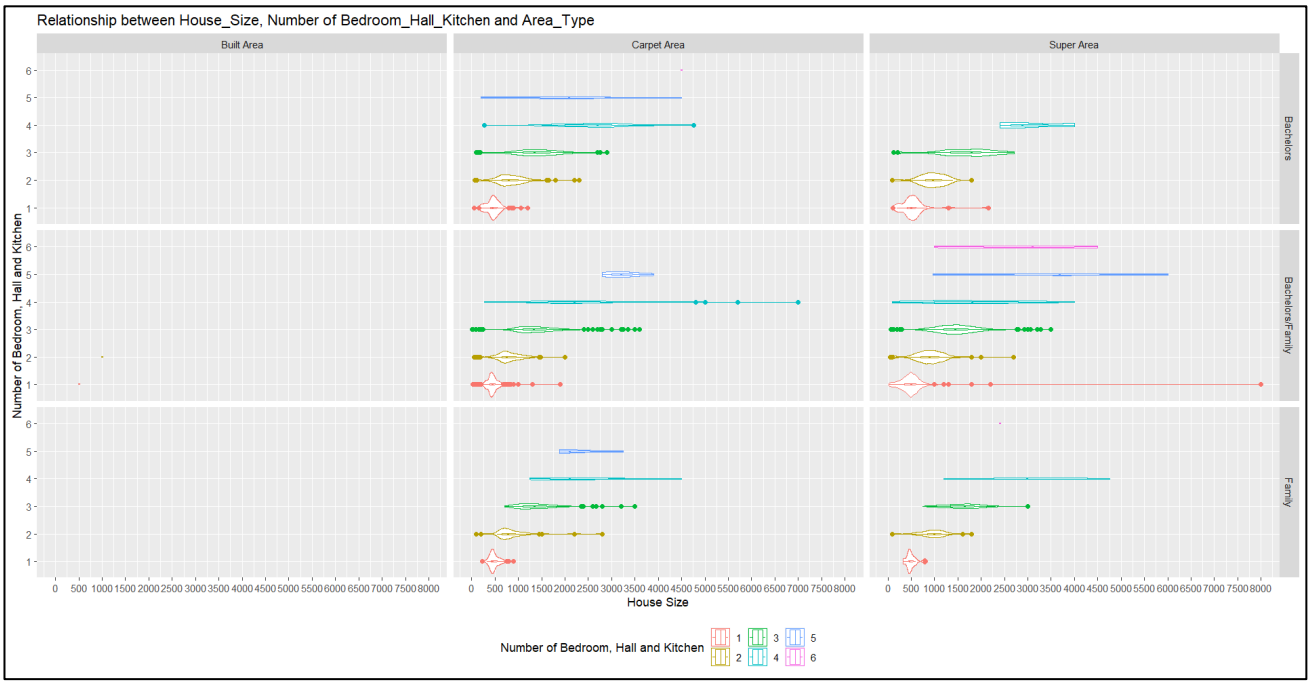



*Figure 6.6.18: Output – Violin Plot with Box Plot (Relationship between House_Size, Bedroom_Hall_Kitchen and Area_Type)*

**Explanation**

From previous Analysis 3-3, Analysis 3-6, Analysis 6-3 and Figure 6.6.18, more bachelors prefer houses located in carpet area with 2 bedrooms, hall, and kitchen and house size between 650 sqft and 1000 sqft. More bachelors/family prefer houses located in super area with 2 bedrooms, hall, and kitchen and house size between 700 sqft and 1150 sqft. More family prefer houses located in carpet area with 2 bedrooms, hall, and kitchen and house size between 650 sqft and 900 sqft.

**Findings**

- More bachelors prefer houses located in carpet area with 2 bedrooms, hall, and kitchen and house size of 700 sqft
- More bachelors/family prefer houses located in super area with 2 bedrooms, hall, and kitchen and house size of 1000 sqft.
- More family prefer houses located in carpet area with 2 bedrooms, hall, and kitchen and house size of 800 sqft.

## Analysis 6-8: Find the Relationship between House_Size, Number_of_Bathroom and Furnishing_Status

This analysis is conducted to investigate how tenant choose their house based on the size of the house, the number of bathrooms and the furnishing status of the house.

```
# Violin Plot with Box Plot
ggplot(house_rental_data,aes(x=House_Size,y=factor(Number_of_Bathroom),
                             color=factor(Number_of_Bathroom))) +
  geom_violin() +
  geom_boxplot(width=0.05) +
  labs(x= "House Size",y="Number of Bathroom",
       color="Number of Bathroom",
       title="Relationship between House_Size, Number_of_Bathroom and Furnishing_Status") +
  scale_x_continuous(breaks=seq(0,8000,500)) +
  theme(legend.position = "bottom") +
  facet_grid(Tenant_Type~Furnishing_Status)
```

*Figure 6.6.19: Source Code – Bar Chart to Show the Relationship between House_Size, Number_of_Bathroom and Furnishing_Status*

**Analysis Technique: Data Visualization**

Figure 6.6.19 depicts the source code used to show the number of tenants choose house based on House_Size, Number_of_Bathroom and Furnishing_Status. This relationship is represented using violin plot with box plot inside it. **labs()** is used to modify the name of the axes, title and the name of legend. **scale_x_continuous()** is the position scales for continuous data of x-axis. The x-axis label is modified so that it produce 0 to 8000 with 500 gaps between them in the graph and it is referred to breaks. **theme()** is used to control the non-data components such as the position of the legend to the bottom. **facet_grid()** generates a 2D grid of panels with the rows and columns given by Tenant_Type and Furnishing_Status.

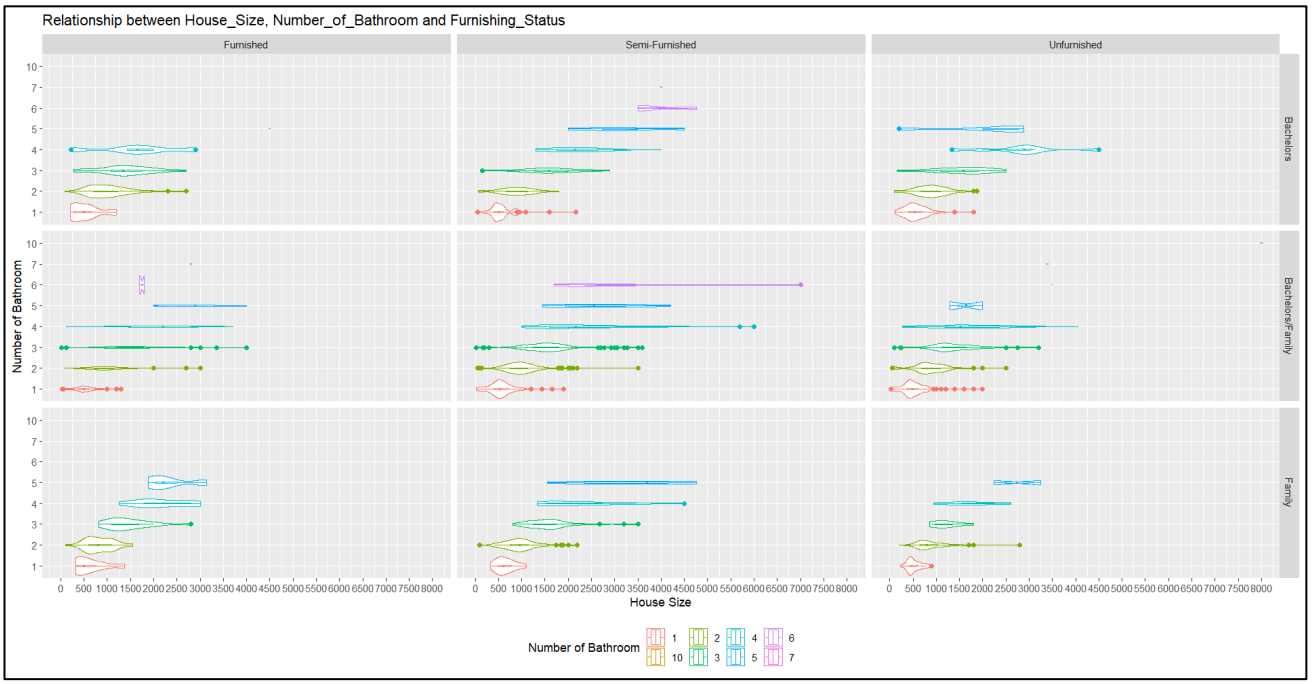



*Figure 6.6.20: Output – Violin Plot with Box Plot (Relationship between House_Size, Number_of_Bathroom and Furnishing_Status)*

**Explanation**

From Analysis 4-6 and Figure 6.6.20, More bachelors prefer unfurnished houses with 2 bathrooms and house size between 700 sqft and 1100 sqft. More bachelors/family prefer semi-furnished houses with 2 bathrooms and house size between 750 sqft and 1200 sqft. More family prefer semi-furnished houses with 2 bathrooms and house size between 750 sqft and 1200 sqft.

**Findings**

- More bachelors prefer unfurnished houses with 2 bathrooms and house size of 1000 sqft
- More bachelors/family prefer semi-furnished houses with 2 bathrooms and house size of 1000 sqft.
- More family prefer semi-furnished houses with 2 bathrooms and house size of 1000 sqft.

## Conclusion for Question 6

1. More tenants prefer to choose house with house size of smaller than 2000 sqft and in the range of RM30,000 and RM40,000.
2. More tenants prefer to choose semi-furnished houses with house size of 1100 sqft.
3. More tenants prefer to choose houses in carpet area with house size of 1000 sqft.
4. More tenants prefer to choose houses in Mumbai with house size of 750 sqft
5. More tenants prefer to contact owner to choose house with house size of 1000 sqft.
6. More tenants prefer semi-furnished houses with 2 bedrooms, hall, and kitchen and house size of 1000 sqft.
7. More tenants prefer houses located in super area with 2 bedrooms, hall, and kitchen and house size of 1000 sqft.
8. More tenants prefer semi-furnished houses with 2 bathrooms and house size of 1000 sqft.

## Question 7: What are the Factors influencing Tenants to Choose Their Houses with respect to Rental Fee?

### Analysis 7-1: Find the Relationship between Rental_Fee and Furnishing_Status

This analysis is conducted to investigate how tenants choose their house based on the furnishing status and rental fee.

```
# Violin Plot with Box Plot to Show Tenant Preference
ggplot(rf_no_outliers,aes(x=Rental_Fee,y=Furnishing_Status,
                          color=Furnishing_Status)) +
  geom_violin() +
  geom_boxplot(width=0.05) +
  labs(x= "Rental Fee",y="Furnishing Status",
       color="Furnishing Status",
       title="Relationship between Rental Fee and Furnishing Status") +
  scale_x_continuous(breaks=seq(0,70000,5000),
                     labels = label_number(suffix = "k", scale = 1e-3)) +
  theme(legend.position = "bottom") +
  facet_wrap(~Tenant_Type)
```

*Figure 6.7.1: Source Code – Bar Chart to Show the Relationship between Rental_Fee and Furnishing_Status*

**Analysis Technique: Data Visualization and Manipulation**

Figure 6.7.1 depicts the source code used to show how tenants choose their house based on Furnishing_Status and Rental_Fee. The outliers of rental fee have been removed. **Labs()** is used to modify the name of the axes, the name of the legend and title. **scale_x_continuous()** is the position scales for continuous data of x-axis. The x-axis label is modified so that it produce 0 to 70000 with 5000 gaps between them in the graph and it is referred to breaks. The x-axis label is modified so that thousands can be represented as k in the label. **theme()** is used to control the non-data components such as the position of the legend to the bottom. **facet_wrap()** is utilised to generate graphics tables that show the same graph for each group of tenants.

```
>   # Count Number of Tenants Based on Furnishing_Status
>   house_rental_data %>%
+     group_by(Tenant_Type,Furnishing_Status) %>%
+     count()
# A tibble: 9 x 3
# Groups:   Tenant_Type, Furnishing_Status [9]
  Tenant_Type      Furnishing_Status     n
  <chr>            <chr>             <int>
1 Bachelors        Furnished            97
2 Bachelors        Semi-Furnished      324
3 Bachelors        Unfurnished         409
4 Bachelors/Family Furnished           508
5 Bachelors/Family Semi-Furnished     1675
6 Bachelors/Family Unfurnished        1261
7 Family           Furnished            75
8 Family           Semi-Furnished      252
9 Family           Unfurnished         145
```

*Figure 6.7.2: Source Code and Output – Count Number of Tenants by Tenant_Type Based on Furnishing_Status*

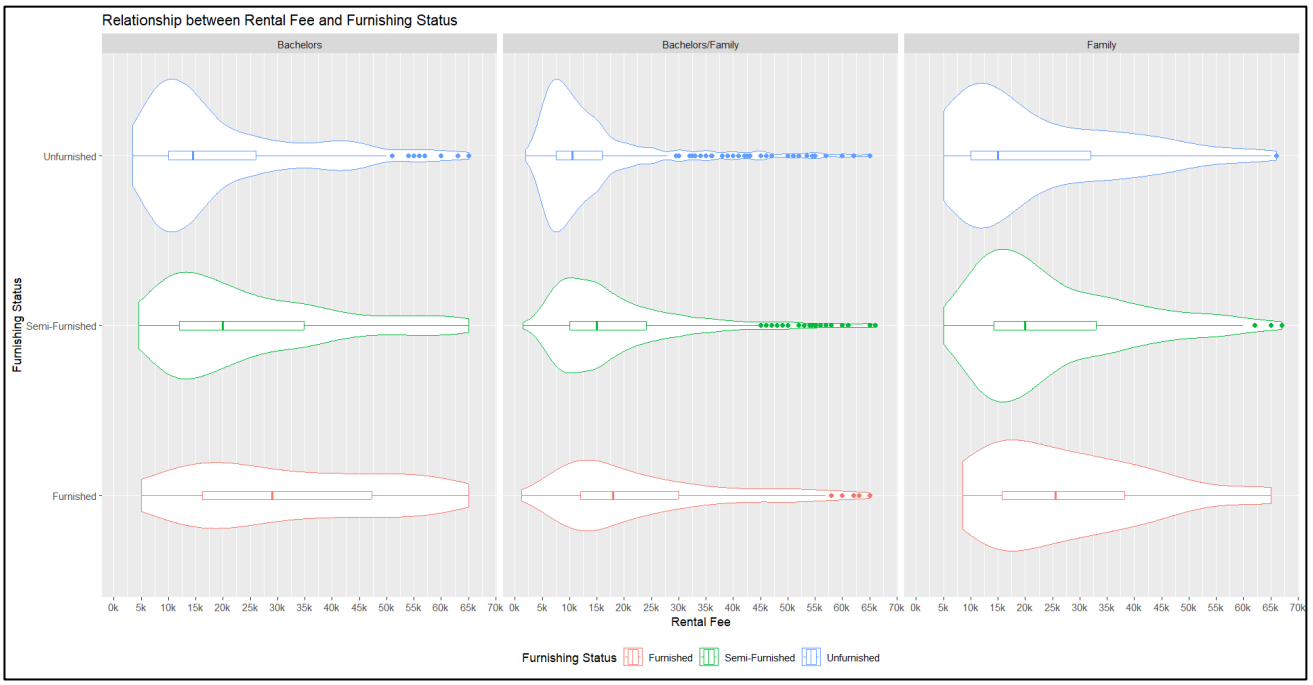



*Figure 6.7.3: Output – Bar Chart (Relationship between Rental_Fee and Furnishing_Status)*

**Explanation**

From Figure 6.7.2 and Figure 6.7.3, more bachelors prefer unfurnished houses with rental fee between RM10,000 and RM26,000. More bachelors/family prefer semi-furnished houses with rental fee between RM10,000 and RM24,000. More family prefer semi-furnished houses with rental fee between RM12,000 and RM30,000.

**Findings**

- More bachelors prefer unfurnished houses with rental fee of RM10,000.
- More bachelors/family prefer semi-furnished houses with rental fee of RM10,000.
- More family prefer semi-furnished houses with rental fee of RM15,000.

## Analysis 7-2: Find the Relationship between Rental_Fee and Area_Type

This analysis is conducted to investigate how much is the rental fee that tenants prefer to choose their house based on area type.

```
# Bar Chart
ggplot(rf_no_outliers,aes(factor(Area_Type),Rental_Fee)) +
  geom_bar(aes(Area_Type,Rental_Fee,fill=as.factor(Area_Type)),
           position="dodge",stat="summary",fun="mean",width=0.5) +
  stat_summary(aes(label=round(..y..,2)),fun="mean",geom="text",vjust=-0.3,size=4) +
  facet_wrap(~Tenant_Type) +
  scale_y_continuous(labels=scales::comma) +
  labs(x="Area Type",y="Average Rental Fee (RM)") +
  scale_fill_discrete(name="Area Type") +
  theme(legend.position="bottom") +
  ggtitle("Relationship between Rental_Fee and Area_Type")
```

*Figure 6.7.4: Source Code – Bar Chart to Show the Relationship between Rental_Fee and Area_Type*

**Analysis Technique: Data Visualization and Manipulation**

Figure 6.7.4 depicts the source code used to show the amount of rental fee based on area type. This relationship is calculated by find the mean using stat = "summary" and fun = "mean". **facet_wrap()** is utilised to generate graphics tables that show the same graph for each group of tenants. **scale_y_continuous()** is the position scales for continuous data of y-axis. The y-axis label is modified so that it do not show e notation in the graph, and it is referred to labels. **Labs()** is used to modify the name of the axes and title. **scale_fill_discrete(name =)** is used to modify the name of the scale. **Theme()** is used to control the non-data components such as the position of the legend to the bottom. Lastly, the title name is set using **ggtitle()**.

```
>   # Count Number of Tenants by Tenant_Type
>     # Based on Area Type
>   house_rental_data %>%
+     group_by(Tenant_Type,Area_Type)%>%
+     count()
# A tibble: 7 x 3
# Groups:   Tenant_Type, Area_Type [7]
  Tenant_Type      Area_Type       n
  <chr>            <chr>       <int>
1 Bachelors        Carpet Area   691
2 Bachelors        Super Area    139
3 Bachelors/Family Built Area      2
4 Bachelors/Family Carpet Area  1281
5 Bachelors/Family Super Area   2161
6 Family           Carpet Area   326
7 Family           Super Area    146
```

*Figure 6.7.5: Source Code and Output – Count Number of Tenants by Tenant_Type Based on Area_Type*

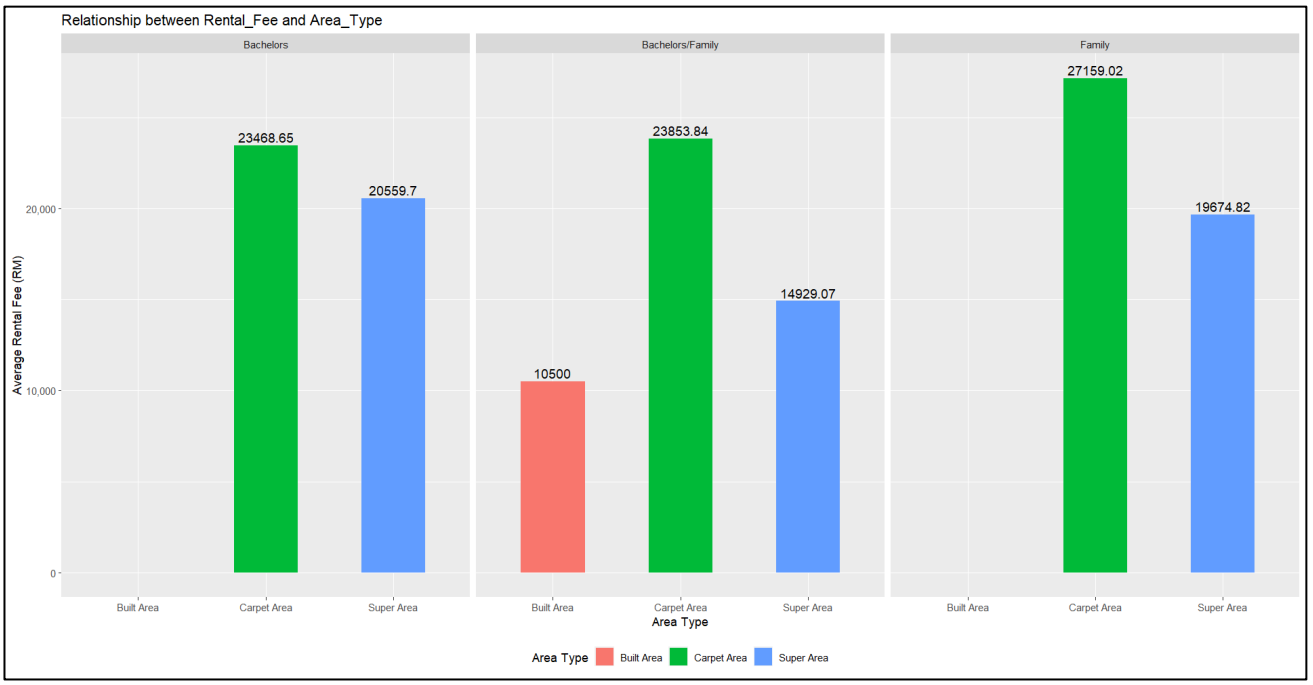



*Figure 6.7.6: Output – Bar Chart (Relationship between Rental_Fee and Area_Type)*

**Explanation**

From Figure 6.7.5 and Figure 6.7.6, carpet area has the highest average rental fee among the three area, followed by super area. None of the bachelors or the family prefer to choose houses in built area.

**Findings**

- More bachelors prefer to choose houses in carpet area with average rental fee of RM23,468.65.
- More bachelors/family prefer to choose houses in super area with average rental fee of RM14,929.07.
- More family prefer to choose houses in carpet area with average rental fee of RM27,159,02.

## Analysis 7-3: Find the Relationship between Rental_Fee and City

This analysis is conducted to investigate how is the rental fee in each city and how tenants choose their houses based on rental fee and city.

```
# Bar Chart
ggplot(rf_no_outliers,aes(factor(City),Rental_Fee)) +
  geom_bar(aes(City,Rental_Fee,fill=as.factor(City)),
           position="dodge",stat="summary",fun="mean") +
  stat_summary(aes(label=round(..y..,2)),fun.y="mean",geom="text",vjust=-0.5,size=3) +
  scale_y_continuous(labels=scales::comma) +
  labs(x="City",y="Average Rental Fee (RM)") +
  scale_fill_discrete(name="City") +
  facet_wrap(~Tenant_Type) +
  theme(legend.position = "bottom") +
  ggtitle("Relationship between Rental_Fee and City")
```

*Figure 6.7.7: Source Code – Bar Chart to Show the Relationship between Rental_Fee and City*

**Analysis Technique: Data Visualization and Manipulation**

Figure 6.7.7 depicts the source code used to show the amount of rental fee based on city and show how tenants choose their houses based on rental fee and city. This relationship is calculated by find the mean using stat = "summary" and fun = "mean". **scale_y_continuous()** is the position scales for continuous data of y-axis. The y-axis label is modified so that it do not show e notation in the graph, and it is referred to labels. **Labs()** is used to modify the name of the axes and title. **scale_fill_discrete(name =)** is used to modify the name of the scale. **facet_wrap()** is utilised to generate graphics tables that show the same graph for each group of tenants. **theme()** is used to control the non-data components such as the position of the legend to the bottom. Lastly, the title name is set using **ggtitle()**.

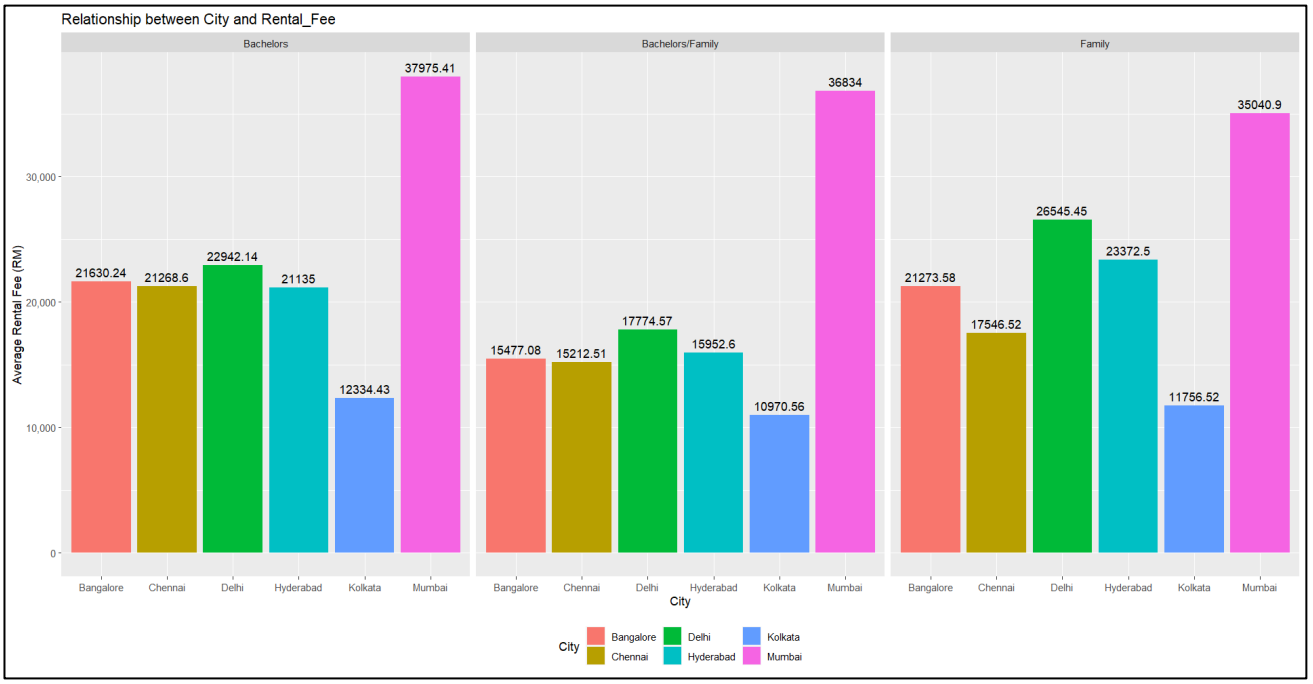



*Figure 6.7.8: Output – Bar Chart (Relationship between Rental_Fee and City)*

```
>   # Count Number of Tenants by Tenant_Type Based on City
>   house_rental_data %>% group_by(Tenant_Type,City) %>%
+     count()
# A tibble: 18 × 3
# Groups:   Tenant_Type, City [18]
   Tenant_Type      City          n
   <chr>            <chr>     <int>
 1 Bachelors        Bangalore   135
 2 Bachelors        Chennai     137
 3 Bachelors        Delhi       162
 4 Bachelors        Hyderabad   102
 5 Bachelors        Kolkata     122
 6 Bachelors        Mumbai      172
 7 Bachelors/Family Bangalore   694
 8 Bachelors/Family Chennai     649
 9 Bachelors/Family Delhi       432
10 Bachelors/Family Hyderabad   676
11 Bachelors/Family Kolkata     379
12 Bachelors/Family Mumbai      614
13 Family           Bangalore    57
14 Family           Chennai     105
15 Family           Delhi        11
16 Family           Hyderabad    90
17 Family           Kolkata      23
18 Family           Mumbai      186
```

*Figure 6.7.9: Source Code and Output – Count Number of Tenants by Tenant_Type Based on City*

**Explanation**

From Figure 6.7.8, it is clearly seen that Mumbai has the highest rental fee in each group of tenants. From Figure 6.7.9, more bachelors prefer to choose houses located in Mumbai. More bachelors/family prefer to choose houses located in Bangalore. More family prefer to choose house located in Mumbai.

**Findings**

- More bachelors prefer to choose houses located in Mumbai with average rental fee of RM37,975,41.
- More bachelors/family prefer to choose houses located in Bangalore with average rental fee of RM15,477,08.
- More family prefer to choose house located in Mumbai with average rental fee of RM35,040.90.

## Analysis 7-4: Find the Relationship between Rental Fee and Point of Contact

This analysis is conducted to investigate how tenants choose house based on rental fee and point of contact.

```
# Violin Plot with Box Plot
ggplot(rf_no_outliers,aes(x=Point_of_Contact,y=Rental_Fee)) +
  geom_violin(aes(x=Point_of_Contact,y=Rental_Fee,color=Point_of_Contact)) +
  geom_boxplot(width=0.05,aes(color=Point_of_Contact)) +
  labs(x= "Point of Contact",y="Rental Fee",
       color="Point of Contact",
       title="Relationship between Rental Fee and Point of Contact") +
  scale_y_continuous(breaks=seq(0,70000,5000),
                     labels = label_number(suffix = "k", scale = 1e-3)) +
  theme(legend.position = "bottom") +
  facet_wrap(~Tenant_Type)
```

*Figure 6.7.10: Source Code – Violin Plot with Box Plot to Show the Relationship between Rental_Fee and Point_of_Contact*

**Analysis Technique: Data Visualization and Manipulation**

Figure 6.7.10 depicts the source code used to create violin plot with box plot inside it to show how tenants choose house based on Rental_Fee and Point_of_Contact. **labs()** is used to modify the name of the axes label, legend, and title plot. **scale_y_continuous()** is the position scales for continuous data of y-axis and can be used to set the breaks from 0 to 70000 with gaps of 5000 for the label. It also can be used to represent thousands as k in the labels. **Theme()** is used to control the non-data components such as the position of the legend to the bottom. **facet_wrap()** is utilised to generate graphics tables that show the same graph for each group of tenants.

```
>   # Count Number of Tenants by Tenant_Type Based on Point_of_Contact
>   house_rental_data%>%group_by(Tenant_Type,Point_of_Contact)%>%count()
# A tibble: 7 x 3
# Groups:   Tenant_Type, Point_of_Contact [7]
  Tenant_Type      Point_of_Contact     n
  <chr>            <chr>            <int>
1 Bachelors        Contact Agent      434
2 Bachelors        Contact Owner      396
3 Bachelors/Family Contact Agent      849
4 Bachelors/Family Contact Builder      1
5 Bachelors/Family Contact Owner     2594
6 Family           Contact Agent      246
7 Family           Contact Owner      226
```

*Figure 6.7.11: Source Code and Output – Count Number of Tenants by Tenant_Type Based on Point_of_Contact*

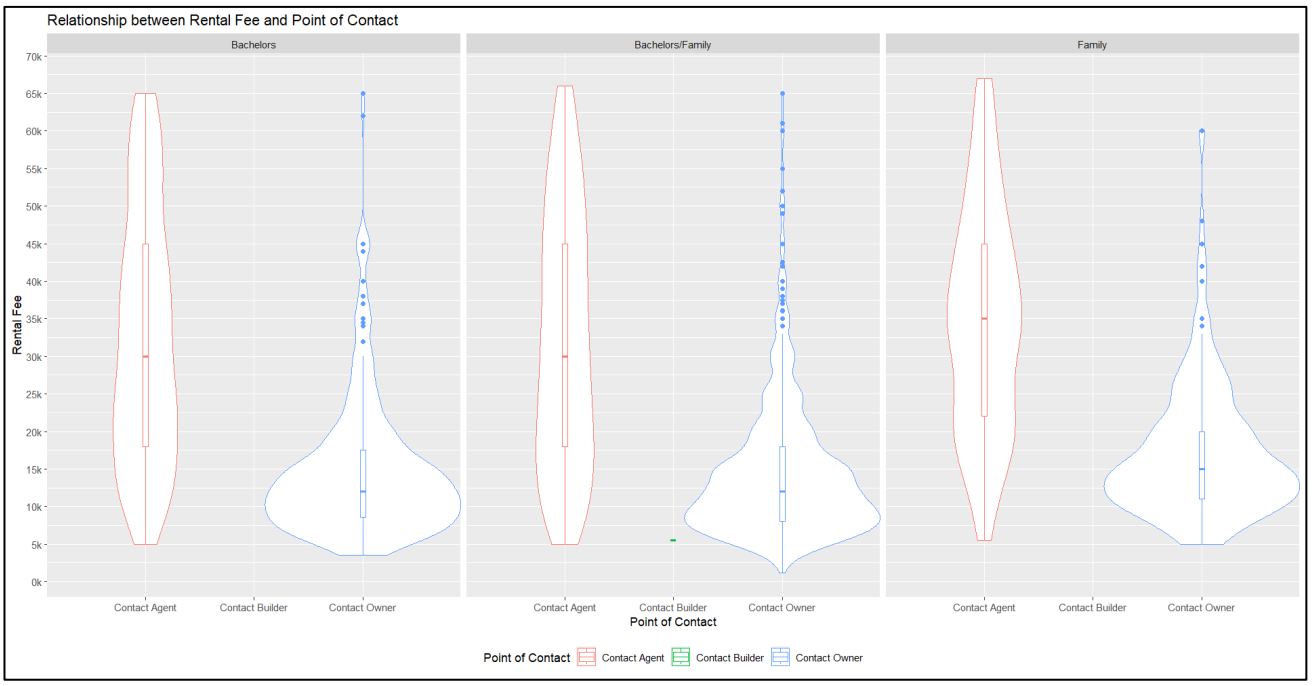



*Figure 6.7.12: Output – Violin Plot with Box Plot (Relationship between Rental_Fee and Point_of_Contact)*

**Explanation**

From Figure 6.7.11 and Figure 6.7.12, more bachelors prefer to contact agent to choose houses with rental fee between RM17,000 and RM45,000. More bachelors/family prefer to contact owner to choose houses with rental fee between RM7000 and RM17,500. More family prefer to contact agent to choose houses with rental fee between RM22,000 and RM45,000.

**Findings**

- More bachelors prefer to contact agent to choose houses with average rental fee of around RM25,000.
- More bachelors/family prefer to contact owner to choose houses with average rental fee of around RM15,000.
- More family prefer to contact agent to choose houses with average rental fee of around RM20,000.

## Analysis 7-5: Find the Relationship between Rental Fee, Number of Bedroom_Hall_Kitchen and Furnishing_Status

This analysis is conducted to investigate how tenants choose house based on rental fee, number of bedrooms, hall, and kitchen and the furnishing status.

```
# Bar Chart
ggplot(rf_no_outliers,aes(x=Bedroom_Hall_Kitchen,y=Rental_Fee,fill=Furnishing_Status)) +
  geom_bar(stat="summary") +
  stat_summary(aes(label=round(..y..,2)),size=3,fun.y="mean",geom="text",vjust=2) +
  scale_x_continuous(breaks=seq(1,6,1)) +
  scale_y_continuous(labels=scales::comma) +
  facet_grid(Tenant_Type~Furnishing_Status) +
  theme(legend.position = "bottom") +
  labs(x="Number of Bedroom, Hall and Kitchen",y="Average Rental Fee",
       color="Number of Bedroom, Hall and Kitchen",
       title="Relationship between Average Rental Fee, Number of Bedroom_Hall_Kitchen and Furnishing_Status")
```

*Figure 6.7.13: Source Code – Bar Chart to Show the Relationship between Rental_Fee, Bedroom_Hall_Kitchen and Furnishing_Status*

**Analysis Technique: Data Visualization and Manipulation**

Figure 6.7.13 depicts the source code used to show how tenants choose house based on rental fee, number of bedrooms, hall, and kitchen and the furnishing status. This relationship is calculated by find the mean using stat = "summary". **stat_summary()** can be used to add mean points to a bar chart. **Geom="text"** is entered so that to add the mean labels into the bar chart. **Scale_x_continuous()** is the position scales for continuous data of x-axis, The x-axis label is modified so that it produce 1 to 6 with 1 space between them in the graph and it is referred to breaks. **Scale_y_continuous()** is the position scales for continuous data of y-axis. The y-axis label is modified so that it do not show e notation in the graph, and it is referred to labels. **facet_grid()** generates a 2D grid of panels with the rows and columns given by Tenant_Type and Furnishing_Status. **theme()** is used to control the non-data components such as the position of the legend to the bottom. **Labs()** is used to modify the name of the axes, name of legend and title.

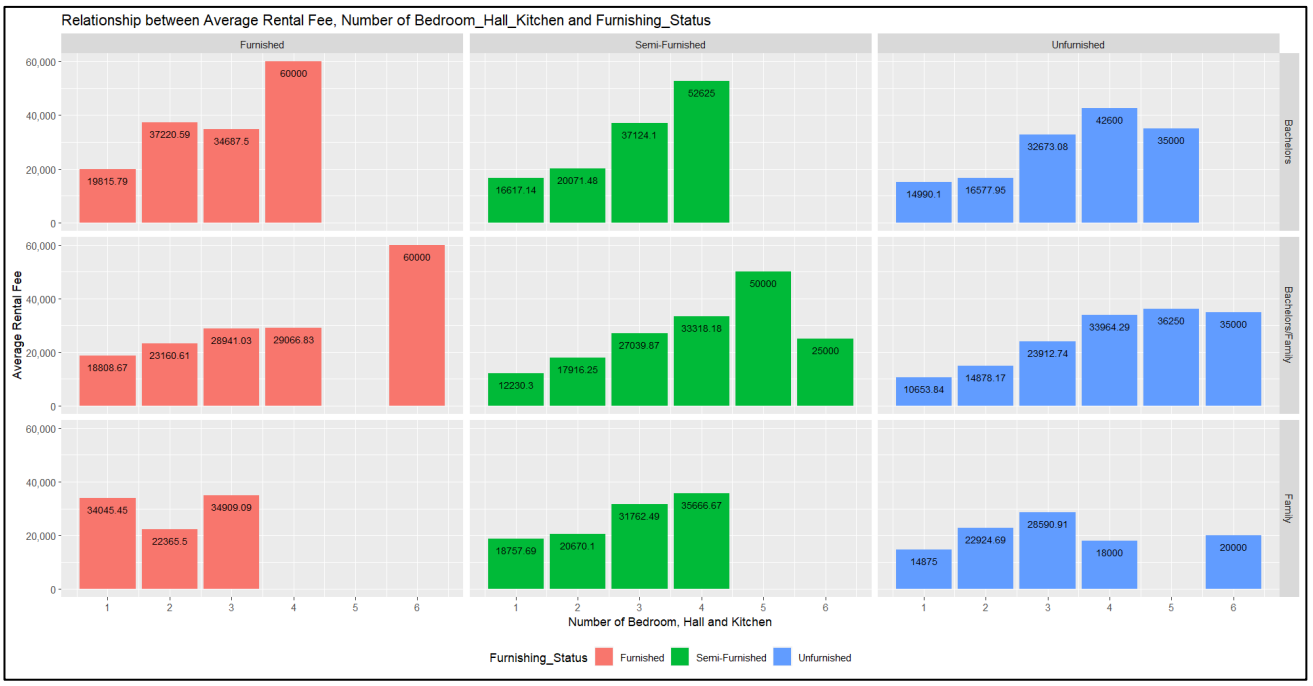



*Figure 6.7.14: Output – Bar Chart (Relationship between Rental_Fee, Bedroom_Hall_Kitchen and Furnishing_Status)*

**Explanation**

As shown in Figure 6.7.14, furnished houses with 4 bedrooms, hall, and kitchen provided to bachelors have the highest rental fee compared to semi-furnished and unfurnished houses. Furnished houses with 6 bedrooms, hall, and kitchen provided to bachelors/family have the highest rental fee compared to semi-furnished and unfurnished houses. Semi-furnished houses with 4 bedrooms, hall, and kitchen provided to family have the highest rental fee compared to furnished and unfurnished houses.

**Findings**

- More bachelors prefer unfurnished houses with 2 bedrooms, hall, and kitchen and rental fee of RM26,577,95.
- More bachelors/family prefer semi-furnished houses with 2 bedrooms, hall, and kitchen and rental fee of RM17,916.25.
- More family prefer semi-furnished houses with 2 bedrooms, hall, and kitchen and rental fee of RM20,670.10.

## Analysis 7-6: Find the Relationship between Rental Fee, Number_of_Bathroom and Area_Type

This analysis is conducted to investigate how tenants choose house based on rental fee, number of bathroom and the area type.

```
# Bar Chart
ggplot(rf_no_outliers,aes(x=Number_of_Bathroom,y=Rental_Fee,fill=Area_Type)) +
  geom_bar(stat="summary") +
  stat_summary(aes(label=round(after_stat(y),2)),size=3,fun="mean",geom="text",vjust=2) +
  scale_x_continuous(breaks=seq(1,7,1)) +
  scale_y_continuous(labels=scales::comma) +
  facet_grid(Tenant_Type~Area_Type) +
  theme(legend.position = "bottom") +
  labs(x="Number of Bathroom",y="Average Rental Fee",
       color="Area Type",
       title="Relationship between Average Rental Fee, Number_of_Bathroom and Area_Type")
```

*Figure 6.7.15: Source Code – Bar Chart to Show the Relationship between Rental_Fee, Number_of_Bathroom and Area_Type*

**Analysis Technique: Data Visualization and Manipulation**

Figure 6.7.15 depicts the source code used to show how tenants choose house based on rental fee, number of bathroom and area type. This relationship is calculated by find the mean using stat = "summary". **stat_summary()** can be used to add mean points to a bar chart. **Geom="text"** is entered so that to add the mean labels into the bar chart. **Scale_x_continuous()** is the position scales for continuous data of x-axis, The x-axis label is modified so that it produce 1 to 7 with 1 space between them in the graph and it is referred to breaks. **Scale_y_continuous()** is the position scales for continuous data of y-axis. The y-axis label is modified so that it do not show e notation in the graph, and it is referred to labels. **facet_grid()** generates a 2D grid of panels with the rows and columns given by Tenant_Type and Area_Type. **theme()** is used to control the non-data components such as the position of the legend to the bottom. **Labs()** is used to modify the name of the axes, name of legend and title.

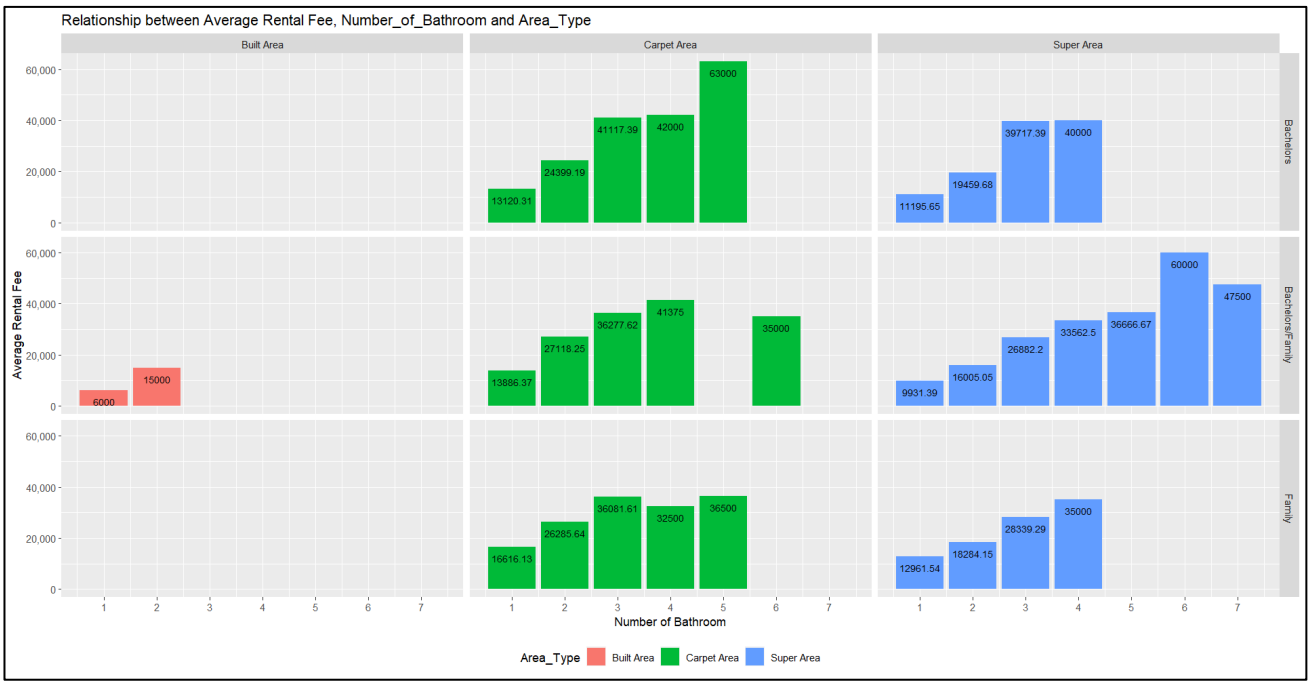



*Figure 6.7.16: Output – Bar Chart (Relationship between Rental_Fee, Number_of_Bathroom and Area_Type)*

**Explanation**

As shown in Figure 6.7.16, houses in carpet area with 5 bathrooms provided to bachelors have the highest rental fee compared to built area and super area. Houses in super area with 6 bathrooms provided to bachelors/family have the highest rental fee compared to built area and carpet area. Houses in carpet area with 3 bathrooms provided to family have the highest rental fee compared to built area and super area.

**Findings**

- More bachelors prefer to choose house that is located at carpet area with 2 bathrooms and average rental fee of RM24,399.19.
- More bachelors/family prefer to choose house that is located at super area with 2 bathrooms and average rental fee of RM16,005.05.
- More family prefer to choose house that is located at carpet area with 2 bathrooms and average rental fee of RM26,285.64.

## Analysis 7-7: Find the Relationship between Rental_Fee, Floor_Preference and Area_Type

```
# Lollipop Graph
ggplot(group_tt_fp_at,aes(x=avg_rental_fee,y=Floor_Preference,color=Area_Type)) +
  geom_point(size=2,colour="black") +
  geom_segment(aes(x=0,xend=avg_rental_fee,y=Floor_Preference,yend=Floor_Preference)) +
  labs(x="Average Rental Fee (RM)",y="Floor_Preference", color="Area Type",
       title="Relationship between Rental_Fee, Floor_Preference and Area_Type") +
  scale_x_continuous(breaks=seq(0,70000,5000),
                     labels = label_number(suffix = "k", scale = 1e-3)) +
  facet_grid(Tenant_Type~Area_Type)
```

*Figure 6.7.17: Source Code – Lollipop Graph to Show the Relationship between Rental_Fee, Floor_Preference and Area_Type*

**Analysis Technique: Data Visualization and Manipulation**

Figure 6.7.17 depicts the source code used to create the graph of tenant choose house based on Rental_Fee, Floor_Preference and Area_Type. The outliers in Rental_Fee have been removed. The number of tenants choose house based on this relationship is calculate using **group_by()** and **count()**. Then calculate the mean of house size, assign it to variable named "avg_rental_fee" and summarise it using **summarise()**.

The next source code is used to generate a lollipop graph to study how number of bathrooms varies the house size. **Geom_point()** is used to create point graph. **Geom_segment()** is used to draw a straight line between two points. **Labs()** is used to modify the names of the axes label and plot title. **scale_x_continuous()** is the position scales for continuous data of x-axis. The x-axis label is modified so that it produce 0 to 70000 with 5000 gaps between them in the graph and it is referred to breaks. It also can be used to represent thousands as k in the labels. **facet_grid()** generates a 2D grid of panels with the rows and columns given by Month and Tenant_Type.

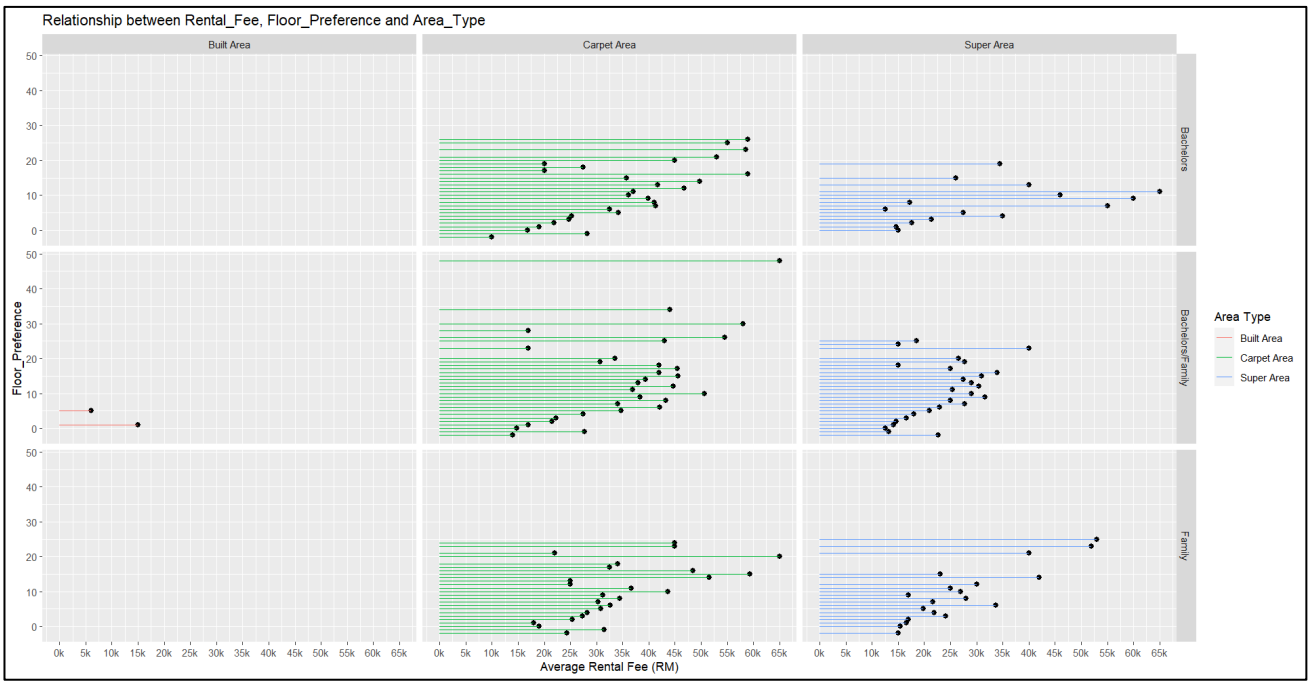



*Figure 6.7.18: Output – Lollipop Graph (Relationship between Rental_Fee, Floor_Preference and Area_Type)*

**Explanation**

From Figure 6.7.18, the house with the highest rental fee is located at the middle floor.

**Findings**

- More bachelors prefer to choose houses in carpet area which located at 1st floor with rental fee of around RM12,000.
- More bachelors/family prefer to choose houses in carpet area which located at 1st floor with rental fee of around RM10,000.
- More family prefer to choose houses in carpet area which located at 1st floor with rental fee of around RM15,000.

## Conclusion for Question 7

1. More tenants prefer semi-furnished houses with average rental fee of around RM12,000.
2. More tenants prefer houses located in super area with average rental fee of around RM15,000.
3. More tenants prefer houses located in Mumbai with average rental fee of around RM40,000.
4. More tenants prefer to contact owner to choose house with average rental fee of around RM15,000.
5. More tenants prefer to choose semi-furnished houses with 2 bedrooms, hall, and kitchen and rental fee of around RM15,000.
6. More tenants prefer to choose houses located in super area with 2 bathrooms and rental fee of around RM15,000.
7. More tenants prefer to choose houses located at 1st floor in super area and rental fee of around RM15,000.

# 7.0 Extra Features

## 7.1. sapply() and n_distinct()

**Code**

```
# 1. Count Number of Unique Values in Each Column
# From 3.1 Step 1
count_unique <- sapply(house_rental_data,function(x) n_distinct(x))
```

**Output**

```
>   count_unique
       Posted.On              BHK             Rent             Size
              81                6              243              615
           Floor        Area.Type    Area.Locality             City
             480                3             2235                6
Furnishing.Status  Tenant.Preferred         Bathroom  Point.of.Contact
               3                3                8                3
```

**Explanation**

To count the number of unique values in each column using **n_distinct(x)**

## 7.2. %in% operator

**Code**

```
# 2. Drop Area.Locality Column
# From 3.1 Step 3
house_rental_data <- house_rental_data[,!names(house_rental_data) %in% c("Area.Locality")]
```

**Output**

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

**Explanation**

It checks whether the column names of the data frame (i.e., names(house_rental_data)) are in a vector of variable names that wanted to remove (i.e., c("Area.Locality"). The "!" in front of the names functions is used to drop the variable that fits the logical condition.

## 7.3. as.Date()

**Code**

```
# 3. Format Date_Posted Column
# From 3.3
house_rental_data$Date_Posted <- as.Date(house_rental_data$Date_Posted,"%m/%d/%Y")
```

**Output**

| | Date_Posted | Bedroom_Hall_Kitchen | Rental_Fee | House_Size | Floor | Area_Type | City | Furnishing_Status | Tenant_Type | Number_of_Bathroom | Point_of_Contact |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2022-05-18 | 2 | 10000 | 1100 | Ground out of 2 | Super Area | Kolkata | Unfurnished | Bachelors/Family | 2 | Contact Owner |
| 2 | 2022-05-13 | 2 | 20000 | 800 | 1 out of 3 | Super Area | Kolkata | Semi-Furnished | Bachelors/Family | 1 | Contact Owner |
| 3 | 2022-05-16 | 2 | 17000 | 1000 | 1 out of 3 | Super Area | Kolkata | Semi-Furnished | Bachelors/Family | 1 | Contact Owner |
| 4 | 2022-07-04 | 2 | 10000 | 800 | 1 out of 2 | Super Area | Kolkata | Unfurnished | Bachelors/Family | 1 | Contact Owner |
| 5 | 2022-05-09 | 2 | 7500 | 850 | 1 out of 2 | Carpet Area | Kolkata | Unfurnished | Bachelors | 1 | Contact Owner |
| 6 | 2022-04-29 | 2 | 7000 | 600 | Ground out of 1 | Super Area | Kolkata | Unfurnished | Bachelors/Family | 2 | Contact Owner |

**Explanation**

To convert strings to dates using **as.Date()** so that it can be used to visualise the graph

## 7.4. month()

**Code**

```
# 4.Extract Month from Date_Posted
# From 3.4
house_rental_data$Month <- with(house_rental_data,month(ymd(Date_Posted)))
```

**Output**

| | Date_Posted | Month |
|---|---|---|
| 1 | 2022-05-18 | 5 |
| 2 | 2022-05-13 | 5 |
| 3 | 2022-05-16 | 5 |
| 4 | 2022-07-04 | 7 |
| 5 | 2022-05-09 | 5 |
| 6 | 2022-04-29 | 4 |

**Explanation**

**month()** is a function from the lubridate package to extract the month from a date. **Ymd()** is the format of the current dates. A new variable is created that contains month.

## 7.5. separate()

**Code**

```
# 5. Split Floor using separate()
# From 3.5 Step 2
house_rental_data <- house_rental_data %>% separate(Floor, c("Floor_Preference","Total_Floor_Numbers")," out of ")
```

**Output**

| | Date_Posted | Bedroom_Hall_Kitchen | Rental_Fee | House_Size | Floor_Preference | Total_Floor_Numbers | Area_Type | City | Furnishing_Status | Tenant_Type | Number_of_Bathroom | Point_of_Contact | Month |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2022-05-18 | 2 | 10000 | 1100 | Ground | 2 | Super Area | Kolkata | Unfurnished | Bachelors/Family | 2 | Contact Owner | 5 |
| 2 | 2022-05-13 | 2 | 20000 | 800 | 1 | 3 | Super Area | Kolkata | Semi-Furnished | Bachelors/Family | 1 | Contact Owner | 5 |
| 3 | 2022-05-16 | 2 | 17000 | 1000 | 1 | 3 | Super Area | Kolkata | Semi-Furnished | Bachelors/Family | 1 | Contact Owner | 5 |
| 4 | 2022-07-04 | 2 | 10000 | 800 | 1 | 2 | Super Area | Kolkata | Unfurnished | Bachelors/Family | 1 | Contact Owner | 7 |
| 5 | 2022-05-09 | 2 | 7500 | 850 | 1 | 2 | Carpet Area | Kolkata | Unfurnished | Bachelors | 1 | Contact Owner | 5 |
| 6 | 2022-04-29 | 2 | 7000 | 600 | Ground | 1 | Super Area | Kolkata | Unfurnished | Bachelors/Family | 2 | Contact Owner | 4 |

**Explanation**

The Floor column is separated into two new columns “Floor_Preference” and “Total_Floor_Numbers” with the delimiter “ out of ” using **separate()**.

## 7.6. colSums(is.na())

**Code**

```
# 6. Check for Missing Values
# From 3.7 Step 4
colSums(is.na(house_rental_data))
```

**Output**

```
      Date_Posted Bedroom_Hall_Kitchen       Rental_Fee       House_Size
                0                    0                0                0
 Floor_Preference Total_Floor_Numbers        Area_Type             City
                0                    4                0                0
Furnishing_Status          Tenant_Type Number_of_Bathroom Point_of_Contact
                0                    0                0                0
            Month
                0
```

**Explanation**

**colSums(is.na())** is used to calculate the sum of missing values in each column of the data frame.

## 7.7. User Defined Function (outliers and remove_outliers)

**Code**

```
# 7. User Defined Function (outliers)
# From 3.12
outliers <- function(x) {
  Q1 <- quantile(x, probs=.25)
  Q3 <- quantile(x, probs=.75)
  iqr = Q3-Q1

  upper_limit = Q3 + (iqr*1.5)
  lower_limit = Q1 - (iqr*1.5)

  x > upper_limit | x < lower_limit
}

remove_outliers <- function(house_rental_data, cols = names(house_rental_data)) {
  for (col in cols) {
    house_rental_data <- house_rental_data[!outliers(house_rental_data[[col]]),]
  }
  house_rental_data
}
```

**Output (Example of Usage)**

```
# Usage
rf_no_outliers <- remove_outliers(house_rental_data,c('Rental_Fee'))
```

**Explanation**

A user defined function of outliers is created. In order to find outliers, the first step is to find out the first (Q1) quartiles and third (Q3) quartiles using **quantile()** with **probs=.25** and **probs=.75**. Then, find the interquartile range (iqr) by subtracting Q3 with Q1. The next step is to find the upper limit and lower limit for outliers using Q3+(iqr*1.5) and Q1-(iqr*1.5). Then, set the condition if the value is greater than upper limit or lesser than lower limit.

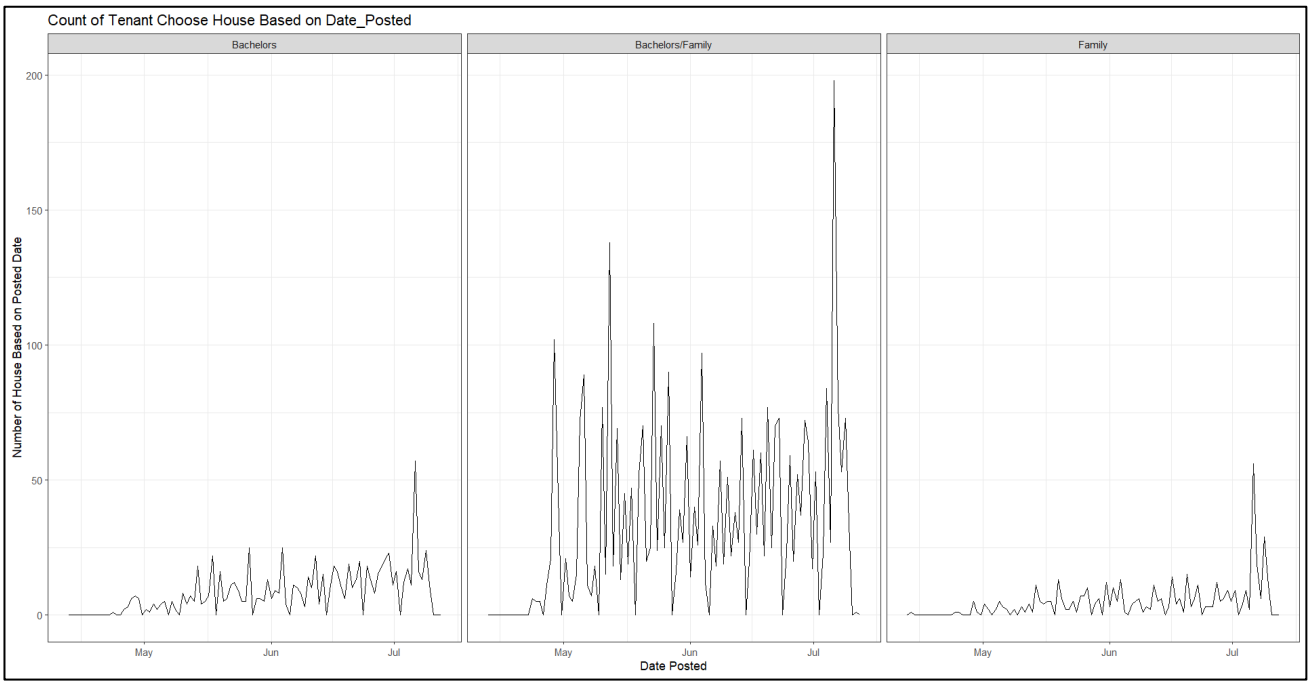
Then another user defined function of remove_outliers is created. The remove_outliers functions basically will go through the rows in the columns and see if the values match the condition. If it matches, then it will remove the outliers.

## 7.8. geom_freqpoly()

**Code**

```
# 8. geom_freqpoly()
# From Analysis 1.1
ggplot(house_rental_data,aes(x=Date_Posted)) +
  geom_freqpoly(bins=100) +
  xlab("Date Posted") +
  ylab("Number of House Based on Date Posted") +
  ggtitle("Count of Tenant CHoose House Based on Date_Posted") +
  theme_bw() +
  facet_wrap(~Tenant_Type)
```

**Output**




**Explanation**

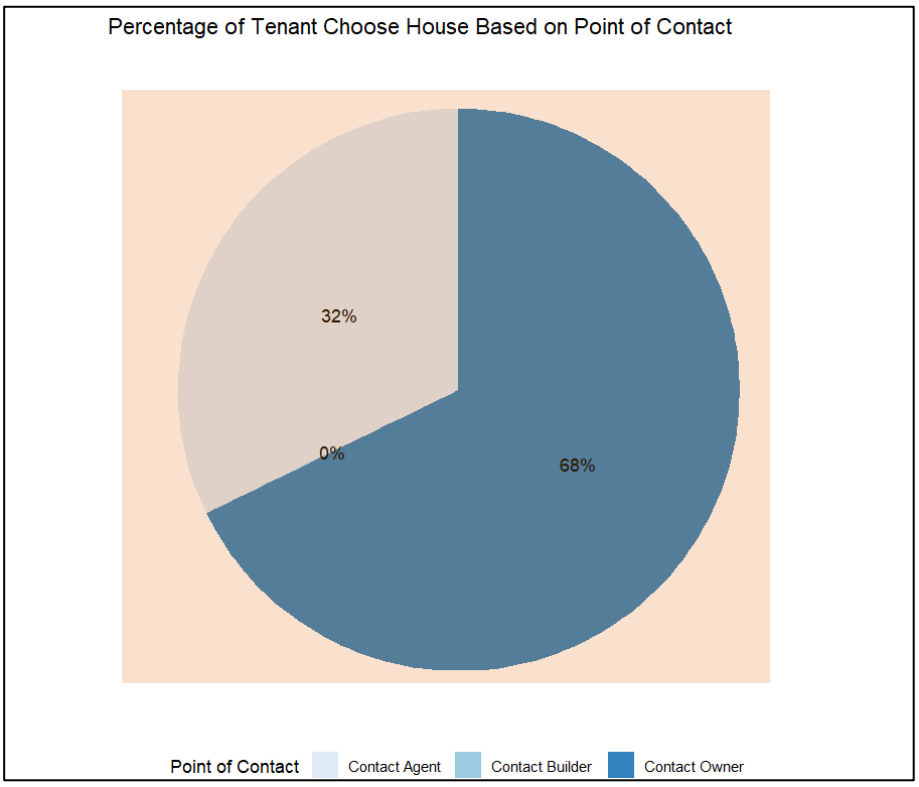
To visualize the distribution of a single continuous variable, a frequency polygon graph divides the x axis into bins and counts the number of observations in each bin.

## 7.9. coord_polar()

**Code**

```
# 9. coord_polar()
# From Analysis 1-2
ggplot(group_poc,aes(x="",y=perc,fill=Point_of_Contact)) +
  geom_bar(stat="identity") +
  guides(fill=guide_legend(title="Furnishing Status")) +
  geom_text(aes(label=labels),position=position_stack(vjust=0.5)) +
  coord_polar(theta="y") +
  theme_void() +
  theme(legend.position = "bottom") +
  scale_fill_brewer() +
  ggtitle("Percentage of Tenant Choose House Based on Point of Contact")
```

**Output**




**Explanation**

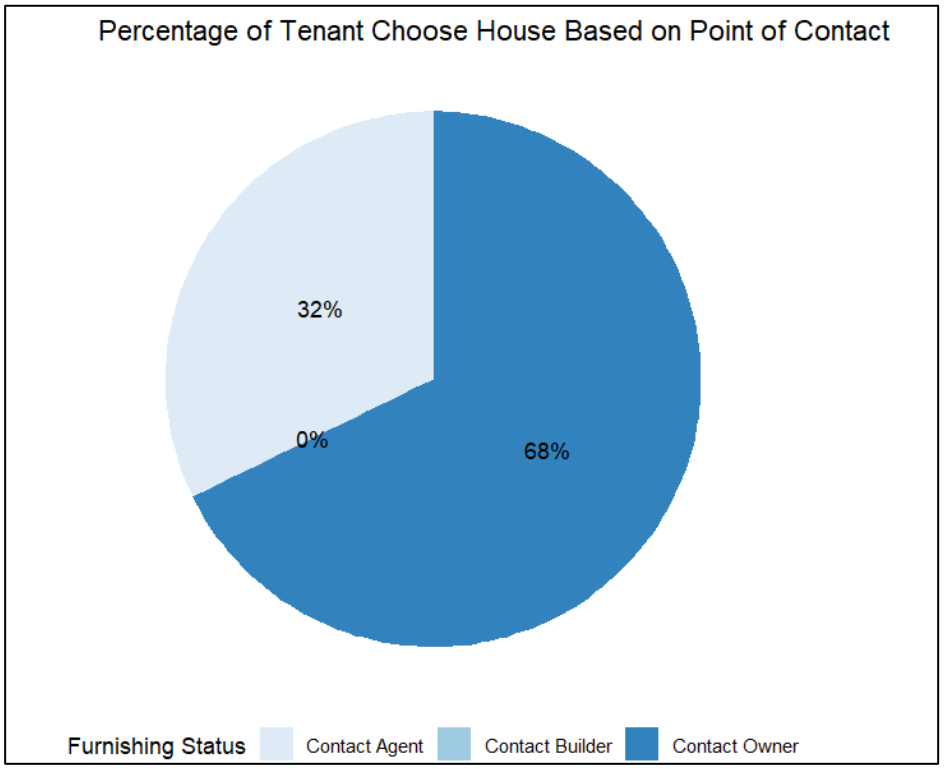
To convert a stacked bar chart to pie chart so that it can be visualized clearer

## 7.10. scale_fill_brewer()

**Code**

```
# 10. scale_fill_brewer()
# From Analysis 1-2
ggplot(group_poc,aes(x="",y=perc,fill=Point_of_Contact)) +
  geom_bar(stat="identity") +
  guides(fill=guide_legend(title="Furnishing Status")) +
  geom_text(aes(label=labels),position=position_stack(vjust=0.5)) +
  coord_polar(theta="y") +
  theme_void() +
  theme(legend.position = "bottom") +
  scale_fill_brewer() +
  ggtitle("Percentage of Tenant Choose House Based on Point of Contact")
```

**Output**




**Explanation**

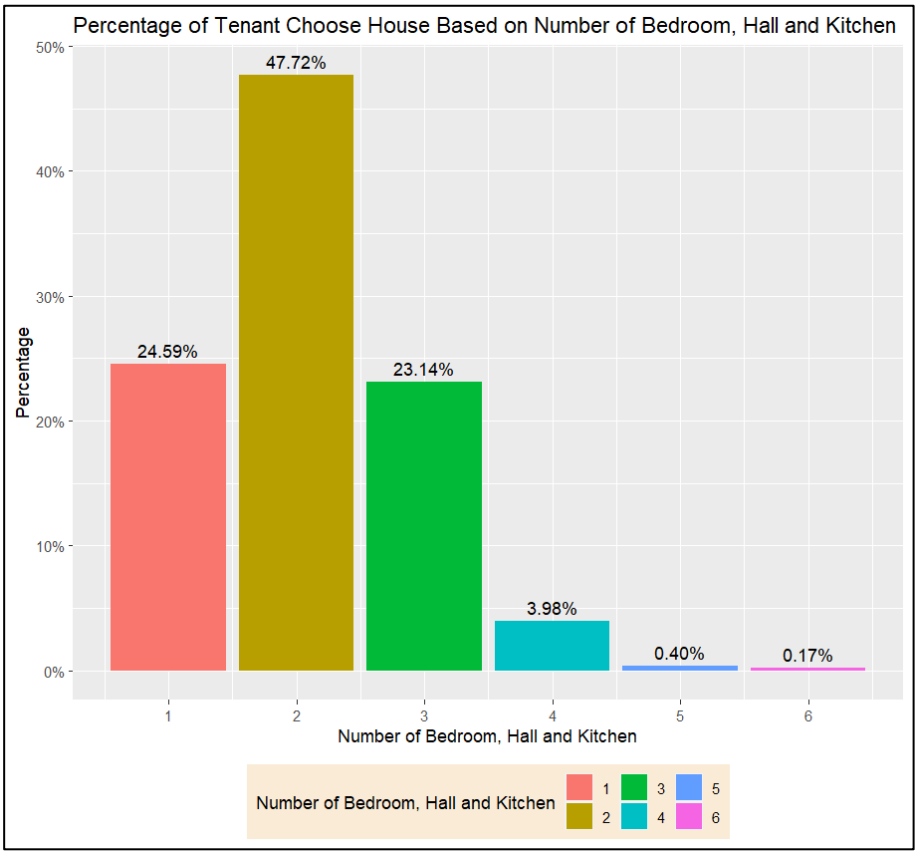
To use the colour schemes from ColorBrewer so that the category can be differentiate with the colour

## 7.11. after_stat(prop)

**Code**

```
# 11. after_stat(prop)
# From Analysis 1-3
ggplot(house_rental_data,aes(x=Bedroom_Hall_Kitchen)) +
  geom_bar(aes(y=after_stat(prop),fill=factor(after_stat(x))),stat="count") +
  geom_text(aes(label=scales::percent(after_stat(prop)),
                y=after_stat(prop)),stat="count",vjust=-0.5) +
  labs(x="Number of Bedroom, Hall and Kitchen",y="Percentage",
       fill="Number of Bedroom, Hall and Kitchen") +
  scale_x_continuous(breaks=seq(1,6,1)) +
  scale_y_continuous(labels=scales::percent) +
  theme(legend.position="bottom",
        legend.background = element_rect(fill="antiquewhite")) +
  ggtitle("Percentage of Tenant Choose House Based on Number of Bedroom, Hall and Kitchen")
```

**Output**




**Explanation**

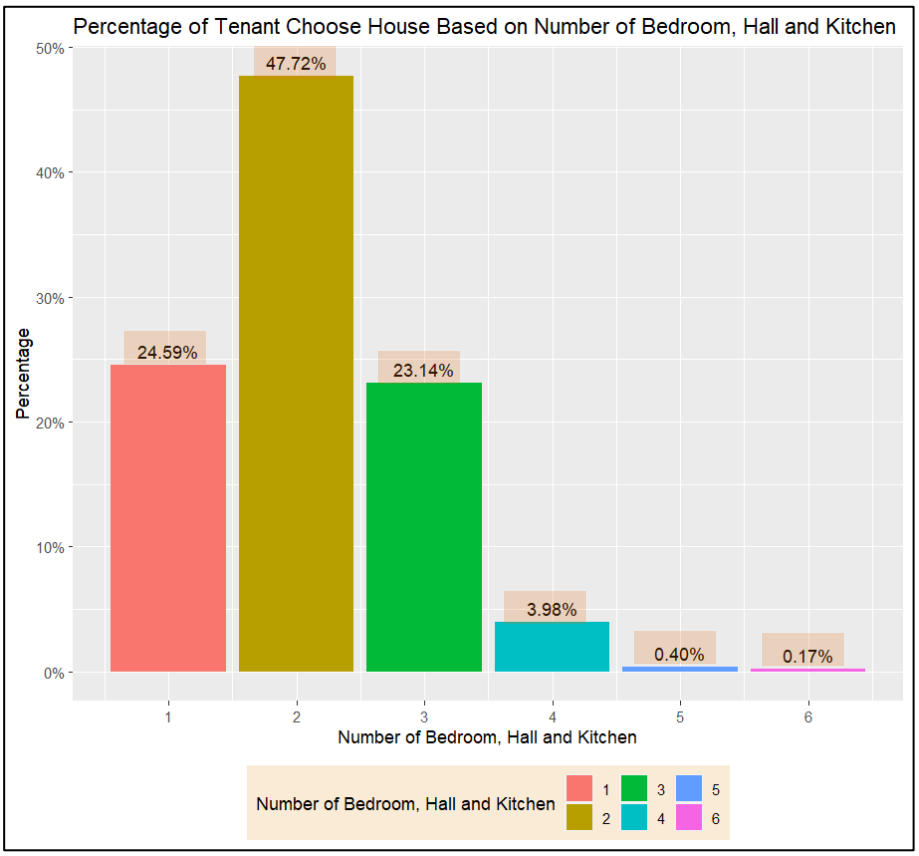
To calculate the proportion of Bedroom_Hall_Kitchen so that it can be converted to percentage later

## 7.12. labels = scales :: percent

**Code**

```
# 12. labels=scales::percent
# From Analysis 1-3
ggplot(house_rental_data,aes(x=Bedroom_Hall_Kitchen)) +
  geom_bar(aes(y=after_stat(prop),fill=factor(after_stat(x))),stat="count") +
  geom_text(aes(label=scales::percent(after_stat(prop)),
                y=after_stat(prop)),stat="count",vjust=-0.5) +
  labs(x="Number of Bedroom, Hall and Kitchen",y="Percentage",
       fill="Number of Bedroom, Hall and Kitchen") +
  scale_x_continuous(breaks=seq(1,6,1)) +
  scale_y_continuous(labels=scales::percent) +
  theme(legend.position="bottom",
        legend.background = element_rect(fill="antiquewhite")) +
  ggtitle("Percentage of Tenant Choose House Based on Number of Bedroom, Hall and Kitchen")
```

**Output**




**Explanation**

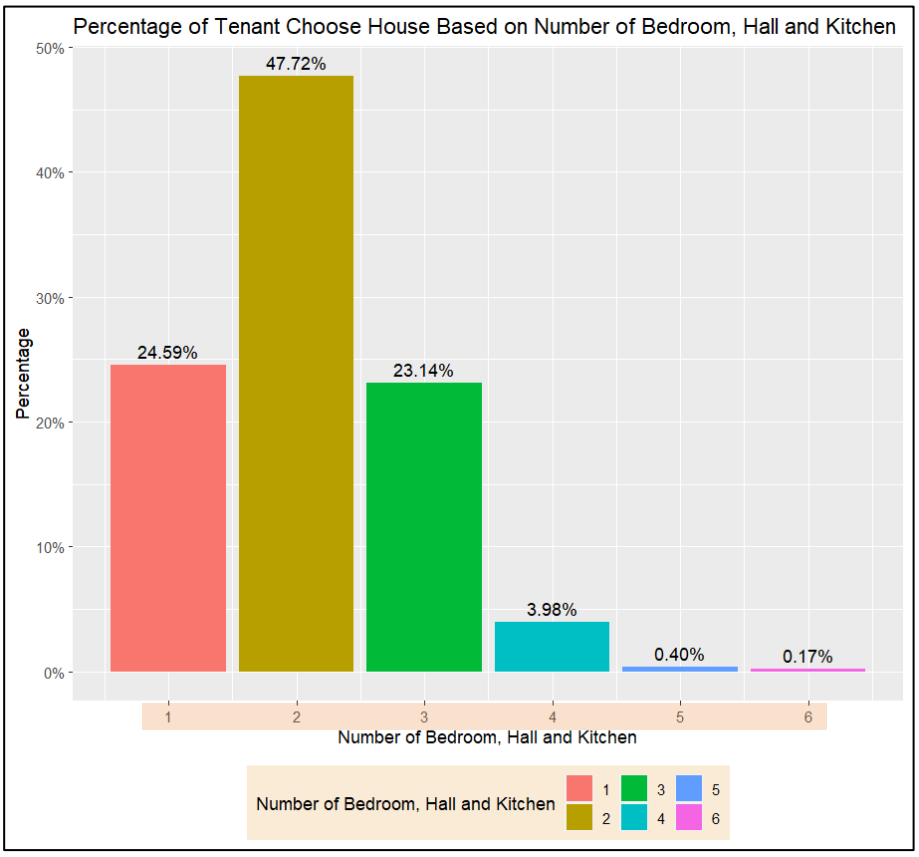
To label the count in percentages on each bar so that accurate analysis can be done

## 7.13. scale_x_continuous(breaks)

**Code**

```
# 13. scale_x_continuous(breaks)
# From Analysis 1-3
ggplot(house_rental_data,aes(x=Bedroom_Hall_Kitchen)) +
  geom_bar(aes(y=after_stat(prop),fill=factor(after_stat(x))),stat="count") +
  geom_text(aes(label=scales::percent(after_stat(prop)),
                y=after_stat(prop)),stat="count",vjust=-0.5) +
  labs(x="Number of Bedroom, Hall and Kitchen",y="Percentage",
       fill="Number of Bedroom, Hall and Kitchen") +
  scale_x_continuous(breaks=seq(1,6,1)) +
  scale_y_continuous(labels=scales::percent) +
  theme(legend.position="bottom",
        legend.background = element_rect(fill="antiquewhite")) +
  ggtitle("Percentage of Tenant Choose House Based on Number of Bedroom, Hall and Kitchen")
```

**Output**




**Explanation**

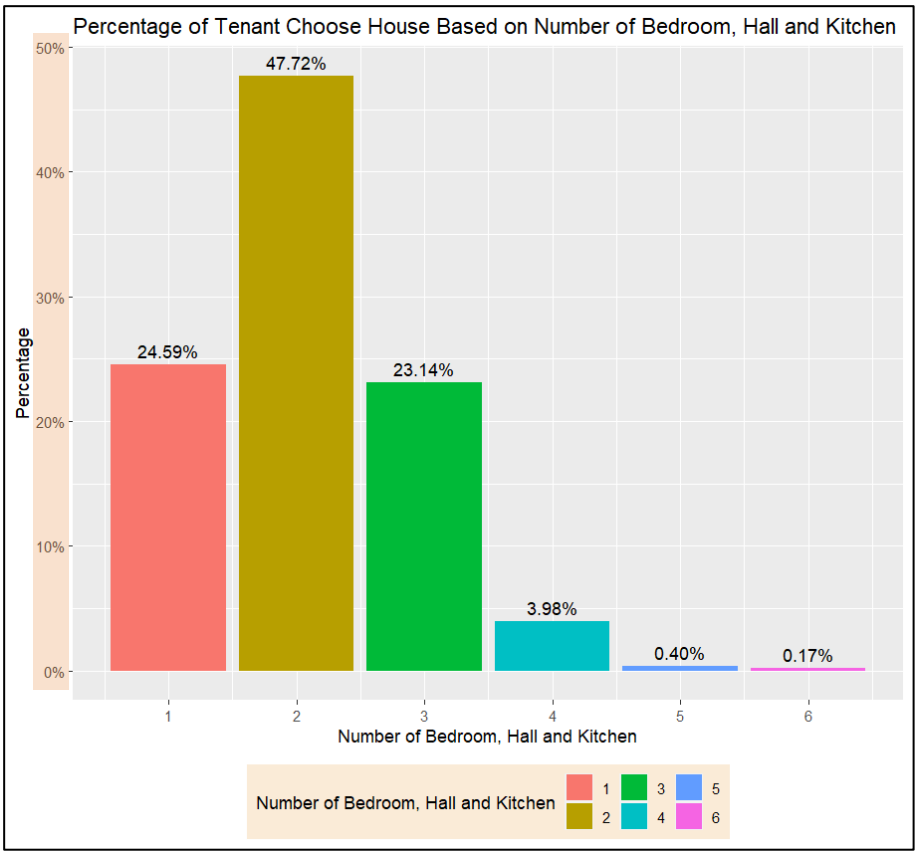
To show the label of x-axis by setting the breaks with **seq()**

## 7.14. scale_y_continuous(labels)

**Code**

```
# 14. scale_y_continuous(labels)
# From Analysis 1-3
ggplot(house_rental_data,aes(x=Bedroom_Hall_Kitchen)) +
  geom_bar(aes(y=after_stat(prop),fill=factor(after_stat(x))),stat="count") +
  geom_text(aes(label=scales::percent(after_stat(prop)),
                y=after_stat(prop)),stat="count",vjust=-0.5) +
  labs(x="Number of Bedroom, Hall and Kitchen",y="Percentage",
       fill="Number of Bedroom, Hall and Kitchen") +
  scale_x_continuous(breaks=seq(1,6,1)) +
  scale_y_continuous(labels=scales::percent) +
  theme(legend.position="bottom",
        legend.background = element_rect(fill="antiquewhite")) +
  ggtitle("Percentage of Tenant Choose House Based on Number of Bedroom, Hall and Kitchen")
```

**Output**




**Explanation**

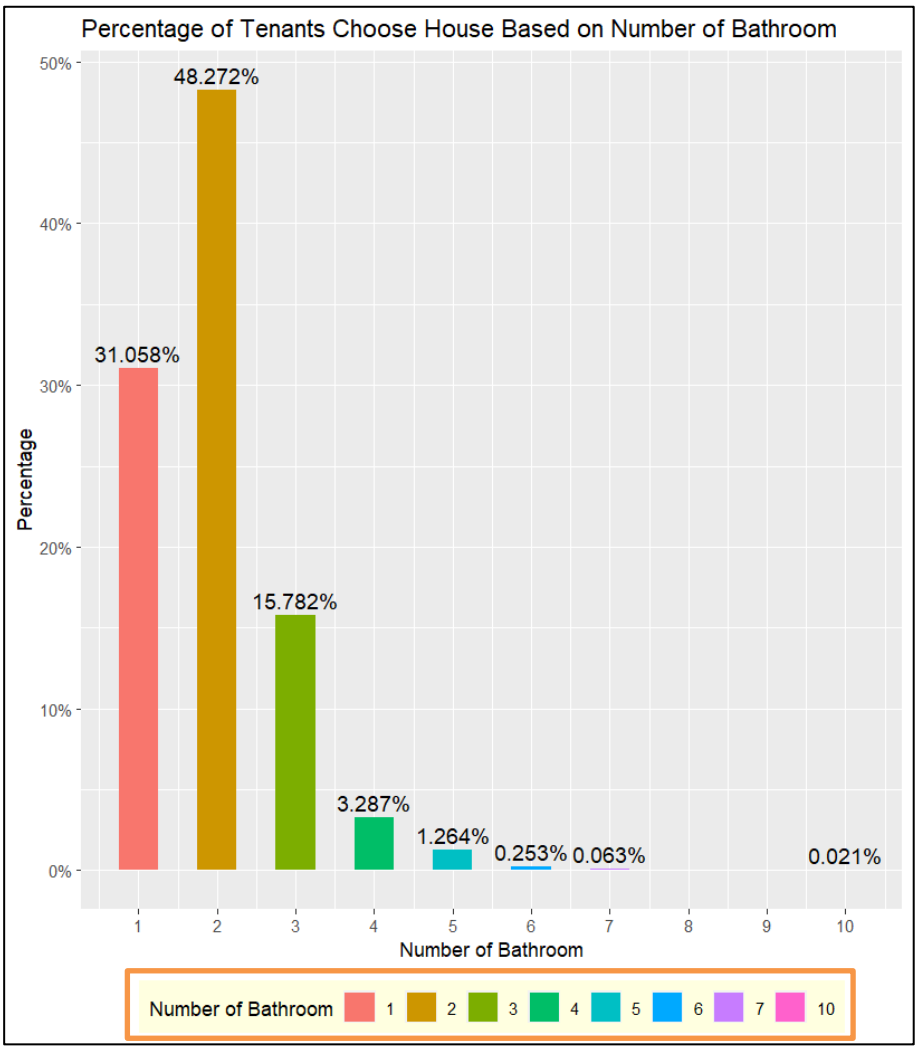
To show the labels in percentage in the y-axis

## 7.15. theme()

**Code**

```
# 15. theme()
# From Analysis 1-4
ggplot(house_rental_data,aes(x=Number_of_Bathroom)) +
  geom_bar(aes(y=after_stat(prop),fill=factor(after_stat(x))),stat="count",width=0.5) +
  geom_text(aes(label=scales::percent(after_stat(prop)),
                y=after_stat(prop)),stat="count",vjust=-0.4,size=4) +
  labs(x="Number of Bathroom",y="Percentage",fill="Number of Bathroom") +
  scale_x_continuous(breaks=seq(1,10,1)) +
  scale_y_continuous(labels=scales::percent) +
  theme(legend.position="bottom",
        legend.background = element_rect(fill="lightyellow")) +
  ggtitle("Percentage of Tenants Choose House Based on Number of Bathroom")
```

**Output**




**Explanation**

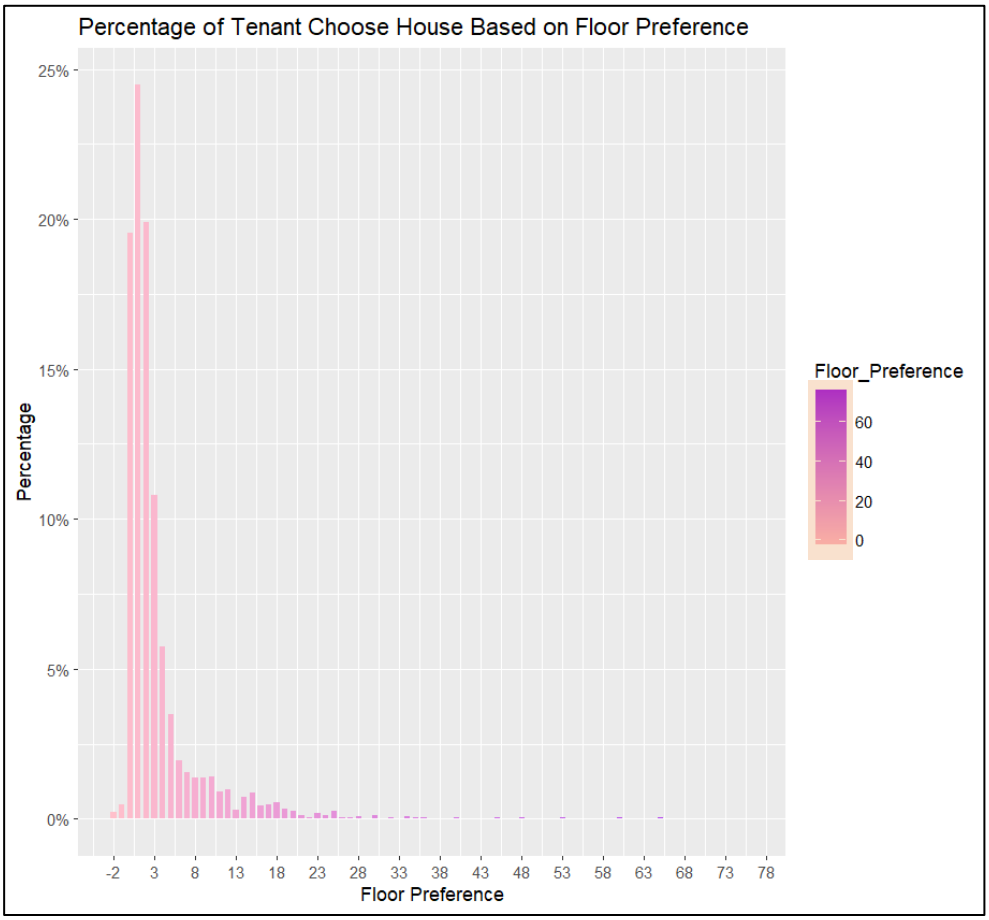
By default, the legend appears at the right side. **legend.position=”bottom”** will bring the legends to the bottom **legend.background** is used to set the background of the legend.

## 7.16. scale_fill_gradient()

**Code**

```
# 16. scale_fill_gradient()
# From Analysis 1-5
# Calculate Percentage Grouped by Floor_Preference
group_fp <- house_rental_data %>%
  group_by(Floor_Preference) %>%
  count() %>% ungroup() %>% mutate(perc=n/sum(n)) %>%
  arrange(perc) %>% mutate(labels=scales::percent(perc))
# Bar Chart
ggplot(group_fp,aes(x=Floor_Preference,y=perc,fill=Floor_Preference,label=labels)) +
  geom_bar(stat="identity",width=0.7) +
  scale_x_continuous(breaks=seq(-2,80,5)) +
  scale_y_continuous(labels=scales::percent) +
  scale_fill_gradient(low="pink",high="purple") +
  labs(x="Floor Preference",y="Percentage",
       title="Percentage of Tenant Choose House Based on Floor Preference")
```

**Output**




**Explanation**

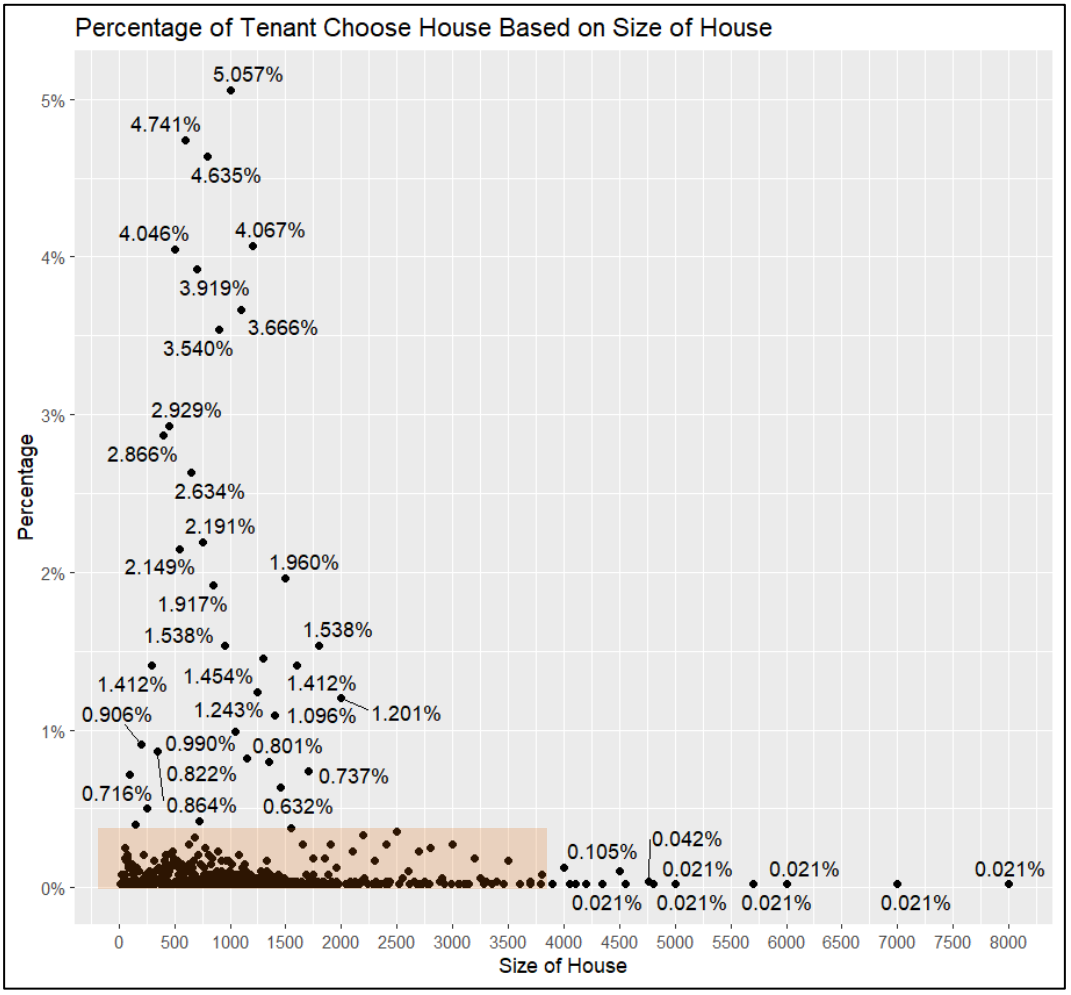
To change the colour of the scale so that it can be visualised

## 7.17. geom_text_repel(max.overlaps)

**Code**

```
# 17. geom_text_repel(max.overlaps)
# From Analysis 1-6
# Calculate Percentage Grouped by House_Size
group_hs <- house_rental_data %>% group_by(House_Size) %>% count() %>% ungroup() %>%
  mutate(perc=n/sum(n)) %>%
  arrange(perc) %>%
  mutate(labels=scales::percent(perc))
# Scatter Plot
ggplot(group_hs,aes(x=House_Size,y=perc,label=labels)) +
  geom_point(aes(x=House_Size,y=perc)) +
  scale_x_continuous(breaks=seq(0,8000,500)) +
  scale_y_continuous(labels=scales::percent) +
  geom_text_repel(max.overlaps = 20) +
  ggtitle("Percentage of Tenant Choose House Based on Size of House") +
  labs(x="Size of House",y="Percentage")
```

**Output**




**Explanation**

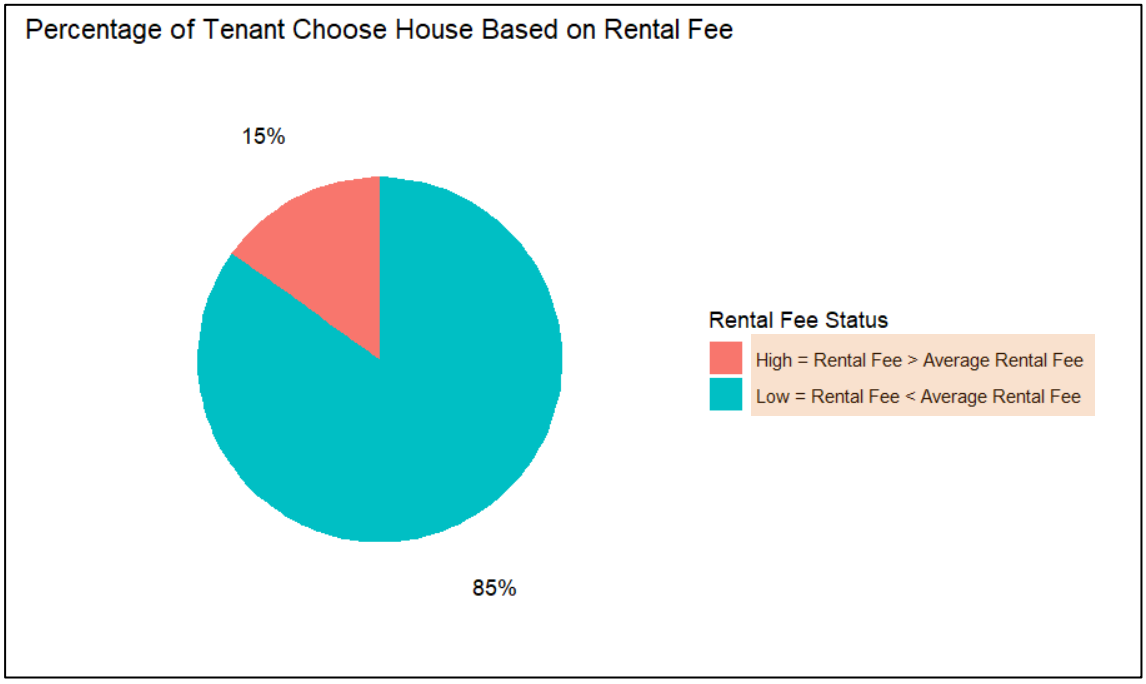
Overlapping labels will be repelled so that the graph looks neat

## 7.18. scale_fill_discrete(labels)

**Code**

```
# 18. scale_fill_discrete(labels)
# From Analysis 1-7
# Calculate Percentage Grouped by Rental_Fee_Status
group_rfs <- rf_no_outliers %>% group_by(Rental_Fee_Status) %>%
  count() %>% ungroup() %>% mutate(perc=n/sum(n)) %>%
  arrange(perc) %>% mutate(labels=scales::percent(perc))
# Pie Chart
ggplot(group_rfs,aes(x="",y=perc,fill=as.factor(Rental_Fee_Status))) +
  geom_bar(stat="identity") +
  geom_text(aes(x=1.8,label=labels),position=position_stack(vjust=0.5)) +
  coord_polar(theta="y") +
  theme_void() +
  ggtitle("Percentage of Tenant Choose House Based on Rental Fee") +
  scale_fill_discrete(labels=c("High = Rental Fee > Average Rental Fee",
                               "Low = Rental Fee < Average Rental Fee")) +
  guides(fill=guide_legend(title="Rental Fee Status")) +
  theme(plot.title=element_text(hjust=0.5))
```

**Output**




**Explanation**

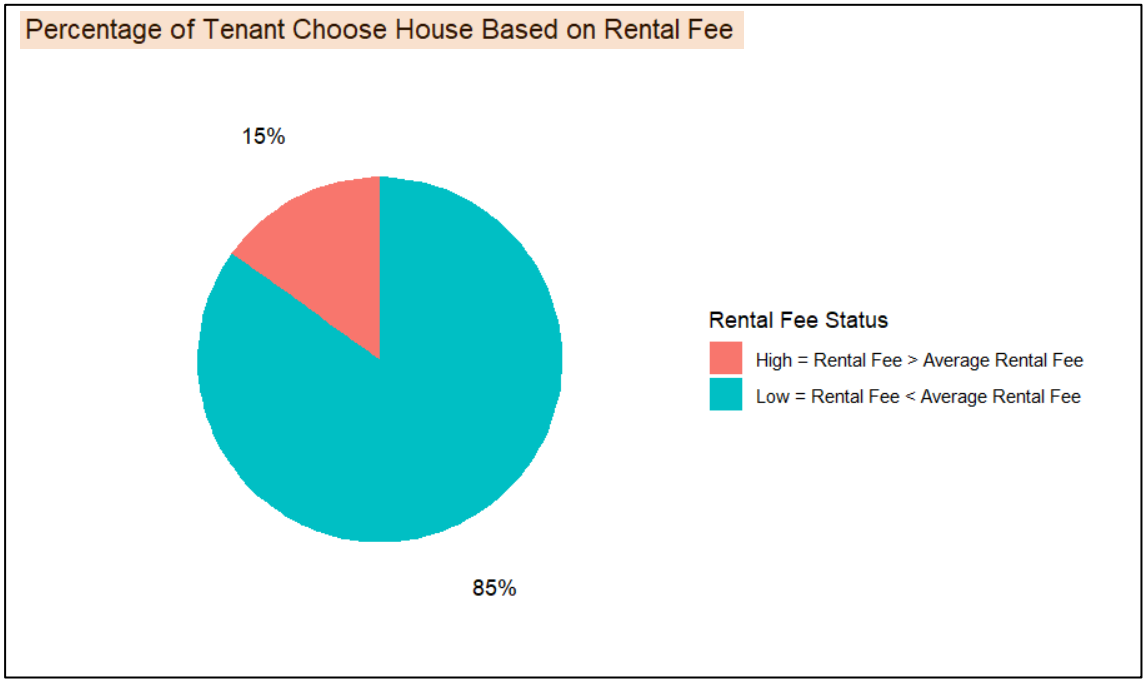
To change the labels of legends so that to provide better understanding of high rental fee status and low rental fee status

## 7.19. theme(plot.title)

**Code**

```
# 19. theme(plot.title)
# From Analysis 1-7
# Calculate Percentage Grouped by Rental_Fee_Status
group_rfs <- rf_no_outliers %>% group_by(Rental_Fee_Status) %>%
  count() %>% ungroup() %>% mutate(perc=n/sum(n)) %>%
  arrange(perc) %>% mutate(labels=scales::percent(perc))
# Pie Chart
ggplot(group_rfs,aes(x="",y=perc,fill=as.factor(Rental_Fee_Status))) +
  geom_bar(stat="identity") +
  geom_text(aes(x=1.8,label=labels),position=position_stack(vjust=0.5)) +
  coord_polar(theta="y") +
  theme_void() +
  ggtitle("Percentage of Tenant Choose House Based on Rental Fee") +
  scale_fill_discrete(labels=c("High = Rental Fee > Average Rental Fee",
                               "Low = Rental Fee < Average Rental Fee")) +
  guides(fill=guide_legend(title="Rental Fee Status")) +
  theme(plot.title=element_text(hjust=0.5))
```

**Output**




**Explanation**

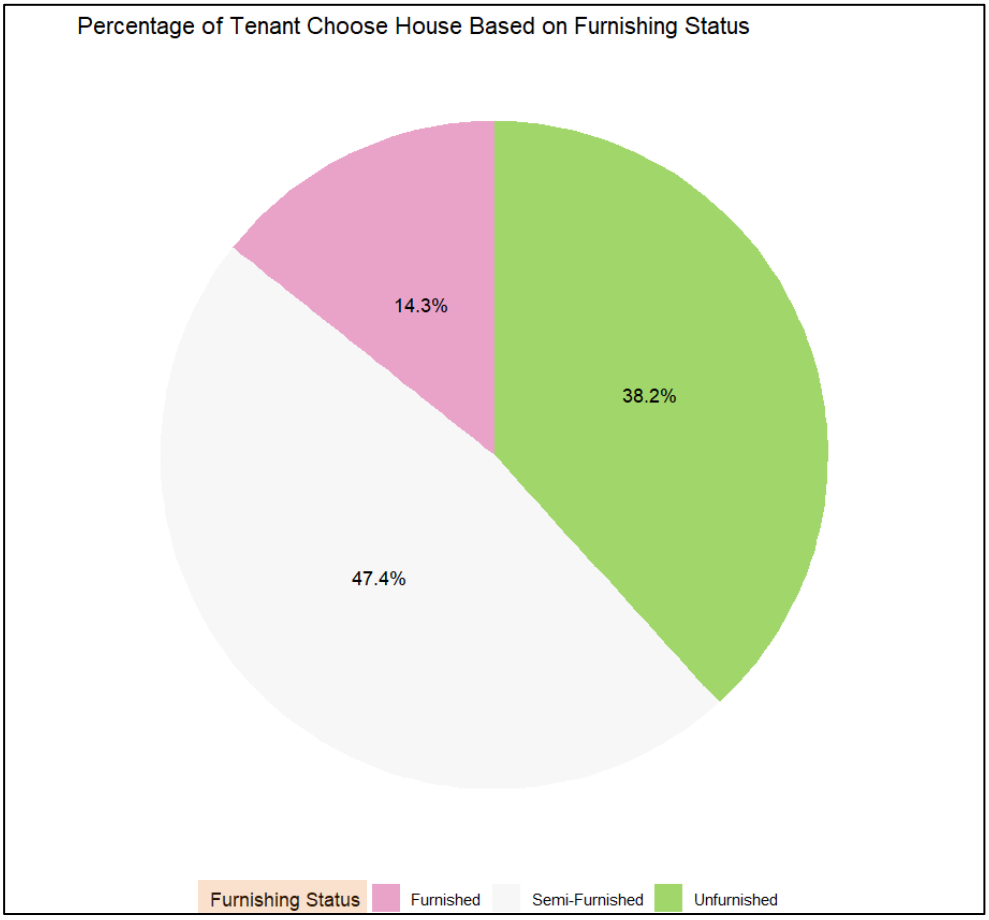
To align the title to the center of the pie chart

## 7.20. guides()

**Code**

```
# 20. guides()
# From Analysis 1-8
# Calculate Percentage Grouped by Furnishing_Status
group_fs <- house_rental_data %>% group_by(Furnishing_Status) %>%
  count() %>% ungroup() %>% mutate(perc=n/sum(n)) %>%
  arrange(perc) %>%
  mutate(labels=scales::percent(perc))
# Pie Chart
ggplot(group_fs,aes(x="",y=perc,fill=Furnishing_Status)) +
  geom_bar(stat="identity") +
  guides(fill=guide_legend(title="Furnishing Status")) +
  geom_text(aes(label=labels),position=position_stack(vjust=0.5)) +
  coord_polar(theta="y") +
  theme_void() +
  theme(legend.position = "bottom") +
  scale_fill_brewer(palette = "PiYG") +
  ggtitle("Percentage of Tenant Choose House Based on Furnishing Status")
```

**Output**




**Explanation**

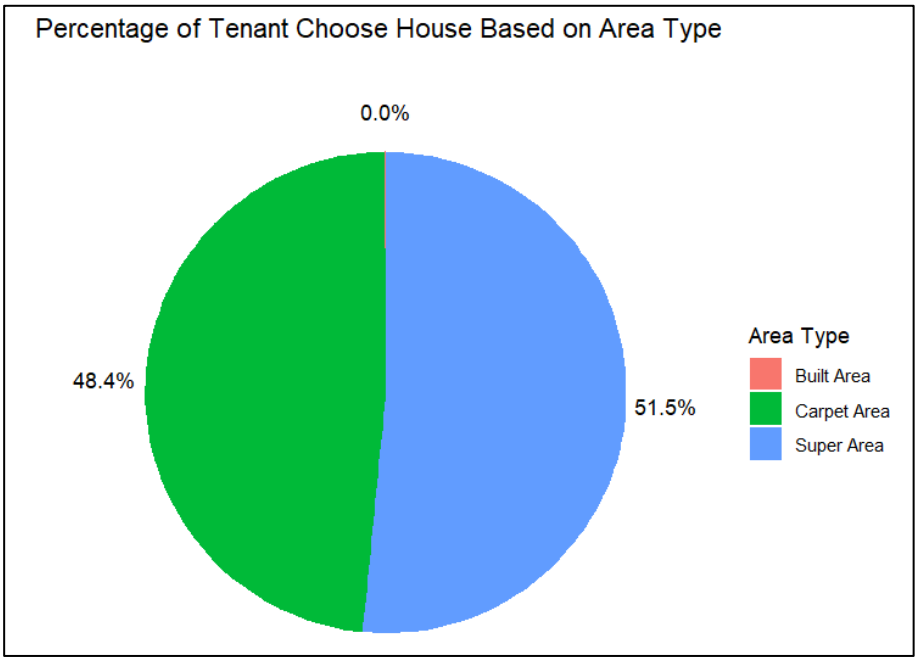
To modify the name of the legend so that the legend has a meaningful name

## 7.21. theme_void()

**Code**

```
# 11. theme_void()
# From Analysis 1-9
# Calculate Percentage Grouped by Area_Type
group_at <- house_rental_data %>% group_by(Area_Type) %>%
  count() %>% ungroup() %>% mutate(perc=n/sum(n)) %>%
  arrange(perc) %>% mutate(labels=scales::percent(perc))
# Pie Chart
ggplot(group_at,aes(x="",y=perc,fill=Area_Type)) +
  geom_bar(stat="identity") +
  guides(fill=guide_legend(title="Area Type")) +
  geom_text(aes(x=1.6,label=labels),position=position_stack(vjust=0.5)) +
  coord_polar(theta="y") +
  theme_void() +
  ggtitle("Percentage of Tenant Choose House Based on Area Type")
```

**Output**




**Explanation**

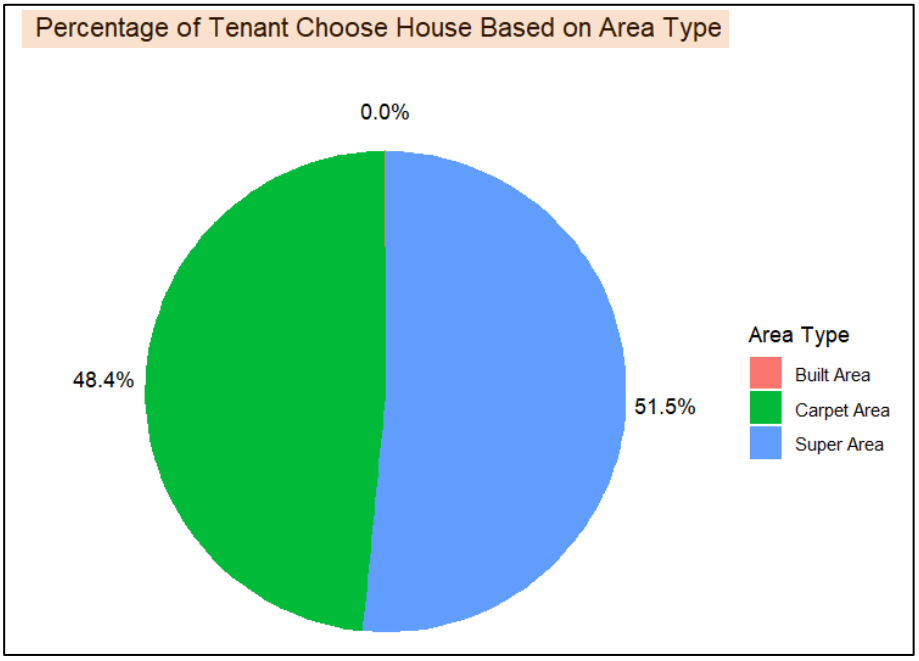
To show only the plot components so that it does not look mess

## 7.22. ggtitle()

**Code**

```
# 14. ggtitle()
# From Analysis 1-9
# Calculate Percentage Grouped by Area_Type
group_at <- house_rental_data %>% group_by(Area_Type) %>%
  count() %>% ungroup() %>% mutate(perc=n/sum(n)) %>%
  arrange(perc) %>% mutate(labels=scales::percent(perc))
# Pie Chart
ggplot(group_at,aes(x="",y=perc,fill=Area_Type)) +
  geom_bar(stat="identity") +
  guides(fill=guide_legend(title="Area Type")) +
  geom_text(aes(x=1.6,label=labels),position=position_stack(vjust=0.5)) +
  coord_polar(theta="y") +
  theme_void() +
  ggtitle("Percentage of Tenant Choose House Based on Area Type")
```

**Output**




**Explanation**

To give a title to the plot so that it is easily to know what is the graph about

## 7.23. month.abb[]

**Code**

```
# 23. Convert Numeric to month names
# From Question 2
house_rental_data$Month <- month.abb[house_rental_data$Month]
```

**Output**

| | Date_Posted | Month |
|---|---|---|
| 1 | 2022-05-18 | 5 |
| 2 | 2022-05-13 | 5 |
| 3 | 2022-05-16 | 5 |
| 4 | 2022-07-04 | 7 |
| 5 | 2022-05-09 | 5 |
| 6 | 2022-04-29 | 4 |

| | Date_Posted | Month |
|---|---|---|
| 1 | 2022-05-18 | May |
| 2 | 2022-05-13 | May |
| 3 | 2022-05-16 | May |
| 4 | 2022-07-04 | Jul |
| 5 | 2022-05-09 | May |
| 6 | 2022-04-29 | Apr |

**Explanation**

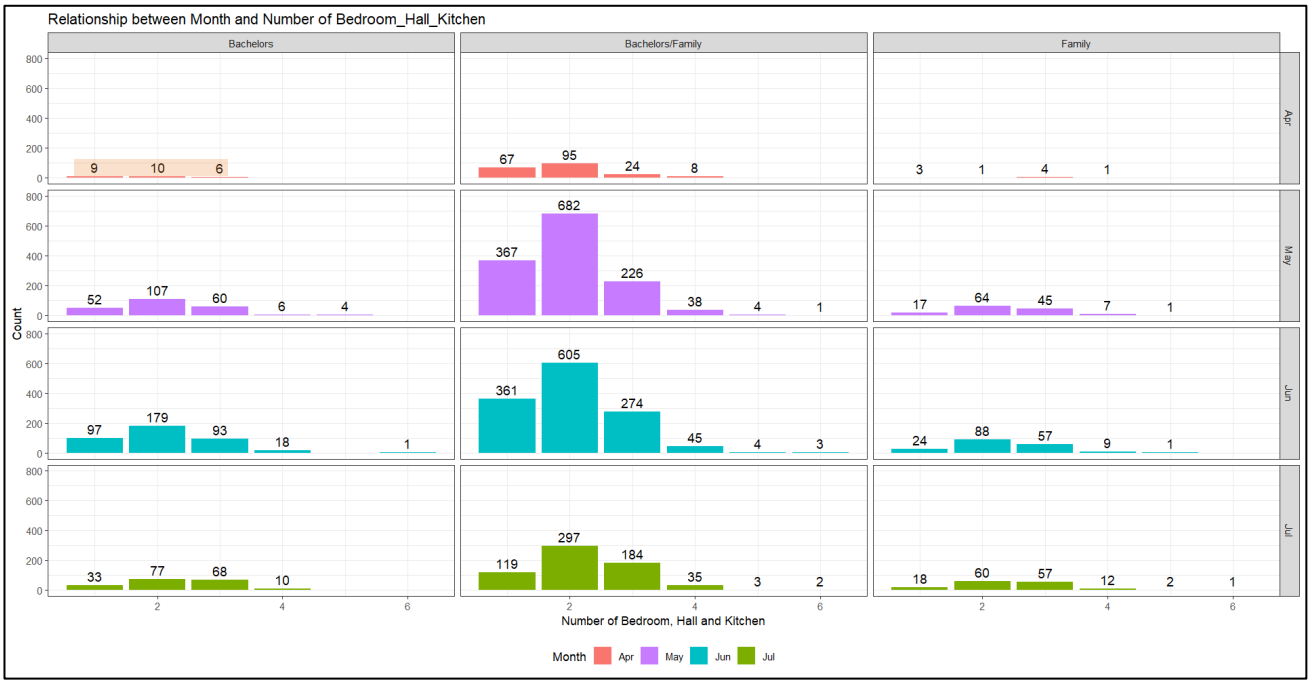
Convert the numeric month to month names for easier visualisation in the graph

## 7.24. stat = “count”

**Code**

```
# 24. stat = "count"
# From Analysis 2-1
ggplot(house_rental_data,aes(x=Bedroom_Hall_Kitchen,fill=Month)) +
  geom_bar(aes(y=after_stat(count)),stat="count") +
  labs(x="Number of Bedroom, Hall and Kitchen",y="Count",
       title="Relationship between Date_Posted and Number of Bedroom_Hall_Kitchen") +
  geom_text(aes(label=after_stat(count),
                y=after_stat(count)),stat="count",vjust=-0.4,size=4) +
  scale_fill_discrete(breaks=c('Apr','May','Jun','Jul')) +
  ylim(0,800) +
  theme_bw() +
  theme(legend.position = "bottom") +
  facet_grid(factor(Month,levels=c('Apr','May','Jun','Jul'))~Tenant_Type)
```

**Output**




**Explanation**

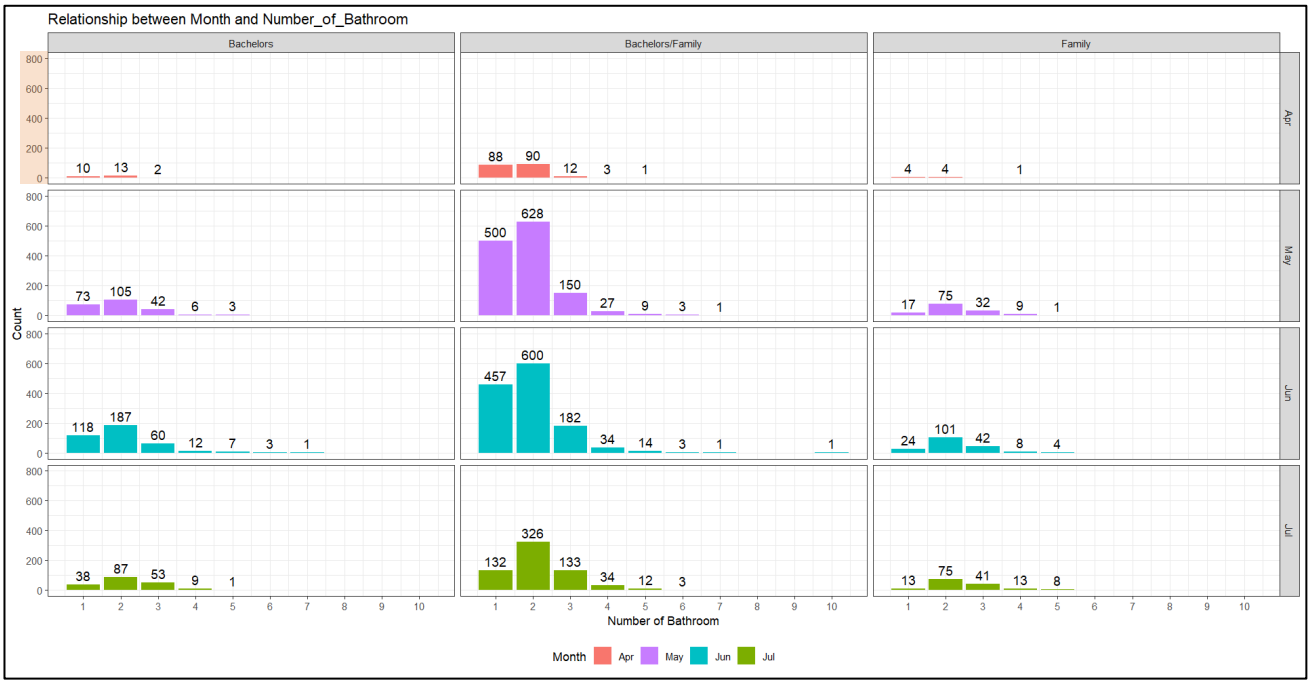
To label the counts the observation for each bar

## 7.25. ylim()

**Code**

```
# 25. ylim()
# From Analysis 2-2
ggplot(house_rental_data,aes(x=Number_of_Bathroom,fill=Month)) +
  geom_bar(aes(y=after_stat(count)),stat="count") +
  labs(x="Number of Bathroom",y="Count",
       title="Relationship between Month and Number_of_Bathroom") +
  geom_text(aes(label=after_stat(count),
                y=after_stat(count)),stat="count",vjust=-0.4,size=4) +
  scale_fill_discrete(breaks=c('Apr','May','Jun','Jul')) +
  ylim(0,800) +
  theme_bw() +
  theme(legend.position = "bottom") +
  facet_grid(factor(Month,levels=c('Apr','May','Jun','Jul'))~Tenant_Type)
```

**Output**




**Explanation**

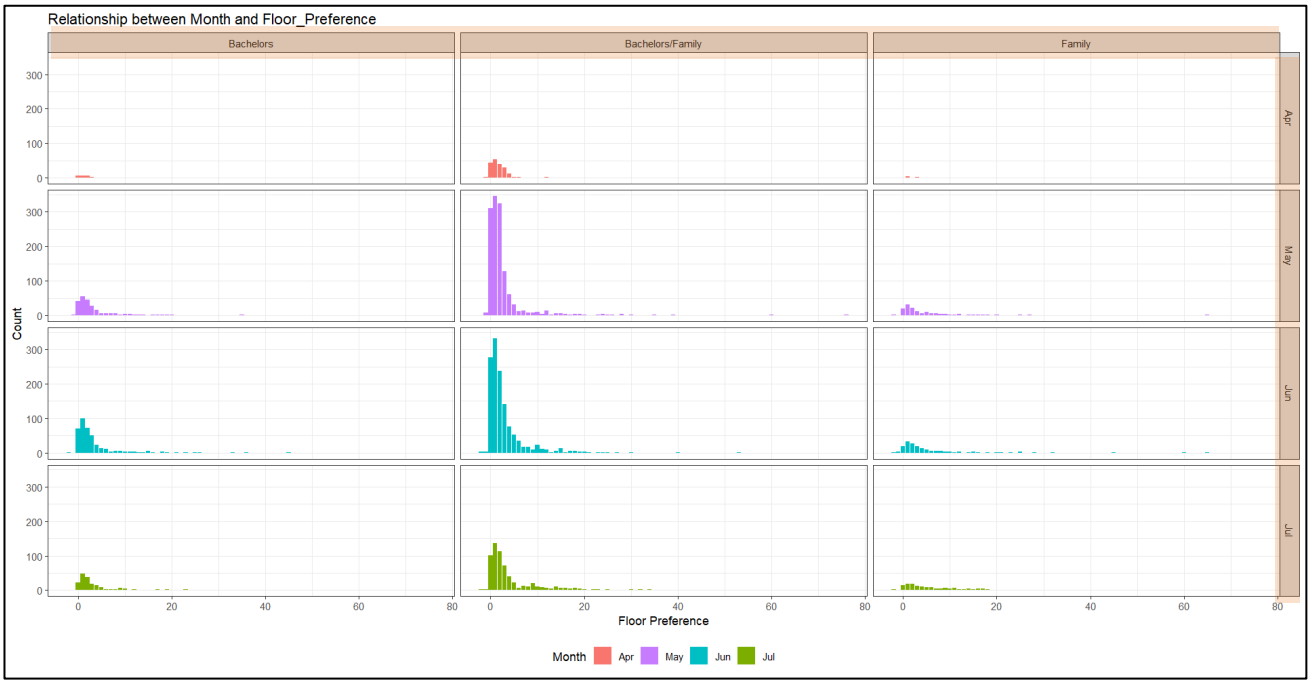
To set the y-axis limit from 0 to 800

## 7.26. facet_grid()

**Code**

```
# 26. facet_grid()
# From Analysis 2-3
ggplot(house_rental_data,aes(x=Floor_Preference,fill=Month)) +
  geom_bar(stat = "count") +
  labs(x="Floor Preference",y="Count",
       title="Relationship between Month and Floor_Preference") +
  scale_fill_discrete(breaks=c('Apr','May','Jun','Jul')) +
  theme_bw() +
  theme(legend.position = "bottom") +
  facet_grid(factor(Month,levels=c('Apr','May','Jun','Jul'))~Tenant_Type)
```

**Output**




**Explanation**

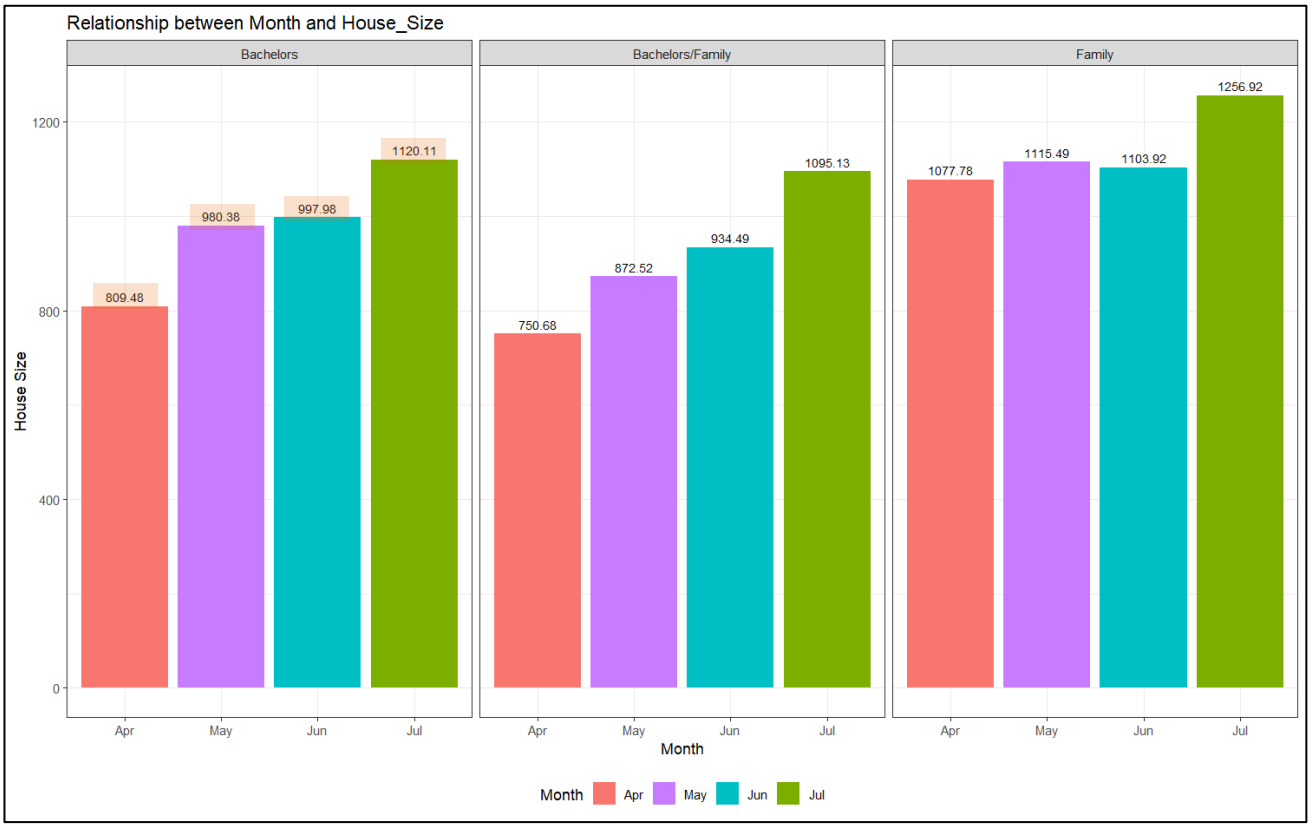
To produce a 2d grid of panels defined by two variables which form the rows and columns

## 7.27. stat_summary()

**Code**

```
# 27. stat_summary()
# From Analysis 2-4
ggplot(house_rental_data,
       aes(x=Month,y=House_Size,fill=Month)) +
  geom_bar(aes(factor(Month,levels=c('Apr','May','Jun','Jul')),House_Size),
           position="dodge",stat="summary",fun="mean") +
  stat_summary(aes(label=round(after_stat(y),2)),fun="mean",geom="text",vjust=-0.5,size=3) +
  labs(x="Month",y="House Size",
       title="Relationship between Month and House_Size") +
  scale_fill_discrete(breaks=c('Apr','May','Jun','Jul')) +
  scale_x_discrete(breaks=c('Apr','May','Jun','Jul')) +
  theme_bw() +
  theme(legend.position = "bottom") +
  facet_wrap(~Tenant_Type)
```

**Output**




**Explanation**

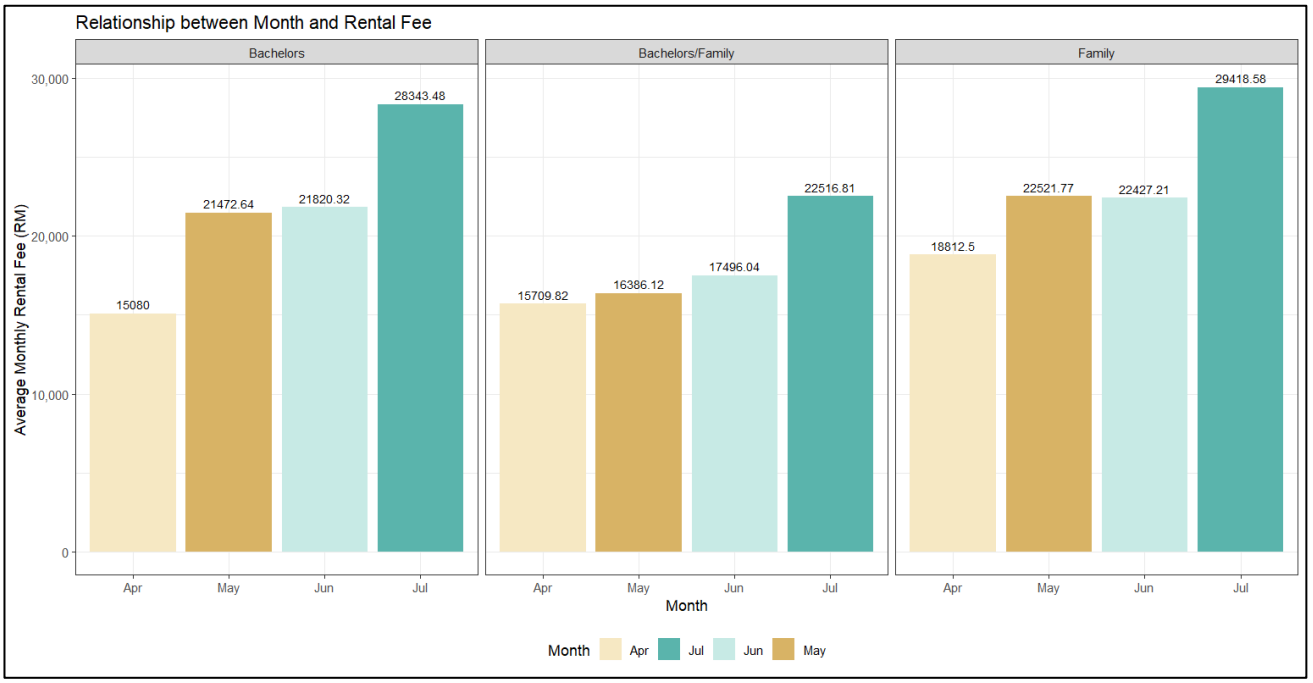
**stat_summary()** is used to add a statistical label to the plots. In this case, the mean of the house size in each month is labelled on top of the related month bar.

## 7.28. scale_fill_manual()

### Code

```
# 28. scale_fill_manual()
# From Analysis 2-5
ggplot(house_rental_data,aes(x=Month,y=Rental_Fee,fill=Month)) +
  geom_bar(aes(factor(Month,levels=c('Apr','May','Jun','Jul')),Rental_Fee),
           position="dodge",stat="summary",fun="mean") +
  stat_summary(aes(label=round(after_stat(y),2)),fun="mean",geom="text",vjust=-0.5,size=3) +
  labs(x="Month",y="Average Monthly Rental Fee (RM)",
       title="Relationship between Month and Rental_Fee") +
  scale_y_continuous(labels=scales::comma) +
  theme_bw() +
  scale_fill_manual(values=c('#f6e8c3','#5ab4ac','#c7eae5','#d8b365')) +
  facet_wrap(~Tenant_Type)
```

### Output




### Explanation

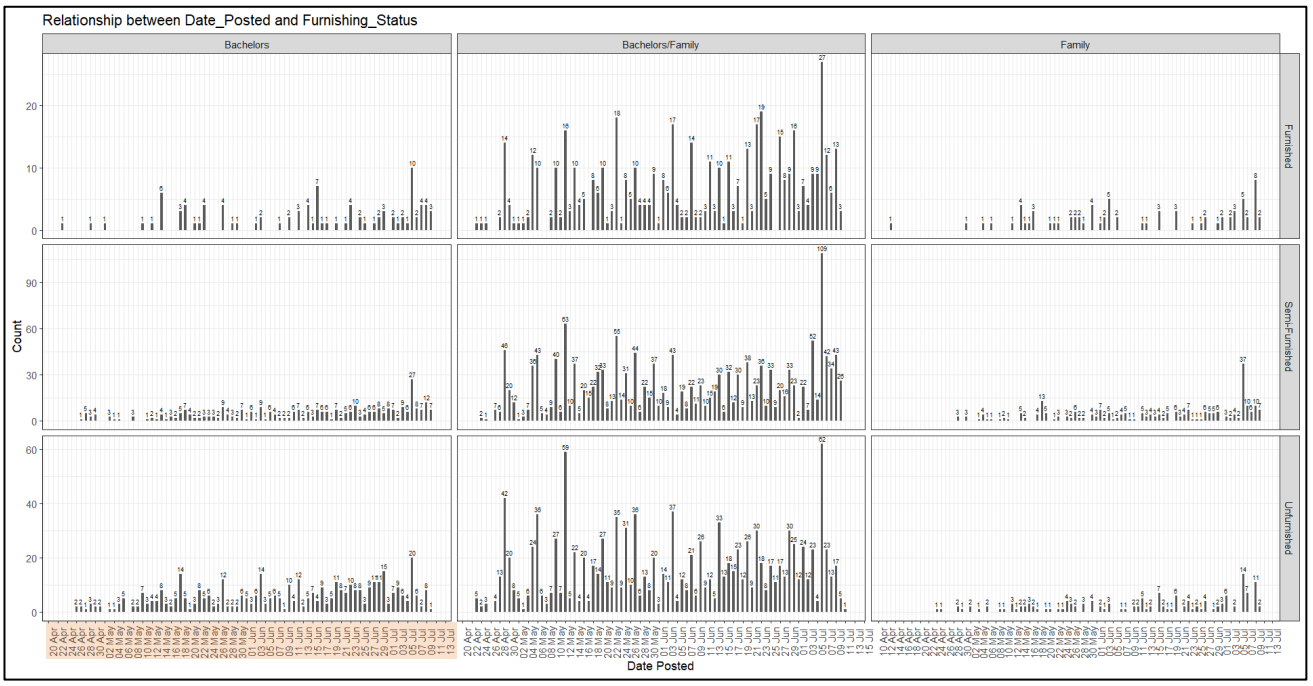
To set the fill colour of the scale manually using hex code of the colour

## 7.29. theme(axis.text.x)

**Code**

```
# 29. theme(axis.text)
# From Analysis 2-6
ggplot(house_rental_data,aes(x=Date_Posted)) +
  geom_bar(aes(y=after_stat(count)),stat="bin",binwidth=0.5) +
  geom_text(aes(label=after_stat(count)),stat="count",vjust=-0.5,size=2) +
  labs(x="Date Posted",y="Count",
       title="Relationship between Date_Posted and Furnishing_Status") +
  scale_x_date(breaks=date_breaks("2 day"),labels=date_format("%d %b")) +
  theme_bw() +
  theme(axis.line=element_line(),axis.text.x=element_text(angle=90)) +
  facet_grid(Furnishing_Status~Tenant_Type,scales="free")
```

**Output**




**Explanation**

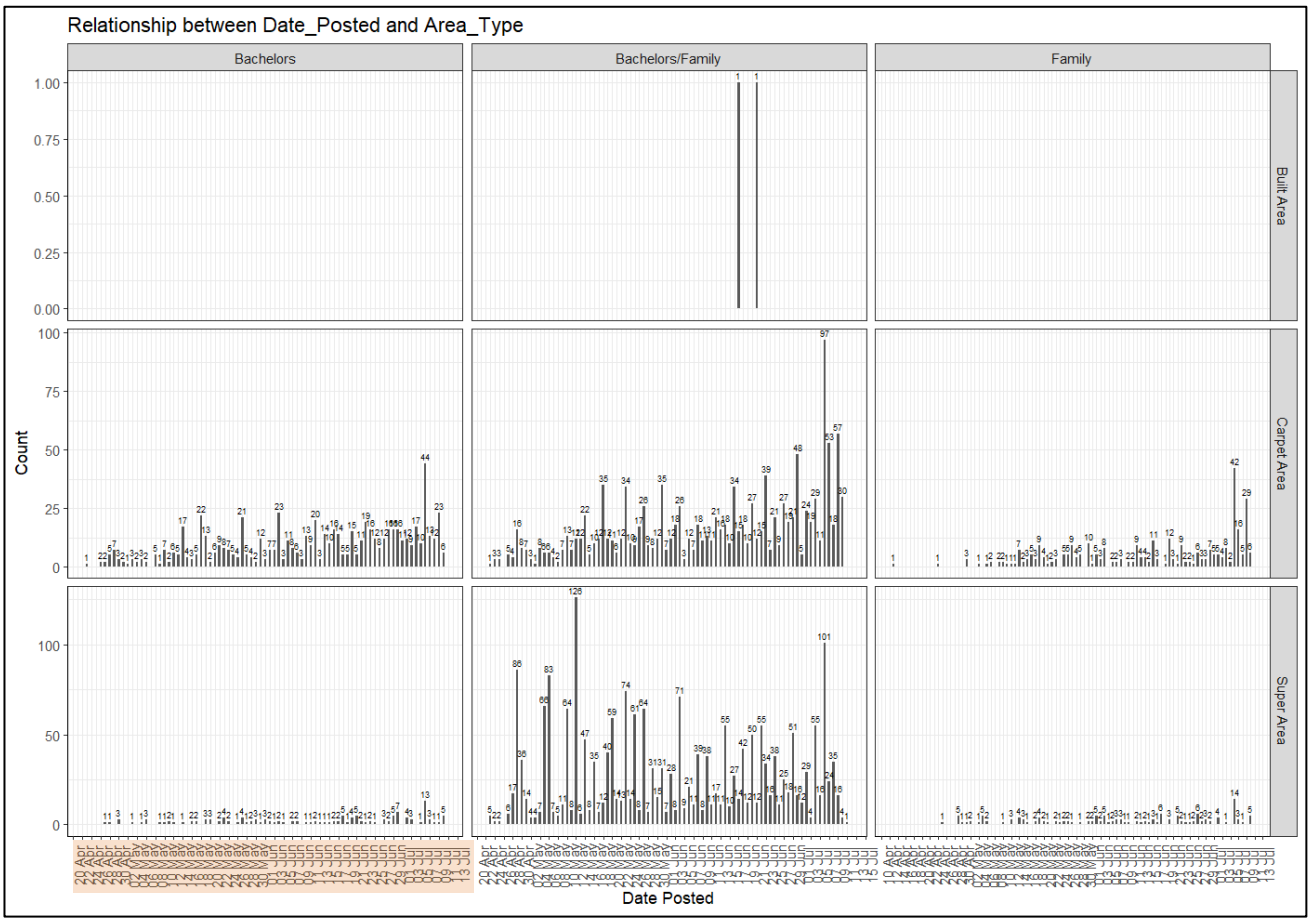
To set the angle of the x-axis label to 90 degrees so that it won't overlap with each other

## 7.30. scale_x_date()

**Code**

```
# 30. scale_x_date()
# From Analysis 2-7
ggplot(house_rental_data,aes(x=Date_Posted)) +
  geom_bar(aes(y=after_stat(count)),stat="bin",binwidth=0.5) +
  geom_text(aes(label=after_stat(count)),stat="count",vjust=-0.5,size=2) +
  labs(x="Date Posted",y="Count",
       title="Relationship between Date_Posted and Area_Type") +
  scale_x_date(breaks=date_breaks("2 day"),labels=date_format("%d %b")) +
  theme_bw() +
  theme(axis.line=element_line(),axis.text.x=element_text(angle=90)) +
  facet_grid(Area_Type~Tenant_Type,scales="free")
```

**Output**




**Explanation**

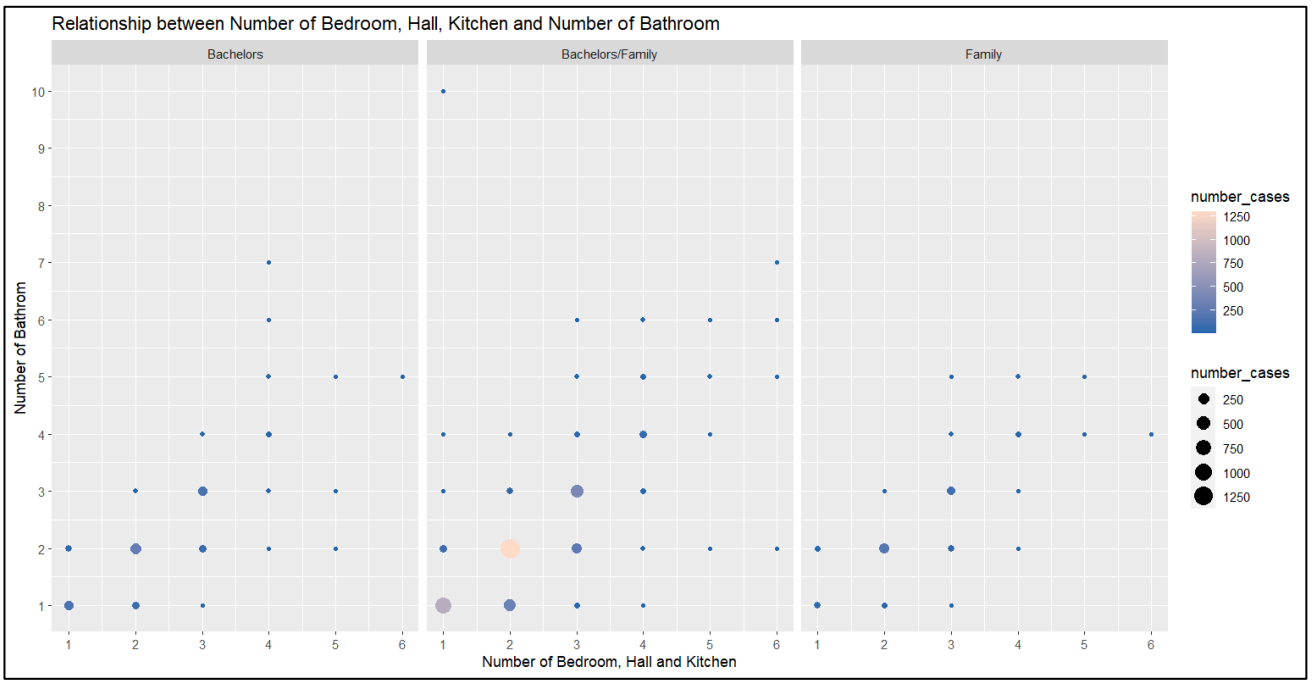
To show only the day and the month with 2 day breaks in between so that the same year do not need to be repeated

## 7.31. Bubble Plot

**Code**

```
# 31. Bubble Plot
# From Analysis 3-1
# calculate Percentage Grouped by Tenant_Type, Bedroom_Hall_Kitchen and Number_of_Bathroom
group_tt_bhk_nob <- house_rental_data %>%
  group_by(Tenant_Type,Bedroom_Hall_Kitchen,Number_of_Bathroom) %>%
  summarise(number_cases=n())
# Scatter Plot
ggplot(group_tt_bhk_nob,aes(x=Bedroom_Hall_Kitchen,y=Number_of_Bathroom)) +
  geom_point(aes(size=number_cases,color=number_cases)) +
  labs(x="Number of Bedroom, Hall and Kitchen",y="Number of Bathroom") +
  ggtitle("Relationship between Bedroom_Hall_Kitchen and Number_of_Bathroom") +
  scale_x_continuous(breaks=seq(1,6,1)) +
  scale_y_continuous(breaks=seq(1,10,1)) +
  scale_color_gradient(low="#2c7fb8",high="pink") +
  facet_wrap(~Tenant_Type)
```

**Output**




**Explanation**

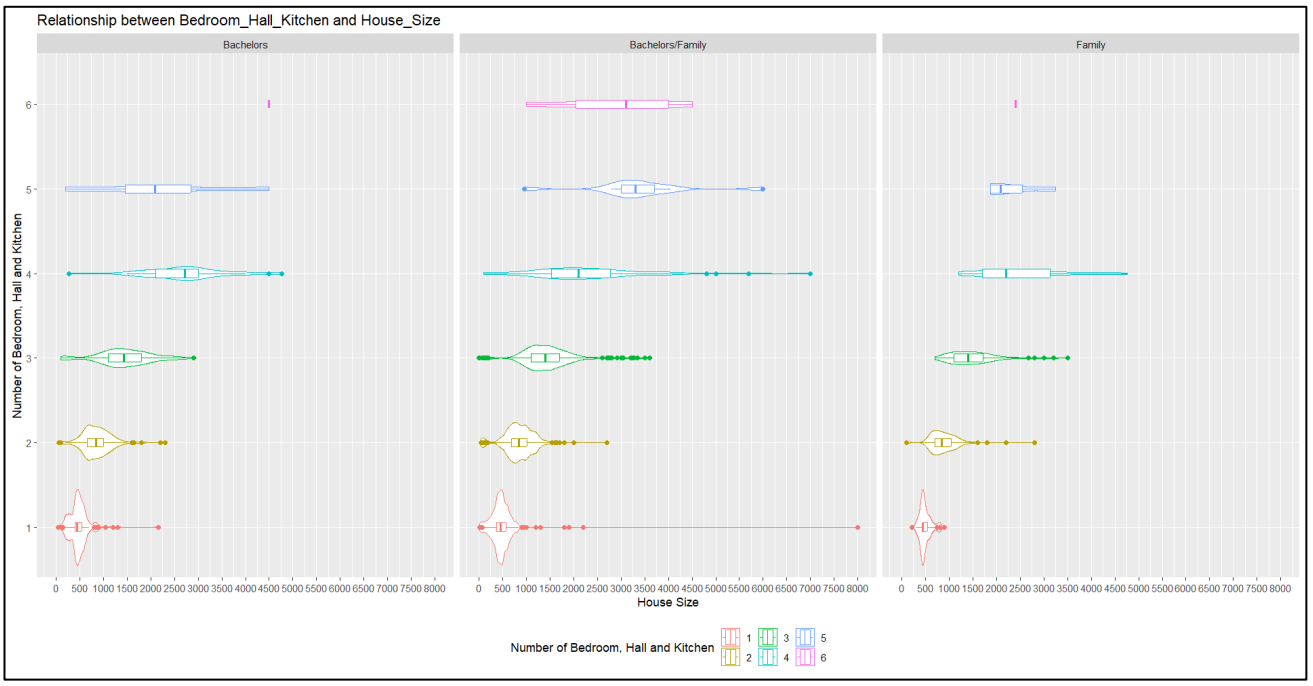
To plot the values of three numerical variables on the same graph for a better visualization

## 7.32. geom_violin()

**Code**

```
# 32. geom_violin()
# From Analysis 3-3
ggplot(house_rental_data,aes(x=House_Size,
                             y=factor(Bedroom_Hall_Kitchen),
                             color=factor(Bedroom_Hall_Kitchen))) +
  geom_violin() +
  geom_boxplot(width=0.1) +
  scale_x_continuous(breaks=seq(0,8000,500)) +
  scale_y_discrete(breaks=seq(1,6,1)) +
  facet_wrap(~Tenant_Type) +
  theme(legend.position = "bottom") +
  labs(x="House Size",y="Number of Bedroom, Hall and Kitchen",
       color="Number of Bedroom, Hall and Kitchen",
       title="Relationship between Bedroom_Hall_Kitchen and House_Size")
```

**Output**




**Explanation**

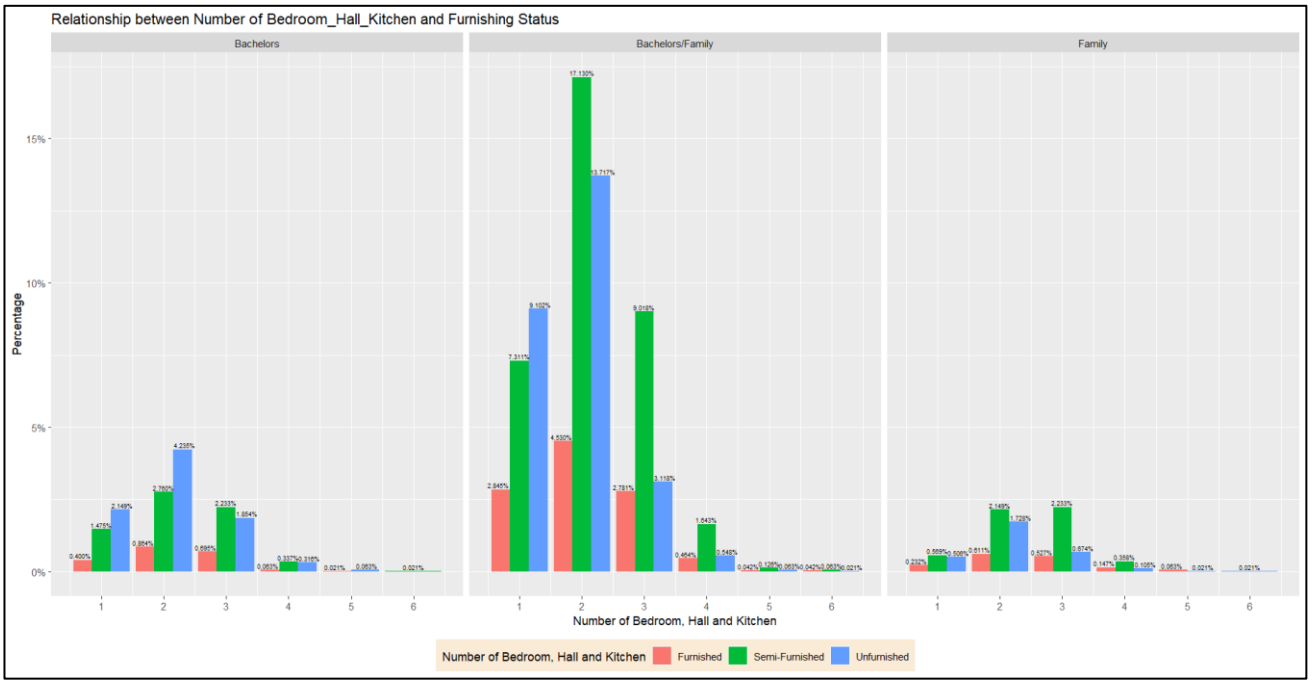
Violin plots enable the visualisation of the distribution of a numeric variable over one or more groups (Holtz, 2022). It is comparable to box plot. It makes it easier to see the distribution of the variables.

## 7.33. position_dodge(width)

**Code**

```
# 33. position_dodge(width)
# From Analysis 3-5
# Calculate Percentage Grouped by Tenant_Type, Bedroom_Hall_Kitchen and Furnishing_Status
group_tt_bhk_fs <- house_rental_data %>%
  group_by(Tenant_Type,Bedroom_Hall_Kitchen,Furnishing_Status) %>%
  count() %>% ungroup() %>% mutate(perc=n/sum(n)) %>% arrange(perc) %>%
  mutate(labels=scales::percent(perc))
# Bar Chart
group_tt_bhk_fs %>% ggplot(aes(Bedroom_Hall_Kitchen,perc,fill=Furnishing_Status)) +
  geom_bar(stat="identity",position="dodge") +
  facet_wrap(~Tenant_Type) +
  labs(title="Relationship between Number of Bedroom_Hall_Kitchen and Furnishing Status") +
  geom_text(aes(label=labels),size=2,vjust=-0.3,position=position_dodge(width=1)) +
  labs(x="Number of Bedroom, Hall and Kitchen",y="Percentage") +
  scale_x_continuous(breaks=seq(1,6,1)) +
  scale_y_continuous(labels=scales::percent) +
  theme(legend.position="bottom",
        legend.background = element_rect(fill="antiquewhite")) +
  scale_fill_discrete(name="Number of Bedroom, Hall and Kitchen")
```

**Output**




**Explanation**

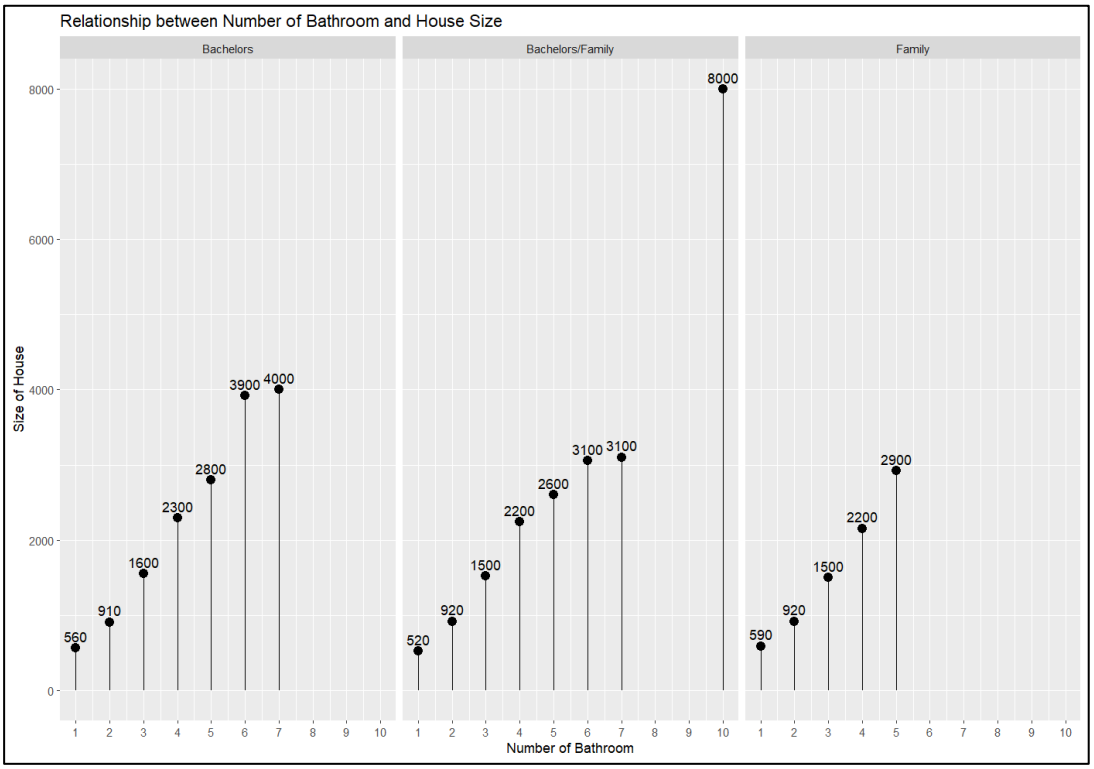
To adjust the position of the label on the dodged bar chart

## 7.34. geom_segment() – Lollipop Graph

**Code**

```
# 34. geom_segment() - Lollipop Graph
# From Analysis 4-2
# Calculate Mean of House_Size Grouped by Tenant_Type and Number_of_Bathroom
group_tt_nob_hs <- house_rental_data %>% group_by(Tenant_Type,Number_of_Bathroom) %>%
  summarise(avg_house_size = mean(House_Size))
# Lollipop Graph
ggplot(group_tt_nob_hs,aes(x=Number_of_Bathroom,y=avg_house_size)) +
  geom_point(size=3,colour="black") +
  geom_segment(aes(x=Number_of_Bathroom,xend=Number_of_Bathroom,y=0,yend=avg_house_size)) +
  geom_text(aes(Number_of_Bathroom,avg_house_size,label=signif(avg_house_size,2),vjust=-0.6)) +
  scale_x_continuous(breaks=seq(1,10,1)) +
  labs(x="Number of Bathroom",y="Size of House",
       title="Relationship between Number of Bathroom and House Size") +
  facet_wrap(~Tenant_Type)
```

**Output**




**Explanation**

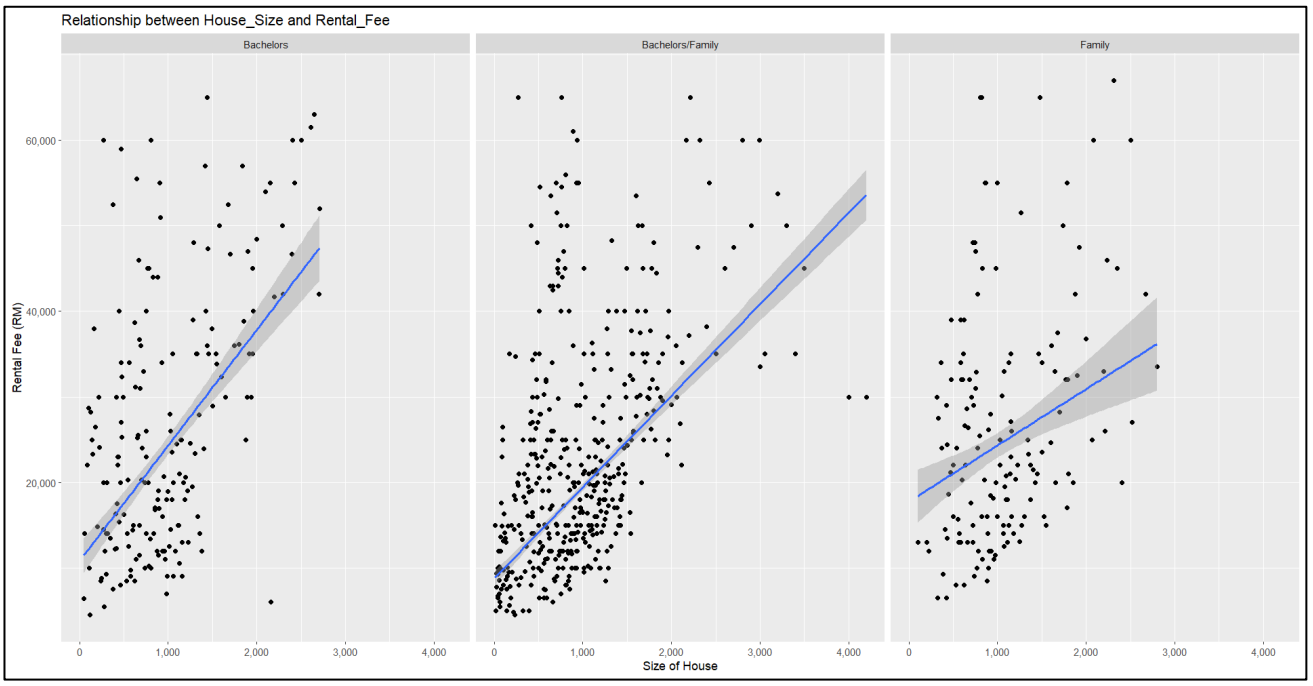
A lollipop graph is essentially a bar plot in which the bars are represented by a dot and a line. It illustrates the connection between a numeric and a categorical variable (Holtz, 2020). Lollipop graph is done using **geom_segment()** by drawing straight line between the coordinates (x, y) and (xend, yend).

## 7.35. geom_smooth(method)

**Code**

```
# 35. geom_smooth(method)
# From Analysis 6-1
# Point Graph
ggplot(rf_no_outliers,aes(x=House_Size,y=Rental_Fee)) +
  geom_point(stat="summary",fun="mean") +
  geom_smooth(method="lm") +
  facet_wrap(~Tenant_Type) +
  scale_x_continuous(labels = scales::comma) +
  scale_y_continuous(labels = scales::comma) +
  labs(x="Size of House",y="Rental Fee (RM)") +
  ggtitle("Relationship between House_Size and Rental_Fee")
```

**Output**




**Explanation**

To draw a best fit line for each plot to see the distribution

## 8.0 Conclusion

Throughout this assignment, I have done 7 questions with 58 analysis and 35 extra features. The analysis included data cleaning, data manipulation, data exploration and data visualisation, Throughout the process, I learned how to read the data from the graph such as violin plot which is one of the extra features that have been added and how to use the data to investigate how multiple situations can affect the tenants' house decision.

## 9.0 References

Abhishek. (2020, January 3). *How to display plot values with comma separated on Ggplotly ?* RStudio Community. Retrieved November 19, 2022, from https://community.rstudio.com/t/how-to-display-plot-values-with-comma-separated-on-ggplotly/48498

Alboukadel. (2020, November 13). *How to easily customize GGPLOT legend for Great Graphics*. Datanovia. Retrieved November 19, 2022, from https://www.datanovia.com/en/blog/ggplot-legend-title-position-and-labels/#rename-legend-labels-and-change-the-order-of-items

Datavizpyr. (2020, January 29). How to make lollipop plot in R with ggplot2? Data Viz with Python and R. Retrieved November 22, 2022, from https://datavizpyr.com/lollipop-plot-in-r-with-ggplot2/

Datavizpyr. (2021, December 31). How to add percentage label on bars in barplot with GGPLOT2. Data Viz with Python and R. Retrieved November 20, 2022, from https://datavizpyr.com/add-percentage-label-on-bars-in-barplot-ggplot2/

Holtz, Y. (2020, November 20). *Lollipop*. the R Graph Gallery. Retrieved November 30, 2022, from https://r-graph-gallery.com/lollipop-plot.html

Holtz, Y. (2022, March 20). *Violin chart*. the R Graph Gallery. Retrieved November 30, 2022, from https://r-graph-gallery.com/violin.html

Jim. (2022, June 7). *How to Count Distinct Values in R*. R-Bloggers. Retrieved November 13, 2022, from https://www.r-bloggers.com/2022/06/how-to-count-distinct-values-in-r/

Kumar, G. S. (2022, July 19). *How to drop columns by name in R?* Spark by {Examples}. Retrieved November 13, 2022, from https://sparkbyexamples.com/r-programming/drop-columns-by-name-in-r/#:~:text=Drop%20R%20Dataframe%20Columns%20by,using%20the%20select()%20method.

Priyank, M. (2022, January 15). How to make violin plots with GGPLOT2 in R? GeeksforGeeks. Retrieved November 21, 2022, from https://www.geeksforgeeks.org/how-to-make-violin-plots-with-ggplot2-in-r/

Schork, J. (2020, April 22). *Split data frame variable into multiple columns in R (3 examples)*. Statistics Globe. Retrieved November 26, 2022, from https://statisticsglobe.com/split-data-frame-variable-into-multiple-columns-in-r

Schork, J. (2021, January 19). *Convert numeric values to month names & Abbreviations R (2 examples)*. Statistics Globe. Retrieved November 17, 2022, from https://statisticsglobe.com/convert-numeric-to-month-names-and-abbreviations-r

Schweinberger, M. (2022, October 30). Data Visualization with R. Ladal. Retrieved November 19, 2022, from https://ladal.edu.au/dviz.html#Pie_charts

Stulp, G. (2019, March 21). R visualization workshop - university of groningen. R visualization workshop. Retrieved November 21, 2022, from https://stulp.gmw.rug.nl/ggplotworkshop/twodiscretevariables.html

Sturis, J. (2021, December 20). *Count in R, more than 10 examples*. Data Cornering. Retrieved November 13, 2022, from https://datacornering.com/count-in-r-more-than-10-examples/

Vermani, G. (2022, August 29). *How to change column headers of a dataframe in R* . ProjectPro. Retrieved November 13, 2022, from https://www.projectpro.io/recipes/change-column-headers-of-dataframe-r#:~:text=colnames%20()%20function%20can%20be,can%20be%20changed%20at%20once.

Zach. (2020, October 12). *How to impute missing values in R (with examples)*. Statology. Retrieved November 13, 2022, from https://www.statology.org/impute-missing-values-in-r/

Zach. (2021, August 13). *How to plot categorical data in R (with examples)*. Statology. Retrieved November 19, 2022, from https://www.statology.org/plot-categorical-data-in-r/

Zach. (2021, July 29). R: How to add column to data frame based on other columns. Statology. Retrieved November 20, 2022, from https://www.statology.org/r-add-column-to-data-frame-based-on-other-columns/

Zach. (2021, June 7). *How to count observations by group in R*. Statology. Retrieved November 18, 2022, from https://www.statology.org/count- oup-by-grr/

Zach. (2021, September 21). *How to find and count missing values in R (with examples)*. Statology. Retrieved November 13, 2022, from https://www.statology.org/r-find-missing-values/

Zach. (2022, April 29). *How to extract month from date in R (with examples)*. Statology. Retrieved November 13, 2022, from https://www.statology.org/extract-month-from-date-in-r/

Zach. (2022, April 7). *How to use STR() function in R (4 examples)*. Statology. Retrieved November 13, 2022, from https://www.statology.org/str-function-in-r/