GROUP ASSIGNMENT

TECHNOLOGY PARK MALAYSIA

CT127-3-2-PFDA

PROGRAMMING FOR DATA ANALYSIS

APD2F2311CS(DA)

HAND OUT DATE: 18 DECEMBER 2023

HAND IN DATE: 31 JANUARY 2024

WEIGHTAGE: 50%

INSTRUCTIONS TO CANDIDATES:

1. **Students are advised to underpin their answers with the use of references (cited using the American Psychological Association (APA) Referencing).**
2. **Late submission will be awarded zero (0) unless Extenuating Circumstances (EC) are upheld.**
3. **Cases of plagiarism will be penalized.**
4. **Submit the assignment to APU Learning Management System.**
5. **You must obtain 50% overall to pass this module**

Group No: 15

Ooi Chong Ming: TP072667

Yeoh Zi Qing Bryan: TP072717

Sim Sau Yang: TP065596

Lim Wen Hann: TP065443

Table of Contents

Table of Figures

1.0 Introduction

1.1 Data Description

In the given dataset which contains the residential properties in Kuala Lumpur, our group mainly focused on the prices, bedrooms of each house, furnished status of the houses, built area of the houses and the square feet of the houses. There are other variables in the dataset as well such as bathrooms, car parks and more. We cleaned out all the unclean and filled out all the unknown data from the dataset before we proceeded to the analysis part of the assignments.

1.2 Assumptions

The following assumptions have been made for the data analysis process of the Kuala Lumpur residential property dataset:

1. The data analysis will exclusively focus on key variables relevant in the hypothesis and objectives, which include Location, Price, Rooms, Size, and Furnishing Status. Consequently, variables such as Bathrooms and Car Parks will be disregarded during the data analysis process.
2. Records containing two or more empty or null values will be considered incomplete and will be subsequently excluded during the data cleaning process.
3. Records with the location listed only as "Kuala Lumpur" without a specific neighborhood designation (Bukit Jalil, KLCC) will be excluded due to their more generalized nature.
4. If the number of bedrooms in a property is listed as studio, it is assumed that the number of bedrooms is one.
5. Any aberrant or uninterpretable values found in the size column such as "Unknown", "Malaysia", "NIL" will be replaced with "NA".
6. Property sizes provided in units other than square feet (acre, meter, hectare) will be converted to square feet for standardization and consistency during the analysis.
7. Ranges in the property size columns (1000 - 2000) will be replaced with the mean value of said records.
8. During the data validation process, outliers in the Price, Rooms and Size columns are identified and addressed with the Interquartile Range (IQR) method. The upper fence, representing values exceeding 1.5 times the IQR from the third quartile (Q_3), is used as a threshold for identifying outliers. Any data values exceeding this cut-off point will be subsequently replaced with the third quartile value. Same goes to the lower fence, representing values lower than 1.5 times the IQR from the first quartile (Q_1), is also use as a threshold for the lower outliers.
9. Records with missing or zero values in the Price, Rooms and Size columns are replaced with the mean value of their respective columns.

10. Records containing empty or unknown values in the Furnishing and Type columns are substituted with the most frequently occurring values in their respective columns.

1.3 Hypothesis and Objectives

Based on the Kuala Lumpur residential properties dataset, the team has formulated a hypothesis to identify the property trends and provide useful recommendations to stakeholders. The hypothesis posits that 70% of houses in KLCC with price greater than RM 1,000,000 are fully furnished, built-up, within the square feet of 1000 to 1800 with more than 3 bed rooms. The main objectives of this in-depth analysis are as follows:

1. To determine the impact of built area on property price in KLCC.
2. To determine the impact of furnishing status on property price in KLCC.
3. To determine the impact of number of bedrooms on property price in KLCC.
4. To determine the impact of property size (square feet) on property price in KLCC.

2.0 Data Preparation

2.1 Data Import

```
# Assigning Library in use
library(tidyr)
library(stringr)
library(DescTools)
library(ggplot2)

#-----------------------------------Data Import-----------------------------------
filepath = "C:\\Users\\Lenovo\\Desktop\\5. kl_property_data.csv"
kl_property_data = read.csv(filepath, header=TRUE)
```

Figure 1-2.1.1: Import library & dataset

Libraries used in data preprocessing and KL property dataset are imported to R Studio.

2.2 Data Cleaning / Pre-processing

2.2.1 Forming subset with only variables we need

```
# Forming a subset with only variables we need
Filtered_dataset <- subset(kl_property_data,select=c(Location,Price,Rooms,Size,Furnishing))
```

Figure 2-2.2.1.1: Selecting columns required

The only columns needed in our data analysis, which are location, price, rooms, size, and furnishing are taken out.

	Location	Price	Rooms	Size	Furnishing
1	KLCC, Kuala Lumpur	RM1,250,000	2+1	Built-up : 1,335 sq. ft.	Fully Furnished
2	Damansara Heights, Kuala Lumpur	RM6,800,000	6	Land area : 6900 sq. ft.	Partly Furnished
3	Dutamas, Kuala Lumpur	RM1,030,000	3	Built-up : 1,875 sq. ft.	Partly Furnished
4	Cheras, Kuala Lumpur				

Figure 3-2.2.1.2: Result of Filtered_dataset

2.2.2 Checking number of NA and empty value in each column

```
# Check number NA and "" in each Column
colSums(is.na(Filtered_dataset))
colSums(Filtered_dataset== "",na.rm=TRUE)
```

```
> colSums(is.na(Filtered_dataset))
  Location      Price      Rooms       Size Furnishing
         0          0          0          0          0
> colSums(Filtered_dataset== "",na.rm=TRUE) #0
  Location      Price      Rooms       Size Furnishing
         0        248       1706       1063       6930
```

Figure 4-2.2.2.1: Check null & empty string value

```
#Replace the empty values to NA for columns: "Price", "Rooms", "Size", "Furnishing"
Filtered_dataset$Price[Filtered_dataset$Price ==""] = NA
Filtered_dataset$Rooms[Filtered_dataset$Rooms ==""] = NA
Filtered_dataset$Size[Filtered_dataset$Size ==""] = NA
Filtered_dataset$Furnishing[Filtered_dataset$Furnishing ==""] = NA
```

Figure 5- 2.2.2.2: Replacing empty string to null

There are many empty values in the Price, Rooms, Size and Furnishing column. Thus, we decided to replace all the empty values into NA.

2.2.3 Removing records with 2 or more NA values or empty values

```
# Delete the record when 2 or more NA or 2 or ore empty values in the columns: "Price", "Rooms","Size", "Furnishing"
# >= 2 NA value in a record is considered useless
Filtered_dataset <- Filtered_dataset[rowSums(is.na(Filtered_dataset[, c("Price", "Rooms", "Size", "Furnishing")])
                                             | Filtered_dataset[, c("Price", "Rooms", "Size", "Furnishing")] == "") <= 1, ]
```

Figure 6- 2.2.3.1: Removing meaningless record

All the records with 2 or more null values are known as unmeaningful data and significantly flawed data. Those records are removed from the dataset.

2.2.4 Remove locations that only contains “Kuala Lumpur”

17208	Kuala Lumpur	RM758,000	5	Land area : 22x65 sq. ft.	Partly Furnished
41887	Kuala Lumpur	RM990,000	5		Partly Furnished

```
# Remove the location that only contains "Kuala Lumpur"
Filtered_dataset = Filtered_dataset[Filtered_dataset$Location != "Kuala Lumpur",]
```

Figure 7-2.2.4.1: Remove unspecified location records

In the location column, there are several records containing only “Kuala Lumpur”, where its exact location is not specified. Thus, we decided to remove it from our dataset.

2.2.5 Separating “Location” column into “Location” and “State”

```
# Separating the "Location" column into "Location" and "State"
Filtered_dataset = separate(Filtered_dataset, col=Location, into=c("Location", "State"),sep=", ", extra = "merge")
```

Figure 8-2.2.5.1: Split Location column into Location & State

The location column is separated into two, which are location and state, using “separate” function.

2.2.6 Handling special records in the Location column

30171	Taman Yarl	UOG, Kuala Lumpur	RM4,000,000	5+1	Land area : 9440 sq. ft.	Partly Furnished

```
# Handling the record where the State contains "UOG, Kuala Lumpur" by moving "UOG" to Location Column
# Concatenate "UOG" into the Location column of the targeted row
Filtered_dataset$Location[grepl("UOG", Filtered_dataset$State)] <-
  paste(Filtered_dataset$Location[grepl("UOG", Filtered_dataset$State)], "UOG", sep = ", ")

# Removing the "UOG" and "," in the State column of the targeted row
Filtered_dataset$State[grepl("UOG", Filtered_dataset$State)] <-
  gsub("UOG, ", "", Filtered_dataset$State[grepl("UOG", Filtered_dataset$State)])
```

Figure 9-2.2.6.1: “UOG” in State Column

A special record where the “UOG” is set into State column because there is a ‘,’ before it. The records with “UOG” in State column are taken and added back into the “Location” column.

30171	Taman Yarl, UOG	Kuala Lumpur	RM4,000,000	5+1	Land area : 9440 sq. ft.	Partly Furnished

Figure 10-2.2.6.2: Result of “UOG” in Location column

2.2.7 Capitalizing the Location and State column

```
# Making all the values in the "Location" and "State" column into Capitalized Form(The first letter is capital and the rest is not)
Filtered_dataset$Location = str_to_title(Filtered_dataset$Location)
Filtered_dataset$State = str_to_title(Filtered_dataset$State)
```

Figure 11-2.2.7.1: Reset the casing in Location & State columns

The casing in Location and State column are set. The first letter of each word is uppercase, while the following letter is lowercase using the function “str_to_title”.

	Location	State	Price	Rooms	Size	Furnishing
1	Klcc	Kuala Lumpur	RM1,250,000	2+1	Built-up : 1,335 sq. ft.	Fully Furnished
2	Damansara Heights	Kuala Lumpur	RM6,800,000	6	Land area : 6900 sq. ft.	Partly Furnished
3	Dutamas	Kuala Lumpur	RM1,030,000	3	Built-up : 1,875 sq. ft.	Partly Furnished

Figure 12-2.2.7.2: Result of Casing

2.2.8 Checking for properties not in Kuala Lumpur

```
# Check whether there are places not in Kuala Lumpur
Filtered_dataset[!grepl("Kuala Lumpur", Filtered_dataset$State), ]
[1] Location   State      Price      Rooms      Size       Furnishing
<0 rows> (or 0-length row.names)
```

Figure 13-2.2.8.1: Looking for non-Kuala Lumpur Record

“grepl” function is used to find records containing the given string. The result shows no properties are outside Kuala Lumpur.

```
# It is known that all locations are in Kuala Lumpur, hence the state column is useless
# Remove the State Column
Filtered_dataset <- subset(Filtered_dataset, select = -State)
```

	Location	Price	Rooms	Size	Furnishing
1	Klcc	RM1,250,000	2+1	Built-up : 1,335 sq. ft.	Fully Furnished
2	Damansara Heights	RM6,800,000	6	Land area : 6900 sq. ft.	Partly Furnished
3	Dutamas	RM1,030,000	3	Built-up : 1,875 sq. ft.	Partly Furnished

Figure 14-2.2.8.2: Removing State Column and its Result

The state column is removed as all the properties are within Kuala Lumpur.

2.2.9 Removing non-numeric characters in Price column and convert it into numeric class.

```
# Removing the non-numeric characters in the "Price" column: "RM", " ", ","
Filtered_dataset$Price <- gsub("[^0-9]", "", Filtered_dataset$Price)
```

Figure 15-2.2.9.1: Removing non-numerical characters

In Price column, all the non-numerical values and characters in price like “RM” and ‘,’ are replaced with empty string “” using regex and “gsub” function.

	Location	Price	Rooms	Size	Furnishing
1	Klcc	1250000	2+1	Built-up : 1,335 sq. ft.	Fully Furnished
2	Damansara Heights	6800000	6	Land area : 6900 sq. ft.	Partly Furnished
3	Dutamas	1030000	3	Built-up : 1,875 sq. ft.	Partly Furnished

```
> # Convert the price column class from character to numeric
> class(Filtered_dataset$Price)
[1] "character"
> Filtered_dataset$Price = as.numeric(Filtered_dataset$Price)
> class(Filtered_dataset$Price)
[1] "numeric"
```

Figure 16-2.2.9.2: Convert price column type from character to numeric

The Price column is then converted to the numeric clasc.

2.2.10 Generating the new column “IsStudio”

	Location	Price	Rooms	Size	Furnishing
183	Klcc	1000000	Studio	Built-up : 657 sq. ft.	Fully Furnished
219	KI City	1207000	Studio	Built-up : 685 sq. ft.	Partly Furnished

Figure 17-2.2.10.1: Studio in Rooms column

```
# Add a new column "IsStudio" to define the property is studio
Filtered_dataset$IsStudio <- ifelse(!is.na(Filtered_dataset$Rooms) & Filtered_dataset$Rooms == "Studio", TRUE, FALSE)
# Moving "IsStudio" Column to 4th column
Filtered_dataset <- Filtered_dataset[, c(1:3, ncol(Filtered_dataset), 4:(ncol(Filtered_dataset)-1))]
```

Figure 18-2.2.10.2: Create a new column named “IsStudio”

In the Rooms column, there are rows written with “Studio”. We decided to create a new logical column named “IsStudio” which gives value “TRUE” when the property is studio house and vice versa. The IsStudio column is moved from last to the 4th column.

	Location	Price	Rooms	IsStudio	Size	Furnishing
1	Klcc	1250000	2+1	FALSE	Built-up : 1,335 sq. ft.	Fully Furnished
2	Damansara Heights	6800000	6	FALSE	Land area : 6900 sq. ft.	Partly Furnished

Figure 19-2.2.10.3: Result of new columns

```
# Replace the value in "Rooms" column that has Studio with 1
Filtered_dataset$Rooms[Filtered_dataset$Rooms == "Studio"] = 1
```

Figure 20-2.2.10.4: Replace “Studio” with 1

The “Studio” in the Rooms column is replaced with 1 as we assume Studio houses only have one room.

2.2.11 Solving special cases in Rooms column

```
# Calculate the sum of rooms for entries with the symbol "+" and convert to integer
Filtered_dataset$Rooms <- sapply(strsplit(Filtered_dataset$Rooms, "\\+"), function(x) sum(as.integer(x)))
```

Figure 21-2.2.11.1: Solving special cases in Rooms column

There are values separated with '+' which we assume means plus. By using strsplit, each element is split using separator "+" to get the two values and "sapply" is used to apply the "function(x) sum(as.integer(x))" to converts the splitted string into integer and add both of them up.

53881	Bangsar	5500000	6	FALSE	Land area : 7168 sq. ft.	Partly Furnished
53882	Wangsa Maju	480000	3	FALSE	Built-up : 1,150 sq. ft.	Unfurnished

Figure 22-2.2.11.2: Result of Cleaned Rooms Column

2.2.12 Separating Size column into Type and Size

```
#Separating the "Size" column into "Type" and "Size"
Filtered_dataset=separate(Filtered_dataset, col=Size, into=c("Type", "Size"),sep=" : ")
```

Figure 23-2.2.12.1: Separating Size column into Type & Size

The Size column is seperated into Type and Size with " : " as the delimiter to distinguish the Built type and the Size of the property.

	Location	Price	Rooms	IsStudio	Type	Size	Furnishing
1	Klcc	1250000	3	FALSE	Built-up	1,335 sq. ft.	Fully Furnished
2	Damansara Heights	6800000	6	FALSE	Land area	6900 sq. ft.	Partly Furnished

Figure 24-2.2.12.2: Result of separation

2.2.13 Checking the elements in the Type column

```
Levels: Built-up Land area
> # Check Type Column
> nlevels(factor(Filtered_dataset$Type))
[1] 2
```

Figure 25-2.2.13.1: Check levels of factor of Type column

Using the "factor" and "nlevels" function, there are 2 types of factors in the Type column, and it can be believed that they are Built-up and Land area. The Type column is well-prepared.

2.2.14 Solving abnormal characters in the Size column

10383	Pantai	480000	1	FALSE	Built-up	750 sq. ft.	Partly Furnished
10384	Seputeh	1480000	5	FALSE	Land area	20×80 sq. ft.	Partly Furnished
10385	Pantai	2600000	5	FALSE	Built-up	5,500 sq. ft.	Fully Furnished

Figure 26-2.2.14.1: Abnormal character in Size column

```
abnormal_characters <- c(" ", "CornerUnit", "corner", "sq\\.ft.", "sqft", "sq\\.ft","sf", "SF", "ft", "\\(.*"
                         , "approx", ",", "~", "'", "@", "or.*", "/.*", "&#215;", "&#8217;", "&#8221;", "X", "x", "arce")
replacements <- c("", "", "", "", "", "", "", "", "", "", "", "","", "", "", "", "", "", "", "", "", "\\*", "\\*", "acre")

View(Filtered_dataset)
# Apply replacements to the 'Size' column
Filtered_dataset$Size <- str_replace_all(Filtered_dataset$Size, setNames(replacements, abnormal_characters))
```

Figure 27-2.2.14.2: Define all abnormal character and define replacement character

In Size column, except sq. ft., there are many abnormal characters. The values cannot be replaced by empty string as there are values with different unit such as meter, acres, and hectare, and the expression. These characters are defined, and stored into a vector, and a replacement vector for each respective element in the abnormal_character vector is created. Then, the str_replace_all function is applied to replace all the abnormal characters into their respective replacement.

```
# Replace abnormal values to NA
abnormal_values <- c("", "WP", "WilayahPersekutuan", "unknown", "nil", "NA", "N/A", "Malaysia", "KualaLumpur", "-",
                     "0", "N", "27**", "2000+")
Filtered_dataset$Size[Filtered_dataset$Size %in% abnormal_values] = NA
```

Figure 28- 2.2.14.3: Replace remaining abnormal value

There are some values which their size is not able to be defined. They are stored in the abnormal_values vector and replaced by NA.

38155	Cheras	1650000	5	FALSE	Land area	50*80acres	Unfurnished
38156	Jalan Klang Lama (Old Klang Road)	600000	2	FALSE	Built-up	852	Partly Furnished

```
# Convert acre, meter, and hectare to sq ft
acres = gsub("acre.*","",Filtered_dataset$Size[grepl("acre", Filtered_dataset$Size)])
meter = gsub("sq\\.m.","", Filtered_dataset$Size[grepl("m", Filtered_dataset$Size)])
hectare = gsub("hectare","", Filtered_dataset$Size[grepl("hectare", Filtered_dataset$Size)])
Filtered_dataset$Size[grepl("acre", Filtered_dataset$Size)] = sapply(acres, function(expr) eval(parse(text = expr)))*43560
Filtered_dataset$Size[grepl("m", Filtered_dataset$Size)] = sapply(meter, function(expr) eval(parse(text = expr)))*10.7639104
Filtered_dataset$Size[grepl("hectare", Filtered_dataset$Size)] = sapply(hectare, function(expr) eval(parse(text = expr)))*107639.104
```

Figure 29-2.2.14.4: Convert other units values into sqft

Specific records of units are taken out, remove the unit, and parse the string to an expression to calculate the value (if the value contains '*' or '+' symbol), and lastly converted to the unit of square feet using their respective formulas.

39360	Kepong	930000	3	FALSE	Land area	16*55wt19	Unfurnished

```
# Clean unique data
Filtered_dataset$Size[grepl("wt", Filtered_dataset$Size)] = NA
Filtered_dataset$Size = gsub("t", "", Filtered_dataset$Size)
Filtered_dataset$Size = gsub("q\\.\\.", "", Filtered_dataset$Size)
```

Figure 30-2.2.14.5: Handling remaining abnormal values

The remaining abnormal values are replaced with empty string one by one.

366	Sentul	320000	4	FALSE	Land area	646~1001	Unfurnished

```
calculate_average <- function(expr, char) {
  values <- as.numeric(strsplit(expr, "-")[[1]])
  mean(values)
}

calculate_average2 <- function(expr, char) {
  values <- as.numeric(strsplit(expr, "~")[[1]])
  mean(values)
}

# Get the average of the value
Filtered_dataset$Size[grepl("-", Filtered_dataset$Size)] = sapply(Filtered_dataset$Size[grepl("-", Filtered_dataset$Size)], calculate_average)
Filtered_dataset$Size[grepl("~", Filtered_dataset$Size)] = sapply(Filtered_dataset$Size[grepl("~", Filtered_dataset$Size)], calculate_average2)
```

Figure 31-2.2.14.6: Solving range value

The range values that are expressed using '~' and '-' are solved by calculating their average value.

366	Sentul	320000	4	FALSE	Land area	823.5	Unfurnished

Figure 32-2.2.14.7: Result of average of range value

```
# Calculate the size by add and multiply with symbol + & *
Filtered_dataset$Size = sapply(Filtered_dataset$Size, function(expr) eval(parse(text = expr)))
```

451	Sungai Besi	2250000	6	FALSE	Land area	40+30*80	Partly Furnished
452	Taman Tun Dr Ismail	4800000	6	FALSE	Land area	8300	Partly Furnished
453	Taman Tun Dr Ismail	1890000	5	FALSE	Land area	20*95	Partly Furnished

Figure 33-2.2.14.8: Parse remaining value into expression

The remaining strings with "*" and "+" expression are parsed into the function expression to calculate the result.

451	Sungai Besi	2250000	6	FALSE	Land area	2440.0	Partly Furnished
452	Taman Tun Dr Ismail	4800000	6	FALSE	Land area	8300.0	Partly Furnished
453	Taman Tun Dr Ismail	1890000	5	FALSE	Land area	1900.0	Partly Furnished

Figure 34-2.2.14.9: Calculated final result

2.2.15 Checking the elements in the Furnishing column

```
Levels: Fully Furnished Partly Furnished Unfurnished Unknown
> # Check Furnishing Column
> nlevels(factor(Filtered_dataset$Furnishing))
[1] 4
```

Figure 35-2.2.15.1: Check levels of factor of Furnishing column

There are 4 levels which are fully furnished, partly furnished, unfurnished, and unknown without any other noise data.

2.3 Data Validation

2.3.1 Checking the number of NA values in each column and other details of each column

```
# Check NA values in every column of the dataset
colSums(is.na(Filtered_dataset))
```

```
> colSums(is.na(Filtered_dataset))
  Location      Price      Rooms   IsStudio       Type       Size Furnishing
         0        165        603          0        720        861       5571
```

Figure 36-2.3.1.1: Checking null value

To start off data validation, the "NA" values in each column are determined.

```
# Check the details of each column
summary(Filtered_dataset)
```

```
> summary(Filtered_dataset)
   Location            Price             Rooms          IsStudio          Type              Size          Furnishing
 Length:52499       Min.   :3.080e+02   Min.   : 0.000   Mode :logical   Length:52499       Min.   : 0.000e+00   Length:52499
 Class :character   1st Qu.:5.800e+05   1st Qu.: 3.000   FALSE:51629     Class :character   1st Qu.: 1.018e+03   Class :character
 Mode  :character   Median :9.800e+05   Median : 3.000   TRUE :870       Mode  :character   Median : 1.426e+03   Mode  :character
                    Mean   :1.895e+06   Mean   : 3.635                                      Mean   :3.704e+129
                    3rd Qu.:1.920e+06   3rd Qu.: 4.000                                      3rd Qu.: 2.378e+03
                    Max.   :1.980e+09   Max.   :18.000                                      Max.   :1.913e+134
                    NA's   :165         NA's   :8                                           NA's   :861
```

Figure 37-2.3.1.2: Summary of dataset

The details of each column are determined and recorded for further uses.

2.3.2 Checking the outliers in Price, Size and Rooms column

```
# Checking the outliers in "Price","Size" and "Rooms" column
boxplot(Filtered_dataset$Price,ylab = "Price",main = "Price")
boxplot(Filtered_dataset$Size,ylab = "Size",main = "Size")
boxplot(Filtered_dataset$Rooms,ylab = "Rooms",main = "Rooms")
```

Figure 38-2.3.2.1: Boxplot Price, Size and Rooms to look for outliers

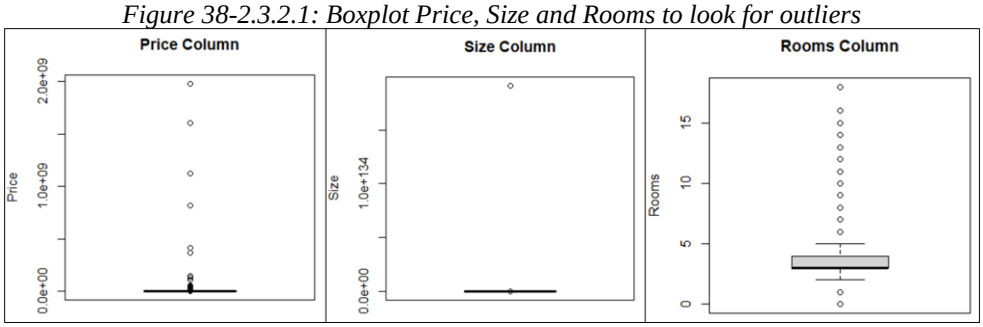


Figure 39-2.3.2.2: Boxplot of Price, Size, and Rooms

There are some outliers' issue in the Price, Size and Rooms column.

2.3.3 Calculating the maximum and minimum outliers in the Price column

```
# Calculating the maximum and minimum outliers of the "Price" column
IQR_Price <- 1920000-580000
MaxOut_Price <- 1920000+(1.5*IQR_Price)
MinOut_Price <- 580000-(1.5*IQR_Price) #Since that the MinOut_Price is negative, means that there are no low outliers
```

Figure 40-2.3.3.1: Calculating outliers in the Price column

The maximum and minimum outliers in the Price column are calculated using the formula stated below:

$$IQR = Q_3 - Q_1$$

Maximum Outlier $= Q_3 + (1.5 \times IQR)$

Minimum Outlier $= Q_1 - (1.5 \times IQR)$

where $Q_3 =$ third quartile, $Q_1 =$ first quartile and $IQR =$ interquartile range.

```
> IQR_Price
[1] 1340000
```

```
> MaxOut_Price
[1] 3930000
```

```
> MinOut_Price
[1] -1430000
```

Figure 41-2.3.3.2: Calculated maximum and minimun outliers' price

Since the minimum outlier of the Price column is negative which is illogical, thus, it is assumed that there are no lower outliers in the Price column.

2.3.4 Calculating the mean of the Price column

```
# Calculate the mean of the "Price" column
mean_price <- mean(Filtered_dataset$Price,na.rm=TRUE)
```

```
> mean_price
[1] 1894731
```

Figure 42-2.3.4.1: Calculate mean

Using the "mean" function, the value of mean calculated is RM 1 894 731.

2.3.5 Replacing NA values in the Price column with mean

```
# Replace NA values in "Price" column  with the mean
Filtered_dataset$Price[is.na(Filtered_dataset$Price)] <- mean_price
```

Figure 43-2.3.4.2: Replace null in price column

The NA values in the Price column are then replaced by the mean calculated.

2.3.5 Checking the new 3rd quartile value

```
# Check the new 3rd Quartile value
summary(Filtered_dataset$Price)
```

```
> summary(Filtered_dataset$Price)
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
3.080e+02 5.800e+05 9.800e+05 1.895e+06 1.900e+06 1.980e+09
```

Figure 44-2.3.5.1: Determine 3^{th} quartile value

Using the "summary" function. The 3rd quartile value had changed from 1 920 000 to 1 900 000.

2.3.6 Replacing the upper outliers with new 3rd quartile value and checking the final result

```
# Replacing upper outliers in the "Price" column with the 3rd quartile
Filtered_dataset$Price[Filtered_dataset$Price >= MaxOut_Price] <- 1900000
```

Figure 45-2.3.6.1: Replace upper outliers

The upper outliers is replaced by the new 3rd quartile value calculated.

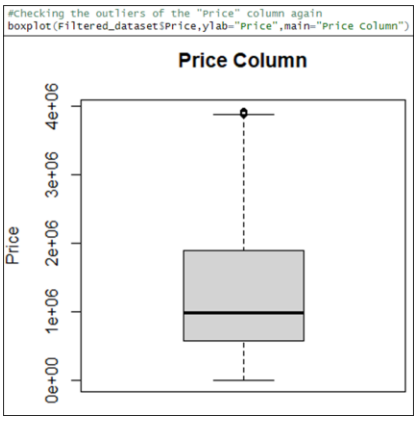


Figure 46-2.3.6.2: Box Plot of price after validation

In the boxplot above, the outliers of the Price column are solved successfully.

2.3.7 Calculating the maximum and minimum outliers in the Rooms column

```
# Calculating the maximum and minimum outliers of the "Rooms" column
IQR_Rooms <- 4-3
MaxOut_Rooms <- 4+(1.5*IQR_Rooms)
MinOut_Rooms <- 3-(1.5*IQR_Rooms)
```

Figure 47-2.3.7.1: Calculating the maximum and minimum outliers in Rooms column

```
> IQR_Rooms
[1] 1
```

```
> MaxOut_Rooms
[1] 5.5
```

```
> MinOut_Rooms
[1] 1.5
```

Figure 48-2.3.7.2: Result of maximum and minimum outliers in Rooms column

The maximum and minimum outliers in the Rooms column are calculated using the same formula as stated in 2.3.3.

2.3.8 Calculating the mean of the Rooms column

```
# Calculate the mean of the "Rooms" column
mean_rooms <- round(mean(Filtered_dataset$Rooms,na.rm = TRUE))
```

```
> mean_rooms
[1] 4
```

Figure 49-2.3.8: Calculate the mean of the Rooms column

The mean of the Rooms column is calculated using the "mean" function and is rounded to the nearest integer using "round" function.

2.3.9 Replacing the NA values with mean

```
# Replace NA values in "Rooms" column with the mean
Filtered_dataset$Rooms[is.na(Filtered_dataset$Rooms)] <- mean_rooms
```

Figure 50-2.3.9: Replacing null value with mean

The NA values are replaced with the mean found in 2.3.8.

2.3.10 Checking the new 1st and 3rd quartile value

```
# Check the new 1st and 3rd Quartile value
summary(Filtered_dataset$Rooms)
```

```
> summary(Filtered_dataset$Rooms)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1.00    3.00    4.00    3.68    4.00   18.00
```

Figure 51-2.3.10.1: Checking the new 1st and 3rd quartile value

The result shows 1st and 3rd quartile remain as 3 and 4 respectively.

2.3.11 Replacing the upper outliers with the 3rd quartile

```
# Replace the upper outliers in the Rooms column with the 3rd quartile of the "Rooms" column
Filtered_dataset$Rooms[Filtered_dataset$Rooms >= MaxOut_Rooms] <- 4
```

Figure 52-2.3.11.1: Replacing upper outliers

The upper outliers are replaced using the 3rd quartile value, which is 4.

2.3.12 Replacing lower outliers which is not Studio into 1st quartile and check final result

```
# Replace lower outliers where is not a Studio house with the first quartile
Filtered_dataset$Rooms[Filtered_dataset$Rooms <= MinOut_Rooms & Filtered_dataset$IsStudio == FALSE] <- 3
```

Figure 53-2.3.12.1: Replacing lower outliers

The lower outliers are also replaced with the 1st quartile value but the Studio house are excused because we assume that Studio houses have only one room.

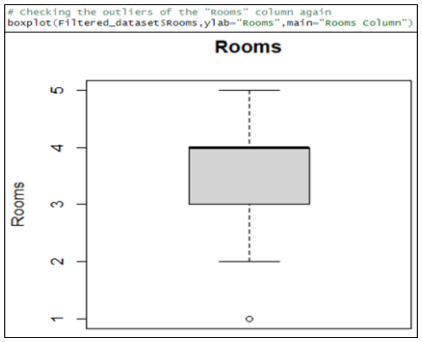


Figure 54-2.3.12.2: Box Plot of Rooms after validation

There are only some lower outliers which belong to the Studio houses left and the rest of the outliers are solved.

2.3.13 Calculating the maximum and minimum outliers of the Size column

```
# Calculating the maximum and minimum outliers of the "Size" column
IQR_Size <- 2378-1018
MaxOut_Size <- 2378+(1.5*IQR_Size)
MinOut_Size <- 1018-(1.5*IQR_Size) #Since that the MinOut_Size is negative, means that there are no low outliers
```

Figure 55-2.3.13.1: Calculating the maximum and minimum outliers of Size

The maximum and minimum outliers in the Size column are calculated using the same formula as stated in 2.3.3 and 2.3.7 and the results are as below.

```
> IQR_Size
[1] 1360
```

```
> MaxOut_Size
[1] 4418
```

```
> MinOut_Size
[1] -1022
```

Figure 56-2.3.13.2: Result of minimum and maximum outliers

The minimum outlier is a negative value, the lower outliers are ignored.

2.3.14 Replacing values greater than or equal to the maximum outliers with 3rd quartile

```
# Replace values greater than or equal to the maximum outliers with the 3rd quartile
Filtered_dataset$Size[ Filtered_dataset$Size >= MaxOut_Size] <- 2378
```

Figure 57-2.3.14.1: Replacing maximum outliers of Size

The upper outliers of Size column are replaced with the 3rd quartile before the mean are calculated.

2.3.15 Calculating the mean of Size column

```
# Calculate the mean of the Size column
mean_size <- mean(Filtered_dataset$Size,na.rm=TRUE)
```

```
> mean_size
[1] 1671.466
```

Figure 58-2.3.15.1: Mean of Size

Using the “mean” function, the mean f Size is 1671.466.

2.3.16 Replacing NA values in Size column with mean and checking the final result

```
# Replace NA values in "Size" column with the mean
Filtered_dataset$Size[is.na(Filtered_dataset$Size)] <- mean_size
```

Figure 59-2.3.16.1: Replace null values in Size

The NA values in the Size column are then replaced with the mean found in 2.3.15.

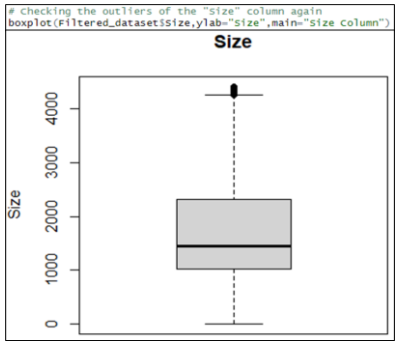


Figure 60-2.3.16.2: Box Plot of Size after validation

The outliers in the Size column are solved successfully.

2.3.17 Find the mode of the Type column

```
# Find the most frequently appearing character value in the Type column
most_frequent_type <- names(which.max(table(Filtered_dataset$Type)))
```

```
> most_frequent_type
[1] "Built-up"
```

Figure 61-2.3.17.1: Mode of Type Column

The mode of the Type column is determined using a series of functions from the “DescTools” library. First of all, a frequency table of the Type column is created using the “table” function. Then, the “which.max” function finds

the index of the maximum value in the frequency table and then the "names" function retrieves the respective name of the index returned by the "which.max" function. The mode is "Built-up".

2.3.18 Replacing the NA values in Type column with the mode

```
# Replacing the NA values in the Type column into the most_frequent_type
Filtered_dataset$Type[is.na(Filtered_dataset$Type)] <- most_frequent_type
```

Figure 62-2.3.18.1: Replacing null in Type

Then, the NA values in the Type column are replaced with the mode found in 2.3.17.

2.3.19 Find the mode of the Furnishing column

```
# Find the most frequently appearing character value in the Furnishing column
most_frequent_furnish <- names(which.max(table(Filtered_dataset$Furnishing)))
```

```
> most_frequent_furnish
[1] "Partly Furnished"
```

Figure 63-2.3.19.1: Mode of Furnishing Column

The mode of the Furnishing column is determined using the same way as stated in 2.3.17. The mode is "Partly Furnished".

2.3.20 Replacing empty and unknown value in Type column into the mode.

```
# Replacing the NA values and Unknown value in the Furnishing column into the most_frequent_furnish
Filtered_dataset$Furnishing[is.na(Filtered_dataset$Furnishing) | Filtered_dataset$Furnishing == "Unknown"] <- most_frequent_furnish
```

Figure 64-2.3.20.1: Replacing empty and unknown value in Type column

The NA and unknown values in the Furnishing column are replaced with the mode found in 2.3.19.

2.3.21 Remove duplicated data.

```
#Remove duplicated data
Filtered_dataset <- unique(Filtered_dataset)
```

Figure 65-2.3.21.1: Remove duplicated data

After everything is settled, the duplicated data is removed using the "unique" function.

2.3.22 Changing column names of the dataset

```
# Define new column names
new_column_names <- c("Property_Location", "Property_Price", "Property_Rooms", "Built_Area","Property_Size", "Property_Furnishing")
# Change column names in the dataset
names(Filtered_dataset)[names(Filtered_dataset) %in% c("Location", "Price", "Rooms","Type", "Size", "Furnishing")] <- new_column_names
```

Property_Location	Property_Price	Property_Rooms	IsStudio	Built_Area	Property_Size	Property_Furnishing

Figure 66-2.3.22.1: Changing column names

The column names are then changed to prevent plagiarism issue.

3.0 Data Analysis

3.1 Objective 1: To investigate the impact of Built_Area on the Price of the house in KLCC (Ooi Chong Ming TP072667)

3.1.1 Analysis 1-1: What is the overall relationship between Built Area and Price in the whole dataset and KLCC? – T-test

```
## Analysis 1-1 What is the overall relationship between Built Area and Price in the
#           whole dataset and KLCC
# Conducting a two-sided t-test for the whole dataset
whole_dataset <- Filtered_dataset
t.test(Property_Price~Built_Area,data=whole_dataset)

# Create a boxplot with mean for the whole properties
ggplot(whole_dataset, aes(x = Built_Area, y = Property_Price, fill = Built_Area)) +
  geom_boxplot() +
  stat_summary(fun = mean, geom = "point", shape = 18, size = 3, color = "black") +
  labs(x = "Built Area",
       y = "Property Price",
       title = "Box Plot of Property Price by Built Area in the whole dataset") +
  theme(
    panel.background = element_rect(fill = "lightgray", color = "black", size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    axis.line = element_line(color = "black"),
    legend.position = "top"
  )

# Create subset for KLCC properties
Klcc <- subset(whole_dataset,Property_Location == "Klcc")

# Conducting a two-sided t-test for the properties in KLCC
t.test(Property_Price~Built_Area,data=Klcc)

# Create a boxplot with mean for properties in KLCC
ggplot(Klcc, aes(x = Built_Area, y = Property_Price, fill = Built_Area)) +
  geom_boxplot() +
  stat_summary(fun = mean, geom = "point", shape = 18, size = 3, color = "black") +
  labs(x = "Built Area",
       y = "Property Price",
       title = "Box Plot of Property Price by Built Area in KLCC") +
  theme(
    panel.background = element_rect(fill = "lightgray", color = "black", size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    axis.line = element_line(color = "black"),
    legend.position = "top"
  )
```

Whole dataset

```
data:  Property_Price by Built_Area
t = -34.781, df = 14459, p-value < 2.2e-16
alternative hypothesis: true difference in means between group Built-up and group Land area is not equal to 0
95 percent confidence interval:
 -390353.8 -348702.8
sample estimates:
 mean in group Built-up mean in group Land area
                1098791                 1468320
```

KLCC

```
data:  Property_Price by Built_Area
t = 3.0439, df = 202.48, p-value = 0.002645
alternative hypothesis: true difference in means between group Built-up and group Land area is not equal to 0
95 percent confidence interval:
  67384.76 315236.19
sample estimates:
 mean in group Built-up mean in group Land area
                1799033                 1607723
```

Figure 67-3.1.1.1 t-test result for Built_Area against Price in whole dataset and KLCC

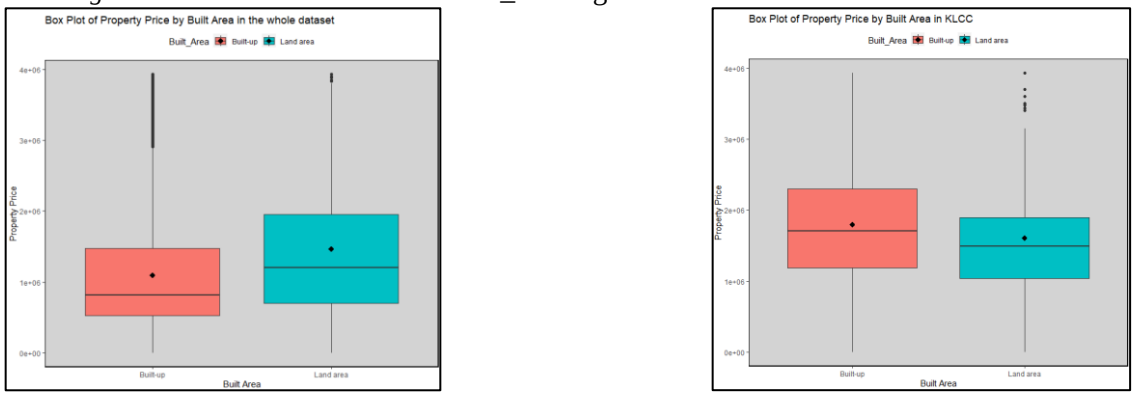


Figure 68-3.1.1.2 BoxPlot between Built_Area against Price in whole dataset and KLCC

To start off, a t-test is conducted between Built_Area and Property_Price for all properties in KL and properties in KLCC only for comparison purpose. As shown in Figure 3.1.1.1, the p-value for the whole dataset is shown as "<2.2e-16", which is extremely small. This indicates a highly significant difference in mean of the Property_Price between Built-up and Land Area properties, which leads to a conclusion that Built_Area is statistically significant to the Price. The same goes to the properties in KLCC, which also shows a low p-value of 0.002645. While it is not as small as the whole dataset, it is still significant as it is less than the conventional significance level of 0.05. This also indicates that the null hypothesis that the difference in mean is 0 is rejected. Based on the confidence interval, the mean of Property_Price for Land Area properties tends to be higher than Built-up properties in the whole dataset. However, it seems the exact opposite for KLCC properties where the mean of Built-up properties tends to be higher than Land Area. To conclude, the Built_Area does affect the Property_Price, and this influence may vary depending on the location. Figure 3.1.1.2 shows the boxplot with mean for both t-test as a visualization to show the difference between means.

3.1.2 Analysis 1-2: What is the distribution of Built_Area in the whole dataset and KLCC?

```
## Analysis 1-2: What is the difference in  distribution of Built_Area in the whole dataset and KLCC
# Check the distribution of the "Built_Area" column in the whole dataset and Klcc
table(whole_dataset$Built_Area)
table(Klcc$Built_Area)

# Combine the dataset and create a new column to indicate the dataset
whole_dataset$Dataset <- "Whole_dataset"
Klcc$Dataset <- "Klcc"
combined_dataset <- rbind(whole_dataset, Klcc)

# Visualization using clustered bar chart
ggplot(combined_dataset, aes(x = Built_Area, fill = Dataset)) +
  geom_bar(position = "dodge") +
  geom_text(stat = 'count',
            aes(label = after_stat(count)),
            position = position_dodge(width = 0.9), vjust = -0.5) +
  labs(x = "Built Area",
       y = "Count",
       title = "Distribution of Built Area: Whole Dataset vs KLCC Dataset") +
  scale_fill_manual(values = c("magenta", "red"),
                    name = "Dataset", labels = c("Klcc Dataset", "Filtered Dataset")) +
  theme(
    panel.background = element_rect(fill = "lightgray", color = "black", size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    axis.line = element_line(color = "black"),
    legend.position = "top"
  )
```

```
> table(whole_dataset$Built_Area)

 Built-up Land area
    28607       9610
> table(Klcc$Built_Area)

 Built-up Land area
     3439        184
>
```

Figure 69-3.1.2.1 Distributions of Built_Area in the whole dataset and KLCC

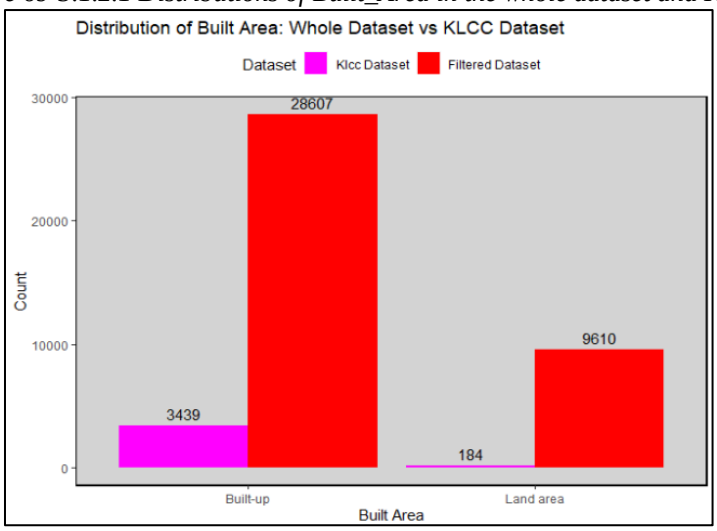


Figure 70-3.1.2.2 Clustered Bar Chart to compare the distributions

As shown in Figure 3.1.2.1, “table” functions are used to determine the overall distribution of Built_Area in both datasets. The distributions are further visualized using a clustered bar chart as shown in Figure 3.1.2.2. The result shows that there are a total of 28,607 Built-up properties and 3,439 properties of it are from KLCC. In the other hand, there are a total of 9,610 Land Area properties and only 184 of it are from KLCC. As a comparison, Built-up properties in KLCC covers a total of 12.02% from the whole dataset while Land Area properties in KLCC covers only 1.91% of the whole dataset. There are a few conclusions I can draw in this analysis. First of all, in the whole dataset or KLCC dataset, the Built-up properties are more prevalent compared to Land Area properties. When comparing percentage, it can also be said that from the whole dataset, the percentage of Built-up properties that is in KLCC is still higher than that of Land Area properties. This means that despite the lower number of Land Area properties in the whole dataset, the number of Built-up properties is still majority when we only consider properties in KLCC. This might mean that we can consider only Built-up properties for the rest of the analysis. But I haven’t made a conclusion yet as only one of the dependent variables, which is Location, is included and the Price hasn’t been taken into consideration yet.

3.1.3 Analysis 1-3: What is the relationship between price category (>1M or <=1M) and Built_Area in KLCC properties? – Pearson's Chi-square Test

```
## Analysis 1-3: What is the relationship between price category (>1M or <=1M) and
#                Built_Area in KLCC properties
# Create a new column of Price Categories (>1M and <=1M)
klccPrice_Category <- ifelse(klccProperty_Price > 1000000, ">1M", "<=1M")

# Chi-square goodness of fit test for the Price_Category
chisq.test(table(klccPrice_Category))

# Chi-square test of independence between Built_Area and Price_Category in KLCC
chisq.test(table(klccPrice_Category, klccBuilt_Area))

# visualizing the result
ggplot(klcc,aes(x=Price_Category,fill=Built_Area))+
  geom_bar(position = "dodge")+
  geom_text(stat = 'count',
            aes(label = after_stat(count)),
            position = position_dodge(width = 0.9),
            vjust = -0.5) +
  labs(x = "Price Category",
       y = "Count",
       title = "Distribution of Built Area against Price_Category: >1M vs <=1M") +
  scale_fill_manual(values = c("blue", "yellow"),
                    name = "Dataset",
                    labels = c("Built-up", "Land Area")) +
  theme(
    panel.background = element_rect(fill = "lightgray", color = black, size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    axis.line = element_line(color = black),
    legend.position = "top"
  )
```

Goodness of Fit test

```
        Chi-squared test for given probabilities

data:  table(Klcc$Price_Category)
X-squared = 1486.9, df = 1, p-value < 2.2e-16
```

Test of Independence

```
        Pearson's Chi-squared test with Yates' continuity correction

data:  table(Klcc$Price_Category, Klcc$Built_Area)
X-squared = 6.0092, df = 1, p-value = 0.01423
```

Figure 71-3.1.3.1 Chi-square Test between Price_Category and Built_Area in KLCC

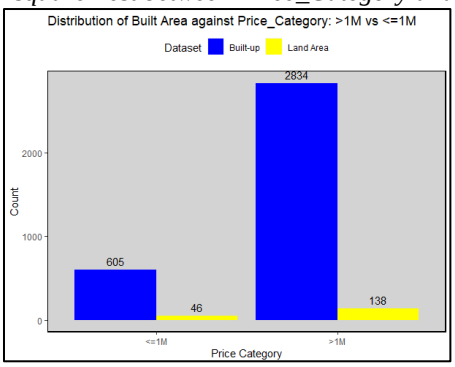


Figure 72-3.1.3.2 Chi-square Test between Price_Category and Built_Area in KLCC

A chi-square test is conducted between Price_Category and Built_Area to determine their relationship. First of all, a new column called "Price_Category" is created to separate them into properties of greater than 1 million than or equal to 1 million. A chi-square goodness of fit test with 95% significance level is then conducted on the Price_Category with the following hypothesis:

H_0: Proportions of properties in KLCC that are >1M and <=1M are equal

H_1: Proportions of properties in KLCC that are >1M and <=1M are not equal

As shown in Figure 3.1.3.1, the result p-value of the goodness of fit test is extremely small (<2.2e-16) which means that the probability of null hypothesis is too low that we can reject the null hypothesis (H_0) and accept the alternative hypothesis (H_1). Then, a test of independence with 95% significance level is conducted between Price_Category and Built_Area to determine their dependency.

H_0: Built_Area and Price_Category are independent

H_1: Built_Area and Price_Category are not independent

As shown in Figure 3.1.3.1, the result p-value for the test of independence is 0.01423 which is smaller than the conventional significance level of 0.05. Thus, the null hypothesis is also rejected, and the alternative hypothesis is accepted. As a conclusion, the Price_Category and Built_Area are dependent of each other and the change in one will affect the other.

3.1.4 Analysis 1-4: What is the distribution of Built_Area in KLCC properties with price greater than 1M?

```
## Analysis 1-4: What is the distribution of Built_Area in KLCC properties with price greater than 1M
# Creating a subset for KLCC properties with price greater than RM 1,000,000
More_than_1M <- subset(Klcc,Property_Price > 1000000)

# Check the distribution of the "Built_Area" column
table(More_than_1M$Built_Area) # 2834 Built-up, 138 Land Area

# Visualize the proportion of Built_Area among the properties in KLCC with price greater than 1M
data = table(More_than_1M$Built_Area)
percentages <- sprintf("%.2f%%",prop.table(data) * 100)
par(cex=0.8)
par(bg="lightgray")
pie3D(data,
    radius = 1,
    height = 0.1,
    theta = 0.7,
    col = c("lightblue","yellow"),
    border = "black",
    main ="Distribution of Built Area among KLCC properties with price >1M",
    labels = paste(names(data),"\n","Freq : ",data," (",percentages,")",sep=""),
    labelcex = 1,
    labelcol = "Red")

# Creating new subset with only Built-up properties in KLCC with price greater than RM 1,000,000
Built_up_Klcc_greater_1M <- subset(More_than_1M,Built_Area == "Built-up")
```

Distribution of Built_Area

```
Built-up Land area
    2834       138
```

Figure 73-3.1.4.1 Distribution of Built_Area in KLCC properties with price greater than 1M

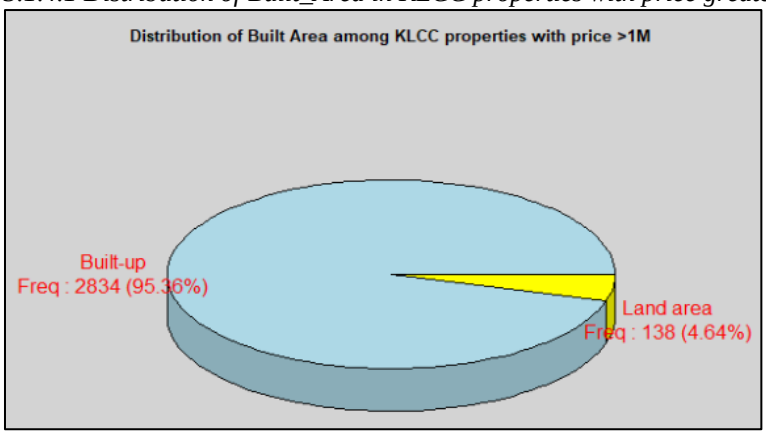


Figure 74-3.1.4.2 Pie chart for the Distribution of Built_Area in KLCC with price greater than 1M

First of all, a subset with only properties in KLCC and price greater than 1 million are created. Then, the total distribution of Built_Area in KLCC properties with price greater than 1 million is first determined using the "table" function as shown in Figure 3.1.4.1. A 3D pie chart is also plotted as shown in Figure 3.1.4.2 to show the distributions of the Built_Area in KLCC with price greater than 1 million. It is shown that among the properties in KLCC with price greater than 1 million, there are 2 834 Built-up properties which holds 95.36% of the properties while the rest 138 or 4.64% of the properties belongs to Land Area properties. Thus, it is clear that the Built-up properties holds the majority compared to Land Area properties. According to analysis 3.1.1 and 3.1.3, it is also shown that properties in KLCC tend to be Built-up instead of Land Area. Thus, with all these supporting analyses, it can be said that Built-up properties have higher percentage to fulfil the dependent variable of KLCC and price greater than 1 million. Therefore, a conclusion that Land Area properties don't affect my objectives much can be made and for the rest of the analysis, the Land Area properties can be ignored. After this conclusion is made, a new dataset with only Built-up properties in KLCC with price greater than 1 million is created using the "subset" function for the use of rest of the analysis.

3.1.5 Analysis 1-5: What are the relationships between Built_Area, Property_Furnishing and Property_Price in KLCC – ANOVA test

```
## Analysis 1-5: What are the relationships between Built_Area, Property_Furnishing
#               and Property_Price in KLCC
# Visualize using boxplot
ggplot(klcc, aes(x = Property_Furnishing, y = Property_Price, fill = Built_Area)) +
  geom_boxplot() +
  scale_fill_manual(values = c("blue", "red"), name = "Property Furnishing") +
  labs(x = "Furnishing Status",
       y = "Property Price",
       title = "Box Plot of Property Price by Built Area and Property Furnishing in KLCC") +
  theme(
    panel.background = element_rect(fill = "lightgray", color = "black", size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    axis.line = element_line(color = "black"),
    legend.position = "top"
  )
#Visualize using Histograms
ggplot(klcc, aes(x = Property_Price, fill = Built_Area)) +
  geom_histogram(binwidth = 100000, position = "dodge") +
  scale_fill_manual(values = c("blue", "red"), name = "Built_Area") +
  labs(x = "Property Price",
       y = "Count",
       title = "Histogram of Property Price by Property Furnishing in KLCC") +
  facet_wrap(~Property_Furnishing, ncol = 1) +
  theme(
    panel.background = element_rect(fill = "lightgray", color = "black", size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    axis.line = element_line(color = "black"),
    legend.position = "top"
  )
# Perform ANOVA test
anova <- aov(Property_Price ~ Built_Area*Property_Furnishing,data=klcc)
summary(anova)
# Check dependency between Furnishing Status
TukeyHSD(anova)
```

ANOVA test

```
> anova <- aov(Property_Price ~ Built_Area*Property_Furnishing,data=klcc)
> summary(anova)
                       Df    Sum Sq   Mean Sq F value  Pr(>F)
Built_Area              1 6.392e+12 6.392e+12   10.01 0.00157 **
Property_Furnishing     2 1.220e+14 6.102e+13   95.54 < 2e-16 ***
Residuals            3619 2.311e+15 6.387e+11
---
signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Tukey's test

```
> TukeyHSD(anova)
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = Property_Price ~ Built_Area + Property_Furnishing, data = klcc)

$Built_Area
                          diff     lwr       upr     p adj
Land area-Built-up -191310.5 -309874 -72746.93 0.0015712

$Property_Furnishing
                                      diff       lwr      upr     p adj
Partly Furnished-Fully Furnished 369545.22  305845.9 433244.6 0.0000000
Unfurnished-Fully Furnished      348058.70  170411.0 525706.4 0.0000134
Unfurnished-Partly Furnished     -21486.52 -200403.7 157430.6 0.9572370
```

Figure 75-3.1.5.1 ANOVA test and Tukey's test and their result

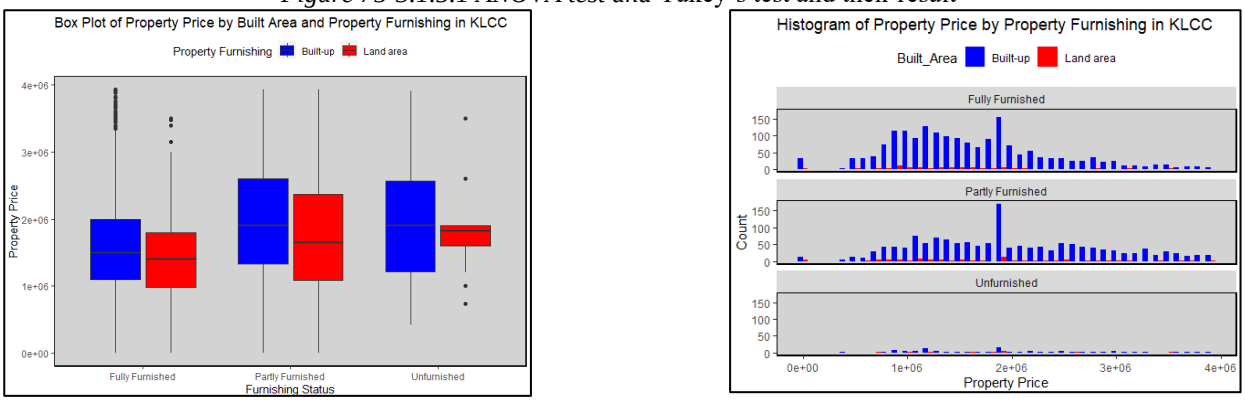


Figure 76-3.1.5.2 Boxplot and Histogram for visualization of ANOVA test

To start off the ANOVA test, three assumptions had to be made, which are the observations, in this case is Built_Area, have to be independent. These assumptions are fulfilled as in this analysis, Built_Area is an independent variable. Next assumption is that the observations have to be approximately normally distributed. This can be proved by the histogram in Figure 3.1.5.2 which shows that other than Unfurnished properties, other histograms are approximately normally distributed. This can also be shown in the whiskers of the boxplot in Figure 3.1.5.2. The last assumption is that the variance is approximately equal within Built-up and Land Area properties. This can be shown in the box plot where the inter-quartile range between the box plot are approximately equal. After all these assumptions are fulfilled, the ANOVA test is run between the Built_Area, Property_Furnishing and Property_Price. As the result in Figure 3.1.5.1 showed that the p value for both Built_Area and Property_Furnishing is extremely low (0.00157 and <2e-16 respectively). This means that they are statistically significant to the Property_Price. The Tukey's test is then conducted to determine the dependency of the columns. The low p-value shows that there is great dependency between Built-up and Land Area (0.0015), Partly Furnished and Fully Furnished (0.00) , and between Unfurnished and Fully Furnished (0.00001). However, the high p value (0.9572) shows that there is no dependency between Unfurnished and Partly Furnished properties.

3.1.6 Analysis 1-6: What are the proportions of Furnishing Status among the Built-up properties in KLCC with price >1M

```
## Analysis 1-6: What are the proportions of Furnishing Status among the
#                 Built-up properties in KLCC with price >1M
# Visualizing the proportion using a 3D Piechart
data_Furnishing = table(Built_up_Klcc_greater_1M$Property_Furnishing)
percentage_Furnishing <- sprintf("%.2f%%",prop.table(data_Furnishing) * 100)
par(cex=0.7)
par(bg="lightgray")
pie3D(data_Furnishing,
      explode = .1,
      radius = 1,
      height = 0.1,
      theta = 0.7,
      col = c("blue", "green","orange"),
      border = "black" ,
      main = "The proportion of Furnishing Status in Built-up properties\nin KLCC with price >1M",
      labels = paste(names(data_Furnishing),"\n","Freq : ",
                     data_Furnishing,
                     " (",percentage_Furnishing,")",
                     sep=""),
      labelcex = 1,
      labelcol = "black")
```

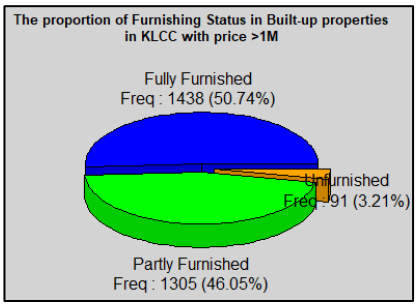


Figure 77-3.1.6.1 Distribution *of Furnishing Status* in KLCC with price >1M

Next, the proportions of the Property_Furnishing column among the Built-up properties in KLCC with price greater than 1 million are determined using a 3D pie chart as shown in Figure 3.1.6.1. The dataset I used in this analysis is the product dataset of analysis 3.1.4. As shown in the 3D pie chart, the proportions of Fully Furnished and Partly Furnished properties are very similar (50.74% and 46.05% respectively). Comparingly, the Unfurnished properties only hold a very small portion (3.21%) of the dataset. According to the ANOVA test in analysis 3.1.5, it is also shown that the Unfurnished properties actually doesn't fulfil all the assumptions for an ANOVA test and the Tukey's test also shows that there is no dependency between Unfurnished and Partly Furnished properties. So, I decided to exclude the Unfurnished properties for the rest of my analysis as it only holds a small portion of the dataset and its dependency with other properties is not significant enough to be considered in the rest of my analysis. Thus, only Partly and Fully Furnished properties are taken into consideration for the rest of my analysis.

3.1.7 Analysis 1-7: What are the proportions of Rooms in Fully Furnished and Partly Furnished Built-up properties in KLCC with price >1M

```
## Analysis 1-7: What are the proportions of Rooms in Fully Furnished and
#                 Partly Furnished Built-up properties in KLCC with price >1M
# Forming dataset with only Fully Furnished and Partly Furnished built-up properties in KLCC with price >1M
data_Rooms <- subset(Built_up_Klcc_greater_1M,Property_Furnishing %in%
                       c("Fully Furnished","Partly Furnished"))
# Subset the data into rooms category
data_Rooms$RoomStatus <- ifelse(data_Rooms$Property_Rooms >3, ">3","<=3")
# Calculate percentages for each room status within Furnishing Status groups
data_Rooms <- data_Rooms %>%
  group_by(Property_Furnishing, RoomStatus) %>%
  summarise(count = n()) %>%
  mutate(percentage = count / sum(count) * 100)
# Create the clustered bar chart
ggplot(data_Rooms, aes(x = Property_Furnishing, y = count, fill = RoomStatus)) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_text(aes(label = paste0(count, " (", sprintf("%.2f%%", percentage), ")")),
            position = position_dodge(width = 0.9), vjust = -0.5, size = 3) +
  labs(x = "Furnishing Status",
       y = "Frequency",
       fill = "Room Status" ,
       title = "Distribution of different rooms category based on Furnishing Status") +
  scale_y_continuous() +
  theme(
    panel.background = element_rect(fill = "lightgray", color = "black", size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    axis.line = element_line(color = "black"),
    legend.position = "top"
  )
```

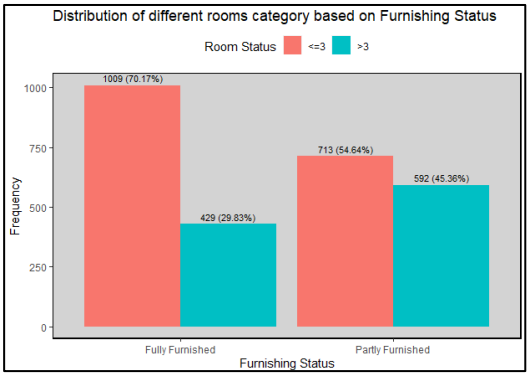


Figure 78-3.1.7.1 Distributions *of Rooms Category based on Furnishing Status*

Continuing the analysis, the distributions of different room categories (>3 or <=3) within Fully Furnished and Partly Furnished Built-up properties in KLCC with price greater than 1 million, are determined by plotting a clustered bar chart as shown in Figure 3.1.7.1. As shown in the clustered bar chart, for Fully Furnished Built-up properties, the distributions of properties with less than or equal to 3 rooms to more than 3 rooms are 1,009

(70.17%) and 429 (29.83%) respectively. It can be concluded that majority of Fully Furnished properties have less than or equal to 3 rooms. Thus, it can be said that Fully Furnished Built-up properties with less than or equal to 3 rooms have a higher percentage to achieve the dependent variable of KLCC and price greater than 1 million compared to properties with more than 3 rooms. In the other hand, for Partly Furnished Built-up properties, the distributions of properties with less than or equal to 3 rooms to more than 3 rooms are almost the same, which is 713 (54.64%) and 592 (45.36%) respectively. Thus, it can be concluded that the Rooms Category doesn't have any impact on Partly Furnished Built-up properties in KLCC. In another word, Rooms Category and Furnishing Status are independent under the Partly Furnished case. Therefore, both rooms category should be taken down for further analysis to determine which path to choose.

3.1.8 Analysis 1-8: What are the proportions of different categories of size in Fully Furnished Built-up properties in KLCC with less than or equal to 3 BHK and price >1M

```
## Analysis 1-8: What are the proportions of different categories of size in Fully Furnished Built-up
#               properties in KLCC with less than or equal to 3 BHK and price >1M

# Forming dataset with only Fully Furnished Built-up properties in KLCC with <3 BHK and price >1M
FF_MoreThan3Rooms <- subset(Built_up_Klcc_greater_1M,
                            Built_up_Klcc_greater_1M$Property_Rooms <=3 &
                            Built_up_Klcc_greater_1M$Property_Furnishing == "Fully Furnished")

# Add a new column to separate data into three room categories
FF_MoreThan3Rooms$SizeStatus <- ifelse(FF_MoreThan3Rooms$Property_Size < 1000, "< 1000",
                                       ifelse(FF_MoreThan3Rooms$Property_Size <= 1800,
                                              "1000-1800", "> 1800"))

# Visualize
data_FF_sizestatus = table(FF_MoreThan3Rooms$SizeStatus)
percentage_FF_SizeStatus <- sprintf("%.2f%%",prop.table(data_FF_sizestatus) * 100)
par(cex=0.7)
par(bg="lightgray")
pie3D(data_FF_sizestatus,
      explode = .1,
      radius = 1,
      height = 0.1,
      theta = 0.7,
      col = c("green", "orange","purple"),
      border = "black",
      main = "The proportion of Size Status in Fully Furnished Built-up properties
      \nin KLCC with price >1M and BHK<=3",
      labels = paste(names(data_FF_sizestatus),"\n","Freq : ",
                     data_FF_sizestatus," (",percentage_FF_SizeStatus,")",sep=""),
      labelcex = 1,
      labelcol = "black")
```

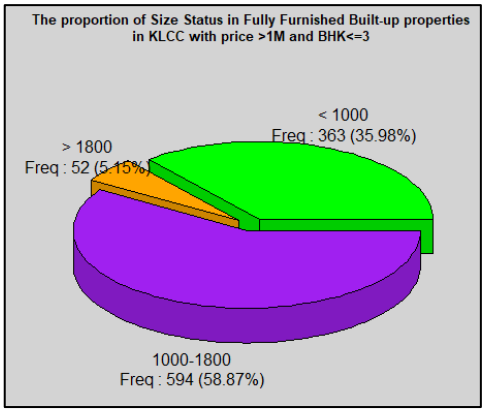


Figure 79-3.1.7.1 Proportions of Size Categories in Fully Furnished properties with <=3 rooms

Continuing with the Fully Furnished Built-up properties in KLCC with price greater than 1 million and lesser than or equal to 3 rooms, the distributions of Size Categories are investigated. Firstly, a subset with only Fully Furnished and lesser than or equal to 3 rooms is created. A new column called SizeStatus was then created to determine the categories of Sizes for each property. The sizes are separated into three main categories, which are 0-1000, 1000-1800 and more than 1800 square feet. After that, a 3D pie chart is plotted to visualize the distribution. As shown in Figure 3.1.7.1, under Fully Furnished Built-up properties with less than or equal to 3 rooms, properties with 1000-1800 square feet holds the majority (58.87%) when compared to other categories. In another way, it can be said that properties with 1000-1800 square feet have a higher chance to obtain Built-up Fully Furnished properties with lesser than or equal to 3 rooms and in KLCC with price greater than 1 million. Therefore, a conclusion can be made that Fully Furnished Built-up properties with less than or equal to 3 bedrooms and 1000-1800 square feet are more likely to have a price greater than 1 million in KLCC.

3.1.9 Analysis 1-9: What are the proportions of different categories of size in Partly Furnished Built-up properties in KLCC with price >1M and different Room category

```
## Analysis 1-9: What are the proportions of different categories of size in Partly Furnished Built-up
#                properties in KLCC with price >1M and different Room category
# Forming dataset with only Partly Furnished Built-up properties in KLCC with price >1M
PF <- subset(Built_up_klcc_greater_1M,Built_up_klcc_greater_1M$Property_Furnishing == "Partly Furnished")
# Add a new column to separate data into three room categories
PF$SizeStatus <- ifelse(PF$Property_Size < 1000, "< 1000",
                        ifelse(PF$Property_Size <= 1800, "1000-1800", "> 1800"))
# Add a new column to separate data into three room categories
PF$RoomStatus <- ifelse(PF$Property_Rooms >3, ">3","<=3")
# Calculate percentages for each Size status
PF <- PF %>%
  group_by(RoomStatus,SizeStatus)%>%
  summarise(count = n()) %>%
  mutate(percentage = count / sum(count) * 100)
# Create the clustered bar chart
ggplot(PF, aes(x = RoomStatus,y=count, fill = SizeStatus)) +
  geom_bar(stat="identity",position = "dodge") +
  geom_text(aes(label = paste0(count, " (", sprintf("%.2f%%", percentage), ")")),
            position = position_dodge(width = 0.9), vjust = -0.5, size = 3) +
  labs(x = "Property_Rooms",
       y = "Frequency",
       fill = "Size Category" ,
       title="Distribution of Partly Furnished properties based on Size and Rooms Category") +
  scale_y_continuous()+
  theme(
    panel.background = element_rect(fill = "lightgray", color = "black", size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    axis.line = element_line(color = "black"),
    legend.position = "top"
  )
```

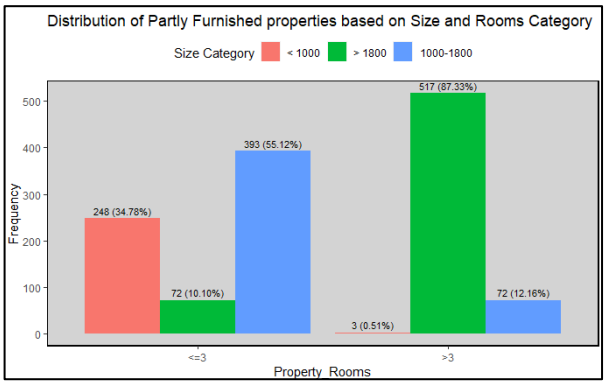


Figure 80-3.1.9.1 Distributions of Size Categories in Partly Furnished properties

Lastly, the distributions of size categories in Partly Furnished properties are determined using a clustered bar chart as shown in Figure 3.1.9.1. A subset with only Partly Furnished Built-up properties in KLCC with price greater than 1 million is first created. Then, two new columns called RoomStatus and SizeStatus are created to categorize the properties into their respective Size (>3 or <=3) and Room (0-1000,1000-1800,>1800) category. As shown in the clustered bar chart in Figure 3.1.9.1, for the category of less than or equal to 3 rooms, rooms with 1000-1800 square feet holds majority with a total of 393 properties which is 55.12% among all the Partly Furnished Built-up properties with less than or equal to 3 rooms. Thus, a conclusion can be made on this part that Partly Furnished Built-up properties with less than or equal to 3 rooms and 1000-1800 square feet are more likely to have a price greater than 1 million in KLCC compared to other size categories. On the other hand, for another room category of more than 3 rooms, rooms with >1800 square feet hold the majority with a total of 517 properties or 87.33% among all the Partly Furnished Built-up properties with more than 3 rooms. Based on this analysis, a conclusion can be made that Partly Furnished Built-up properties with more than 3 rooms and size greater than 1800 square feet are more likely to have a price greater than 1 million in KLCC.

3.1.10 Conclusion

After going through all the analysis for my objective, I came into 3 main conclusions as below:

i. 19.99% of Fully Furnished Built-up properties with less than or equal to 3 rooms and 1000 – 1800 square feet has a price greater than RM 1 million in KLCC.

ii. 17.4% of Partly Furnished Built-up properties with more than 3 rooms and size greater than 1800 square feet have a price greater than RM 1 million in KLCC.

iii. 13.22% of Partly Furnished Built-up properties with less than or equal to 3 rooms and 1000 – 1800 square feet have a price greater than RM 1 million in KLCC.

While comparing with our hypothesis, I came up with a conclusion that our hypothesis is incorrect in the size and room category variable.

3.1.11 Extra Features

Library

- stringr : used for handling and manipulating character strings.
- plotrix : provides various plotting functions for creating complex visualizations.
- tidyverse : collection of R packages designed for data science.
- hmisc : comprehensive package that contains many functions useful for data analysis.
- gmodels : provides various tools for model fitting and statistical modeling.

3.1.1

- t.test() : run a paired t-test between one categorical variable and one quantitative variable.
- stat_summary : adds a summary statistic to the plot.
- fun to specify the function to compute the summary statistic.
- geom to specifies the type of geometric object to use for displaying the summary statistic.
- shape to control the shape of the point used to display the summary statistic.
- labs() : adds labels to the plot.
- theme() : customizes the non-data components of the plot.
- panel.background to define the background of the plot panel.
- panel.grid.major and panel.grid.minor to Control the appearance of major and minor grid lines.
- axis.line to control the appearance of axis lines.
- legend.position to Specifies the location of the legend on the plot.

3.1.2

- table() : computes a contingency table of the counts at each combination of factor levels.
- rbind() : combines vectors, matrices, or data frames by rows.
- geom_text : adds text annotations to the plot.
- stat to specify the type of statistic to compute.
- label to specifies the text to display.
- position to specify where to place the text.
- vjust to adjust the vertical justification of the text.
- after_stat() : Accesses computed statistics within geom_text()
- scale_fill_manual : Manually sets the colors for the fill aesthetic.
- values to specifies the colors
- name to give a name to the legend
- labels to provides custom labels for the legend entries.

3.1.3

- chisq.test() : performs a chi-square goodness-of-fit test and chi-square test of independence.

3.1.4

- sprint() : formats and prints data values with specified formatting.
- prop.table() : calculates the proportions of table entries.
- par() : sets graphical parameters.
- cex to set the character expansion factor.
- bg to set the background colour.
- pie3D() : creates a 3D pie chart.
- height to set the height of the pie chart.
- theta to set the angle of rotation in the pie chart.
- border to set colour of the border around the pie slices.
- labelcex to set the character expansion factor for the labels.
- labelcol to set the colour of the labels.

3.1.5

- geom_histogram : Adds a histogram layer to the ggplot object, creating a histogram.
- binwidth to specify the width of the bins.
- position to specify how bars should be positioned relative to each other.
- facet_wrap() : Divides the data into subsets and creates a separate plot for each subset.
- aov() : Performs an analysis of variance (ANOVA) test.
- TukeyHSD() : Performs Tukey's Honest Significant Difference (HSD) post-hoc test.

3.1.7

- group_by() : groups a dataframe by one or more variables.
- summarise() : calculates summary statistics for grouped data.
- mutate() : adds new variables to a dataframe or modifies existing ones.
- scale_y_continuous() : Modifies the y-axis to be continuous.

3.2 Objective 2: To determine the impact of furnishing status on property price in KLCC. (YEOH ZI QING BRYAN TP072717)

3.2.1 Analysis 2-1 What is the relationship between the furnishing status and property price?

T-test is a statistical hypothesis test used to test whether the difference between the response of two groups is statistically significant or not. I have used t-test to test the relationship between the furnishing status (Fully furnished, partly furnished and unfurnished) and the price (dependent variable.).

Testing:

```
# Conduct t-test
pairwise_test_results <- pairwise.t.test(Filtered_dataset$Property_Price,
                                         Filtered_dataset$Property_Furnishing)
pairwise_test_results
```

Figure 81- 3.2.1.1 Conduct T-test for furnishing status and price.

Objective:

Determine if there is a significant difference in the mean price between houses with different furnishing statuses.

Hypotheses:

- Null Hypothesis (H0): There is no significant difference in the mean price between houses with different furnishing statuses.
- Alternative Hypothesis (H1): There is a significant difference in the mean price between houses with different furnishing statuses.

Results:

Fully Furnished vs. Partly Furnished:	Fully Furnished vs. Unfurnished:	Partly Furnished vs. Unfurnished:
p-value = 0.28	p-value < 2e-16 (extremely small)	p-value < 2e-16 (extremely small)
Conclusion: Fail to reject the null hypothesis, suggesting no significant difference in mean prices between fully furnished and partly furnished houses.	Conclusion: Reject the null hypothesis, indicating a significant difference in mean prices between fully furnished and unfurnished houses.	Conclusion: Reject the null hypothesis, signifying a significant difference in mean prices between partly furnished and unfurnished houses.

Box Plot with Mean to perform visualization on the t-test.

```r
# Create a grouped box plot with mean points
ggplot(Filtered_dataset, aes(x = Property_Furnishing,
                             y = Property_Price,
                             fill = Property_Furnishing)) +
  geom_boxplot() +
  stat_summary(fun = mean,
               geom = "point",
               shape = 18,
               size = 3,
               color = "red",
               position = position_dodge(width = 0.75)) +
  labs(title = "Box Plot with Mean for Property Prices by Furnishing Category",
       x = "Furnishing Category",
       y = "Property Price (RM)") +
  theme_minimal()
```

Figure 82-3.2.1.2 Box Plot with Mean for Property Prices by Furnishing Category

Plot Description: A grouped box plot was constructed to show the range of property prices based on different furnishing status. Each box plot represents a different furnishing category, with specific property prices presented within each one. Furthermore, red points represent the average price for each furniture type.

Observation:

- The box plot and mean points visually depict the relationship between property furnishing status and prices.
- As observed, there is a notable difference in the distribution of property prices between fully furnished and partly furnished categories.
- Outliers are observed in each furnishing category, suggesting variability in property prices within each group.

Conclusion:

The results indicate that the furnishing status has a considerable impact on the mean price. Specific pairwise comparisons found considerable differences, especially between fully furnished and unfurnished homes.

3.2.2 Analysis 2-2: ANOVA Testing to test is there a relationship between furnishing status and square feet category with price.

Objective: Determine if there are significant differences in property prices based on the factors of furnishing status, square feet category, and their interaction.

Testing:

```r
# Perform multiple-factor ANOVA
anova_result <- aov(Property_Price ~ Property_Furnishing * Square_Feet_Category,
                    data = Filtered_dataset)

# Summarize ANOVA results
summary(anova_result)
```

Figure 83-3.2.2.1 ANOVA Testing

Results:

```
                                      Df    Sum Sq   Mean Sq F value Pr(>F)
Property_Furnishing                    2 1.479e+14 7.394e+13   201.4 <2e-16 ***
Square_Feet_Category                   2 1.294e+16 6.472e+15 17630.6 <2e-16 ***
Property_Furnishing:Square_Feet_Category     4 6.108e+13 1.527e+13    41.6 <2e-16 ***
Residuals                          38208 1.403e+16 3.671e+11
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Figure 84-3.2.2.2 ANOVA Results

Hypotheses:

Null Hypotheses:

- There is no significant difference in property prices based on the furnishing status.
- There is no significant difference in property prices based on the square feet category.
- The interaction between property furnishing and square feet category has no significant effect on property prices.

Alternative Hypotheses:

- There is a significant difference in property prices based on the furnishing status.
- There is a significant difference in property prices based on the square feet category.
- The interaction between property furnishing and square feet category has a significant effect on property prices.

Property Furnishing:
Conclusion: Reject the null hypothesis.

Interpretation: Property costs vary significantly depending on the furnishing status. The low p-value ($p < 0.001$) suggests that the observed variations in means are unlikely to arise by coincidence.

Square Feet Category:
Conclusion: Reject the null hypothesis.

Interpretation: Property costs vary significantly based on square feet. The low p-value ($p < 0.001$) indicates that observed variations in means are unlikely to arise by coincidence.

Property Furnishing x Square Feet Category:
Conclusion: Reject the null hypothesis.
Interpretation: The connection between property furnishing and square feet category has a major impact on property pricing. The low p-value ($p < 0.001$) suggests that the observed variations in means are unlikely to arise by coincidence.

Visualize ANOVA Test using Violin Plot

```r
library(ggplot2)

# Extract the residuals for plotting
residuals <- anova_result$residuals

# Combine the residuals with the original dataset for plotting
df_for_plot <- cbind(Filtered_dataset, Residuals = residuals)

# Create a violin plot
ggplot(df_for_plot, aes(x = Property_Furnishing, y = Residuals, fill = Square_Feet_Category)) +
  geom_violin(trim = FALSE, scale = "width", width = 0.8, alpha = 0.7) +
  scale_fill_brewer(palette = "Set3") +
  labs(title = "Violin Plot of Residuals by Furnishing and Square Feet Category",
       x = "Property Furnishing",
       y = "Residuals") +
  theme_minimal() +
  theme(legend.position = "bottom", legend.title = element_blank())
```

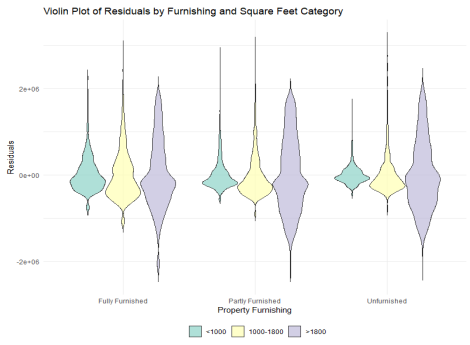


Figure 85-3.2.2.3 Violin Plot of Residuals by Furnishing Status and Square Feet Category.

The violin plot of residuals is intended to visualize the distribution of residuals from the ANOVA model across different levels of property furnishing and square feet category. Residuals indicate the disparities between observed and anticipated values, which provide information about the model's performance.

Conclusion:

The violin plot of residuals summarizes the ANOVA model's ability to accurately capture variations in property prices based on furnishing status and square feet. It aids in the identification of specific patterns, outliers, or places where the model may require adjustment, resulting in a more comprehensive understanding of the model's strengths and limits.

3.2.3 Analysis 2-3: Furnishing Status Distribution

Objective: To visualize the distribution of furnishing statuses in the entire cleaned and pre-processed dataset.

Methodology: A bar chart is shown using the “ggplot2” and “ggthemes” library to represent the count of each furnishing status in the whole dataset.

```
# Bar Plot for Furnishing Status Distribution
library(ggplot2)
library(ggthemes)

# Set the color for each category
colors <- c("#0073C2", "#EFC000", "#868686")

# Calculate percentages
furnishing_counts <- table(Filtered_dataset$Property_Furnishing)
furnishing_percentages <- prop.table(furnishing_counts) * 100

# Create a data frame for labels
label_data <- data.frame(
  Property_Furnishing = names(furnishing_counts),
  Count = as.numeric(furnishing_counts),
  Percentage = as.numeric(furnishing_percentages)
)

ggplot(Filtered_dataset, aes(x = Property_Furnishing, fill = Property_Furnishing)) +
  geom_bar(color = "black", size = 0.3, width = 0.7) +
  geom_text(data = label_data,
            aes(label = paste0(format(round(Percentage, 2), nsmall = 2), "%"), y = Count + 100),
            position = position_stack(vjust = 0.5)) +  # Adjust the position for better visibility
  scale_fill_manual(values = colors) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        legend.position = "none") +
  labs(title = "Distribution of Furnishing Status",
       x = "Furnishing Status",
       y = "Count") +
  ggtitle("Distribution of Furnishing Status") +
  theme_economist()
```

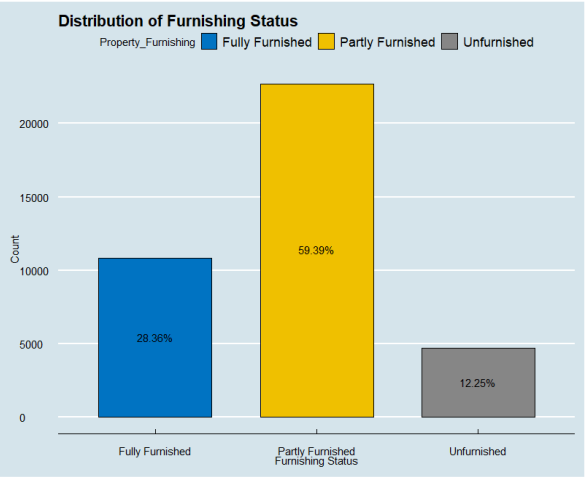


Figure 86-3.2.3.1 Bar Chart of Furnishing Status in whole dataset.

As we can see from the bar chart above, it clearly shows that there are more partly furnished houses in our dataset, which have 54.39% out of the whole dataset after we have done the cleaning and pre-processing on the dataset. Fully furnished houses come in second place with 28.36% and unfurnished houses only have 12.25%.

For plotting the bar chart shown in figure 3.2.3.1, I have used library ggplot2, furnishing status have been mapped to both the x-axis and the fill colour using aes() function. Then I adjusted the bar border colour, size and width of the bar as shown in the code figure. Geom_text will allow me to add text labels to the plot and I have labelled y-axis with “Count” using the aes() function again. Furthermore, paste0(format(round(Percentage, 2), nsmall = 2), "%") formats percentage labels to 2 decimal places. I have adjusts the vertical position of the label

using “position_stack(vjust = 0.5)” for better visibility. scale_fill_manual() manually sets the fill colors using the colors vector which have been set above with 3 different colours.

Theme_minimal() function have been used as a extra features for me to set a minimalistic theme to the bar chart. I have used theme() for additional theme customization as well such as adjusting the angle of x-axis labels and legend. I have used ggtitle() and labs() to add title and labels to the plots to let the plot looks more easy to understand. Lastly, I have used theme_economist which is a custom theme and further modify the appearance of the plot to become more minimalistic manner.

3.2.4 Analysis 2-4: Furnishing Status Distribution in KLCC location.

Objective: To visualize the distribution of furnishing statuses in the KLCC location.

Methodology: A bar chart is shown using the “plotrix” library to represent the count of each furnishing status in the KLCC location.

```
#Analysis 2: Furnishing status distribution in KLCC.
library(plotrix)

# Filter data for KLCC location
klcc_dataset <- Filtered_dataset[Filtered_dataset$Property_Location == "Klcc", ]

# Get values and labels
values <- table(klcc_dataset$Property_Furnishing)
labels <- names(values)

# Calculate percentages
percentages <- sprintf("%.2f%%", (values / sum(values)) * 100)

# Set up plotting parameters
par(cex = 0.7)

# Create 3D explode pie chart
pie3D(values,
      explode = 0.1,
      col = c("lightblue", "lightgreen", "lightyellow"),
      border = "black",
      main = "Furnishing Status Distribution in KLCC",
      labels = paste(labels, "\n", percentages, sep = ""),
      labelcex = 1,
      labelcol = "black")
```

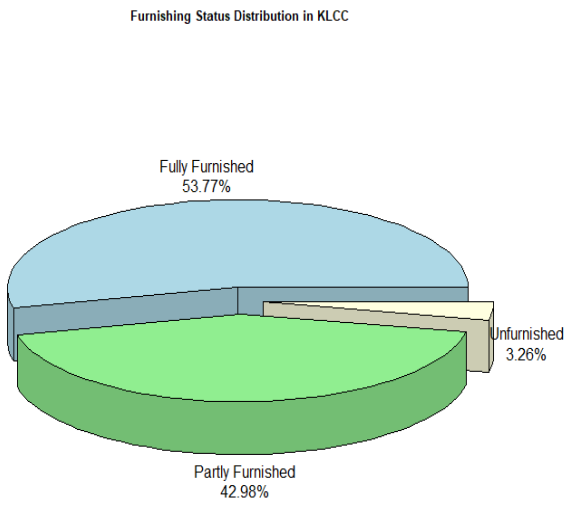


Figure 87-3.2.4.1 Explode 3D Pie Chart of Furnishing Status in KLCC.

From the Explode 3D Pie Chart above, we can clearly see that there are more fully furnished houses in KLCC which stands out 53.77% out of the whole dataset in KLCC location. Partly furnished houses have 42.98% while Unfurnished houses have the least count, which is 3.26% only. This clearly shows that in KLCC location, there are more fully and partly furnished houses which have a similar percentage.

I have used plotrix library in plotting this 3D explode pie chart. This library provides me with the pie3D function. I have created another subset which only includes the dataset in KLCC location only. Then, I proceed to use the table() function to count occurrences of each unique furnishing status in the “klcc_dataset”. Meanwhile “values” contains the count of each furnishing status and “labels” contains the names (categories) of the furnishing statuses. Furthermore, I have used sprintf() to format and output character strings in a flexible way to show the percentage of each furnishing status category in KLCC. “par(cex = 0.7)” will help me to set the character expansion factor to 0.7 which allows me to control the size of the text in 3D pie chart. Last but not least, pie3D() function has been introduced. "explode = 0.1” will let the pie chart have a slightly explode effects. I have set the

fill and border colour using “col” and “border” to the colour I wanted. Then, I added the main title and other labels into the 3D pie chart using “main” and “labels , labelcex, labelcol”.

3.2.5 Analysis 2-5: Furnishing Status Distribution in KLCC location with price greater than 1 million.

Objective: To visualize the distribution of furnishing statuses in the KLCC location with price greater than 1 million.

Methodology: A stacked bar chart is shown using the “ggplot2” and “dplyr” library to represent the count of each furnishing status in the KLCC location with price greater than 1 million.

```
#Import library that are needed
library(ggplot2)
library(dplyr)

# Filter data for KLCC location with price > 1 million
klcc_high_price <- Filtered_dataset[
  (Filtered_dataset$Property_Location == "Klcc") &
    (Filtered_dataset$Property_Price > 1000000), ]


klcc_high_price %>%
  group_by(Property_Furnishing) %>%
  summarize(Count = n()) %>%
  mutate(Percentage = scales::percent(Count / sum(Count), accuracy = 0.1)) %>%
  ggplot(aes(x = factor(1), y = Count, fill = Property_Furnishing)) +
  geom_bar(stat = "identity", color = "white", size = 0.5) +
  geom_text(aes(label = paste0(Percentage, " (", Count, ")")),
            position = position_stack(vjust = 0.5), color = "black", size = 4) +
  labs(title = "Furnishing Status Distribution in KLCC with Price > 1 Million",
       x = "",
       y = "Count") +
  theme_minimal() +
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        plot.title = element_text(size = 16, face = "bold"),
        legend.title = element_blank(),
        legend.position = "bottom",
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_blank(),
        panel.background = element_blank()) +
  scale_fill_brewer(palette = "Set3")
```

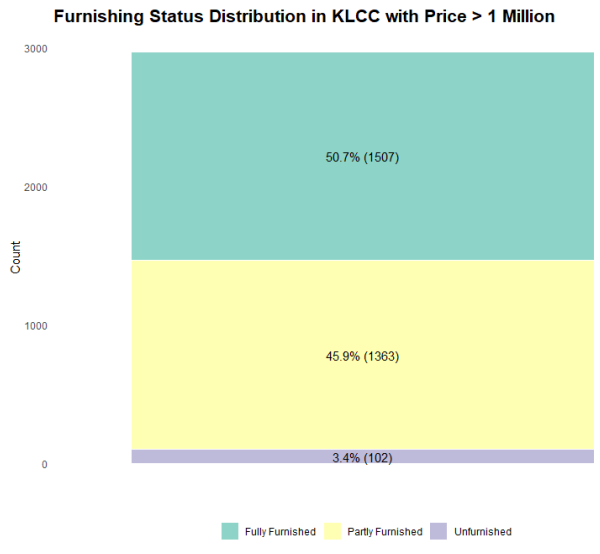


Figure 88-3.2.5.1 Stacked Bar Chart of Furnishing Status in KLCC with price greater than 1 million.

In figure 3.2.5.1, we can clearly see that this stacked bar chart shows us the distribution of furnishing status in KLCC with price greater than 1 million. We can see that fully and partly furnished houses have a higher percentage in comparison to unfurnished houses in KLCC and price greater than 1 million. Fully furnished houses have 50.7% while partly furnished houses have 45.9%. Meanwhile unfurnished houses have the least percentage, which is only 3.4%.

In plotting this clustered bar chart, I have used “ggplot2” and “dplyr” library. “ggplot2” library allows me to plot this clustered bar chart while the “dplyr” library allows me to use the pipe operator, %>%, group by() function, summarize() function and mutate() function. I have created another subset name klcc_high_price which includes data that is only in KLCC and price greater than 1 million. Then I have used the pipe operator, %>% to creates a clustered bar plot after doing data processing. I have grouped the data by furnishing status using “group_by(Property_Furnishing)” and calculated the count of each furnishing status using “summarize(Count = n())”. Since I am adding the percentage label in the clustered bar chart, mutate(Percentage = scales::percent(Count / sum(Count), accuracy = 0.1)) computes the percentage of each category. For theme_minimal() function, detailed explanation have been done on 3.2.1. theme(…) is to modifies several parts of the plot, such as removing x-axis

text and ticks, personalizing the title, and removing extraneous grid lines. Lastly, scale_fill_brewer(...) applies a color palette ("Set3" from Brewer palettes) to fill the bars.

3.2.6 Analysis 2-6: Will amount of bedroom have relationship with furnishing status of houses in KLCC with price greater than 1 million?

Objective: To find out whether the furnishing status houses in KLCC with price greater than 1 million will be affected by the amount of bedroom (<=3 or >3).

Methodology: A clustered bar chart is shown using the “ggplot2” and “dplyr” library to represent the count of each furnishing status in the KLCC location with price greater than 1 million.

```
library(ggplot2)
library(dplyr)

# Filter data for KLCC location with price > 1 million
klcc_high_price <- Filtered_dataset[
  (Filtered_dataset$Property_Location == "Klcc") &
    (Filtered_dataset$Property_Price > 1000000), ]

# Create a new column for Bedroom_Category
klcc_high_price$Bedroom_Category <- ifelse(klcc_high_price$Property_Rooms <= 3, "<=3", ">3")

# Creating Clustered Bar Chart
klcc_high_price %>%
  group_by(Bedroom_Category, Property_Furnishing) %>%
  summarize(Count = n()) %>%
  group_by(Property_Furnishing) %>%
  mutate(TotalCount = sum(Count),
         Percentage = scales::percent(Count / TotalCount, accuracy = 0.01)) %>%
  ggplot(aes(x = Bedroom_Category, y = Count, fill = Property_Furnishing)) +
  geom_bar(stat = "identity", position = "stack", color = "white", size = 0.5) +
  geom_text(aes(label = paste0(Percentage, " (", Count, ")")),
            position = position_stack(vjust = 0.5), color = "black", size = 4) +
  labs(title = "Furnishing Status Distribution in KLCC with Price > 1 Million",
       x = "Bedroom Category",
       y = "Count") +
  facet_wrap(~ Property_Furnishing) +  # Add facet_wrap to separate furnishing statuses
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 0, hjust = 0.5),  # Adjust the angle parameter
        plot.title = element_text(size = 16, face = "bold"),
        legend.title = element_blank(),
        legend.position = "bottom",
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_blank(),
        panel.background = element_blank()) +
  scale_fill_brewer(palette = "Set3")
```

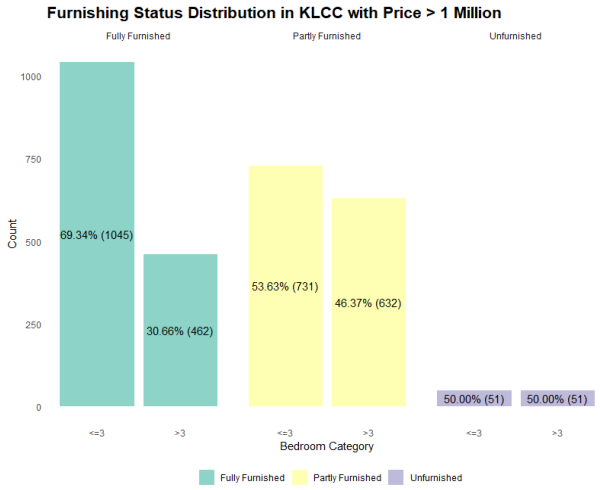


Figure 89-3.2.6.1 Clustered Bar Chart showing the relationship between bedroom category and furnishing status in KLCC with price greater than 1 million.

In figure 3.2.6.1, we can see a clustered bar chart that have been generated to show the relationship between bedroom category with furnishing status in KLCC with price greater than 1 million. In the clustered bar chart, we can see that fully furnished houses have a higher percentage in bedroom category (<=3) which is 63.3% while bedroom category (>3) only have 30.7%. This show that the number of bedrooms has a strong relationship with fully furnished houses in KLCC with price greater than 1 million. Partly furnished status has quite a similar percentage. In bedroom category (<=3), the percentage of partly furnished houses is 53.63% while in bedroom category (>3), it has 46.37%. Both of this observation are pretty close, so we will further investigate which factor will have a larger impact on partly furnished status. For unfurnished status, it has exactly the same percentage, which is 50% for both bedroom category.

In plotting this clustered bar chart, I have used 2 libraries which are “ggplot2” and “dplyr”. Their detailed explanation has been done on 3.2.5. Furthermore, I have to use back the klcc_high_price subset as well. When plotting the bar chart, I have created a new column which is “Bedroom_Category” which help me to summarize the bedroom into 2 categories only, (<=3 or >3). Then I grouped this bedroom category with property furnishing status to calculate the count of houses in each group, and then calculates the total count for each furnishing status. “Percentage Calculation” has been done to determine the percentage of residences in each bedroom category

within each furnishing status group. The x-axis of this clustered bar chart represents the Bedroom Category, y-axis represents the Count of houses, and the fill is the furnishing status of the houses. Theme_minimal and theme(…) have been explained before.

3.2.7 Analysis 2-7: Does property size have impact on the furnishing status houses in KLCC with price greater than 1 million and bedrooms?

Objective: To find out whether the different category of property size have impact on furnishing status houses in KLCC with price greater than 1 million.

Methodology: 3 bar charts is shown using the "ggplot2" and "dplyr" library to represent the count of each furnishing status in the KLCC location with price greater than 1 million and relate it to property size in different category (<1000, 1000-1800, >1800).

3.2.7.1 Analysis 2-7-1*: Does fully furnished houses in KLCC with price greater than 1 million have relationship with property size?*

```r
#Analysis 2.5.1
library(ggplot2)
library(dplyr)

# Create a new column for Square Feet Category and new subset
Filtered_dataset$Square_Feet_Category <- cut(Filtered_dataset$Property_Size,
                                             breaks = c(-Inf, 1000, 1800, Inf),
                                             labels = c("<1000", "1000-1800", ">1800"))

# Create a new subset (filtered dataset)
ff_klcc_gt1million_roomle3 <- Filtered_dataset %>%
  filter(Property_Furnishing %in% c("Fully Furnished"),
         Property_Price > 1000000,
         Property_Location == "Klcc",
         Property_Rooms <= 3)

# Creating a bar chart
ff_klcc_gt1million_roomle3 %>%
  group_by(Square_Feet_Category) %>%
  summarize(Count = n()) %>%
  mutate(Percentage = scales::percent(Count / sum(Count), accuracy = 0.1)) %>%
  ggplot(aes(x = Square_Feet_Category, y = Count, fill = "Fully Furnished")) +
  geom_bar(stat = "identity", position = "stack", color = "white", size = 0.5) +
  geom_text(aes(label = paste0(Percentage, " (", Count, ")")),
            position = position_stack(vjust = 0.5), color = "black", size = 4) +
  labs(title = "Fully Furnished Houses in KLCC with Price > 1 Million (<=3 Bedrooms)",
       x = "Square Feet Category",
       y = "Count") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 0, hjust = 0.5),
        plot.title = element_text(size = 16, face = "bold"),
        legend.title = element_blank(),
        legend.position = "bottom",
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_blank(),
        panel.background = element_blank()) +
  scale_fill_manual(values = "lightblue")  # You can customize the color if needed
```

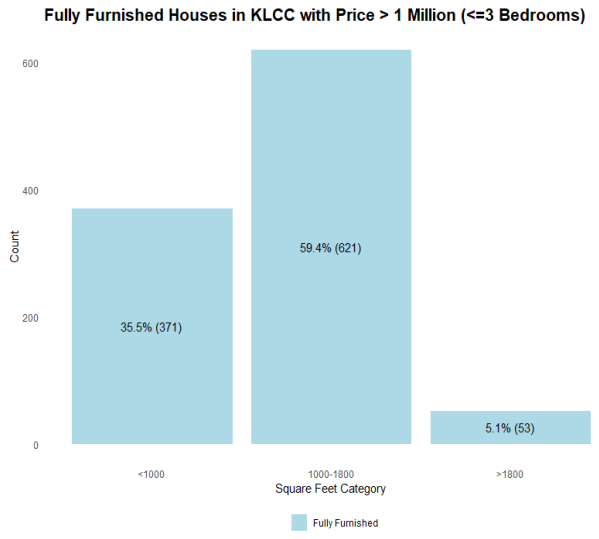


Figure 90-3.2.7.1 Bar Chart showing the relationship between square feet category and full furnished houses in KLCC with price greater than 1 million with bedroom size <=3.

In this bar chart, we can clearly see that there is a higher percentage fall in the 1000-1800 square feet category for fully furnished houses with smaller than or equal to 3 bedrooms in KLCC with price greater than 1 million. It has 59.4% which is 621 houses in the 1000-1800 square feet category. Meanwhile, there is 35.5% which is 371 houses out of a total of 1045 fully furnished houses in KLCC with price greater than1 million and have <=3 bedrooms fall in <1000 square feet category and only 5.1% which is 53 houses out of 1045 fully furnished houses in KLCC with price greater than1 million and have <=3 bedrooms fall in >1800 square feet category. First, I have created a new column in my Filetered_dataset which is Square Feet Category. I have used cut() function during this process to categorize the sizes into 3 categories, which is <1000, 1000-1800 and >1800 square feet. Furthermore, I have created a new subset as well which only includes data that fulfill these conditions (fully furnished, in KLCC, price greater than 1 million and bedroom <=3).To plot this bar chart, I have used the

subset I just created and then grouped it with the Square_Feet_Category and count the data using (summarize(Count = n()) like previously how I explained in previous analysis. Geom_bar() and geom_text() are used to create a stacked bar and add labels with percentages.

3.2.7.2 Analysis 2-7-2 Does partly furnished houses in KLCC with price greater than 1 million with bedroom (<=3) have relationship with property size?

```
#Analysis 2.5.2
# Create a new subset for partly furnished houses
pf_klcc_gt1million_roomle3 <- Filtered_dataset %>%
  filter(Property_Furnishing %in% c("Partly Furnished"),
         Property_Price > 1000000,
         Property_Location == "Klcc",
         Property_Rooms <= 3)

# Calculate percentages within each square feet category for partly furnished houses
pf_klcc_gt1million_roomle3 %>%
  group_by(Square_Feet_Category) %>%
  summarize(Count = n()) %>%
  mutate(Percentage = scales::percent(Count / sum(Count), accuracy = 0.1)) %>%
  ggplot(aes(x = Square_Feet_Category, y = Count, fill = "Partly Furnished")) +
  geom_bar(stat = "identity", position = "stack", color = "white", size = 0.5) +
  geom_text(aes(label = paste0(Percentage, " (", Count, ")")),
            position = position_stack(vjust = 0.5), color = "black", size = 4) +
  labs(title = "Partly Furnished Houses in KLCC with Price > 1 Million (<=3 Bedrooms)",
       x = "Square Feet Category",
       y = "Count") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 0, hjust = 0.5),
        plot.title = element_text(size = 16, face = "bold"),
        legend.title = element_blank(),
        legend.position = "bottom",
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_blank(),
        panel.background = element_blank()) +
  scale_fill_manual(values = "lightgreen")  # You can customize the color if needed
```

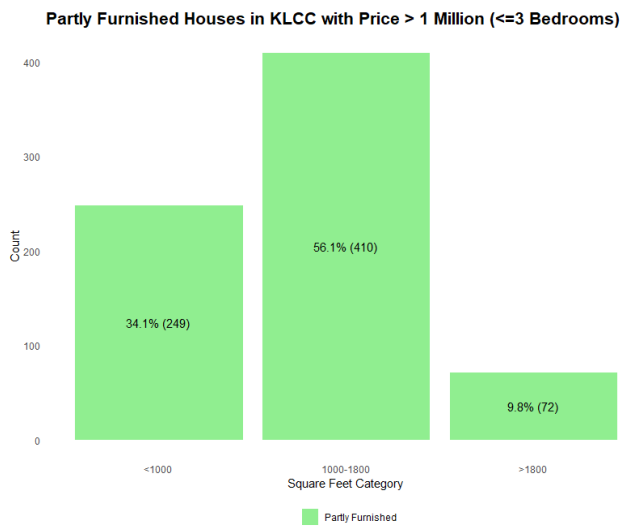


Figure 91-3.2.7.2.1 Bar Chart showing the relationship between square feet category and partly furnished houses in KLCC with price greater than 1 million with bedroom size <=3.

I have plotted this graph using similar method in 3.2.7.1, I have created another subset which named "pf_klcc_gt1million_roomle3" which include the partly furnished houses in KLCC with price greater than 1 million and having less than or equal to 3 bedrooms.

After that, I have use back the same method in generating the bar chart in 3.2.7.1 by changing the subset I used to "pf_klcc_gt1million_roomle3" and changing the colour to make it become more contrast.

3.2.7.3 Analysis 2-7-3 Does partly furnished houses in KLCC with price greater than 1 million with bedroom (>3) have relationship with property size?

```
# Create a new subset for partly furnished houses with more than 3 bedrooms
pf_klcc_gt1million_roomgt3 <- Filtered_dataset %>%
  filter(Property_Furnishing %in% c("Partly Furnished"),
         Property_Price > 1000000,
         Property_Location == "Klcc",
         Property_Rooms > 3)

# Calculate percentages within each square feet category for partly furnished houses
pf_klcc_gt1million_roomgt3 %>%
  group_by(Square_Feet_Category) %>%
  summarize(Count = n()) %>%
  mutate(Percentage = scales::percent(Count / sum(Count), accuracy = 0.1)) %>%
  ggplot(aes(x = Square_Feet_Category, y = Count, fill = "Partly Furnished")) +
  geom_bar(stat = "identity", position = "stack", color = "white", size = 0.5) +
  geom_text(aes(label = paste0(Percentage, " (", Count, ")")),
            position = position_stack(vjust = 0.5), color = "black", size = 4) +
  labs(title = "Partly Furnished Houses Distribution in KLCC with Price > 1 Million (>3 Bedrooms)",
       x = "Square Feet Category",
       y = "Count") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 0, hjust = 0.5),
        plot.title = element_text(size = 16, face = "bold"),
        legend.title = element_blank(),
        legend.position = "bottom",
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_blank(),
        panel.background = element_blank()) +
  scale_fill_manual(values = "lightpink")  # You can customize the color if needed
```

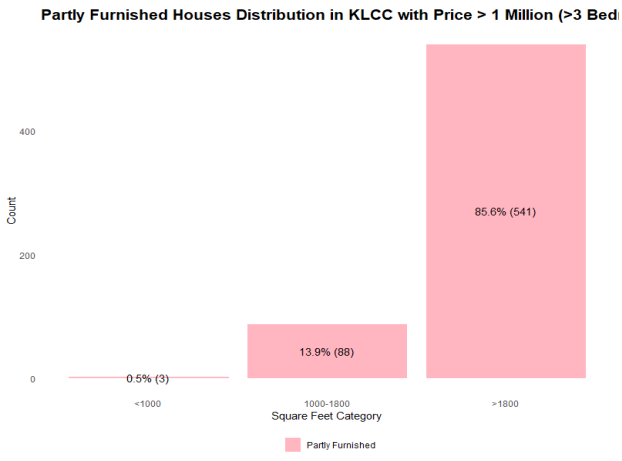


Figure 92-3.2.7.3.1 Bar Chart showing the relationship between square feet category and partly furnished houses in KLCC with price greater than 1 million with bedroom size >3.

Same towards this analysis which is similar to 3.2.7.1, I have created new subset named "pf_klcc_gt1million_roomgt3" which include the partly furnished houses in KLCC with price greater than 1 million and having greater than 3 bedrooms. Next, I edited the subset in the original codes and changed its colour.

Conclusion for square feet category:

From these square feet category analysis, we can know that this 1000-1800 square feet category has a much higher impact on fully and partly furnished houses in KLCC with price greater than 1 million, less than or equal to 3 bedrooms as shown in analysis 2.7.1 and 2.7.2. Meanwhile >1800 square feet category has impact on partly furnished houses in KLCC with price greater than 1 million, bedroom size greater than 3 as shown in analysis 2.7.3.

3.2.8 Analysis 2-8: Does built area (Built up or Land Area) have impact on the furnishing status houses in KLCC with price greater than 1 million?

Objective: To find out whether the different category of built area have impact on furnishing status houses in KLCC with price greater than 1 million.

Methodology: 3 3D pie charts is shown using the "plotrix" library to represent the count of each furnishing status in the KLCC location with price greater than 1 million and relate it to property size in different category (<1000, 1000-1800, >1800).

3.2.8.1 Analysis 2-8-1: Does built area have impact on fully furnished houses in KLCC with price greater than 1 million, less than or equal to 3 bedrooms and have square feet between 1000 and 1800.

```
# Load the required libraries
library(plotrix)

ff_klcc_gt1million_roomle3_sq1000_1800$Built_Area <-
  as.factor(ff_klcc_gt1million_roomle3_sq1000_1800$Built_Area)

# Get values and labels
values <- table(ff_klcc_gt1million_roomle3_sq1000_1800$Built_Area)
labels <- names(values)

# Calculate percentages and frequencies
percentages <- sprintf("%.2f%%", (values / sum(values)) * 100)
frequencies <- paste("(", values, ")", sep = "")

# Create 3D explode pie chart using plotrix
pie3D(values,
      col = c("lightblue", "lightyellow"),
      border = "black",
      main = "Built Area Distribution in KLCC (Fully Furnished, <=3 Bedrooms, 1000-1800 Sq Feet)",
      labels = paste(labels, "\n", percentages, " ", frequencies, sep = ""),
      labelcex = 1,
      labelcol = "black")
```

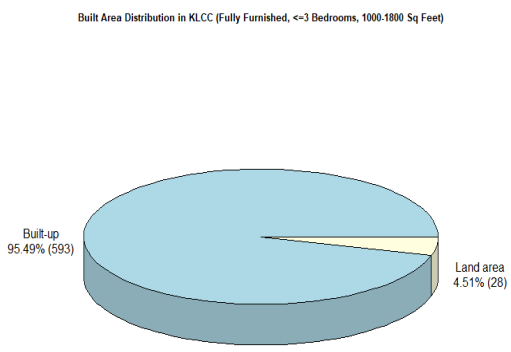


Figure 93-3.2.8.1.1 3D Pie Chart showing the relationship between built area and fully furnished houses in KLCC with price greater than 1 million with bedroom size <=3 and square feet size 1000-1800.

3.2.8.2 Analysis 2-8-2: Does built area have impact on partly furnished houses in KLCC with price greater than 1 million, less than or equal to 3 bedrooms and have square feet between 1000 and 1800.

```
# Create new subset for partly furnished houses
pf_klcc_gt1million_roomle3_sq1000_1800 <- Filtered_dataset %>%
  filter(Property_Furnishing %in% c("Partly Furnished"),
         Property_Price > 1000000,
         Property_Location == "Klcc",
         Property_Rooms <= 3,
         Square_Feet_Category %in% c("1000-1800"))

pf_klcc_gt1million_roomle3_sq1000_1800$Built_Area <-
  as.factor(pf_klcc_gt1million_roomle3_sq1000_1800$Built_Area)

# Get values and labels
values_pf <- table(pf_klcc_gt1million_roomle3_sq1000_1800$Built_Area)
labels_pf <- names(values_pf)

# Calculate percentages and frequencies
percentages_pf <- sprintf("%.2f%%", (values_pf / sum(values_pf)) * 100)
frequencies_pf <- paste("(", values_pf, ")", sep = "")

# Create 3D explode pie chart using plotrix
pie3D(values_pf,
      col = c("#F96167", "#F9E795"),
      border = "black",
      main = "Built Area Distribution in KLCC (Partly Furnished, <=3 Bedrooms, 1000-1800 Sq Feet)",
      labels = paste(labels_pf, "\n", percentages_pf, " ", frequencies_pf, sep = ""),
      labelcex = 1,
      labelcol = "black")
```

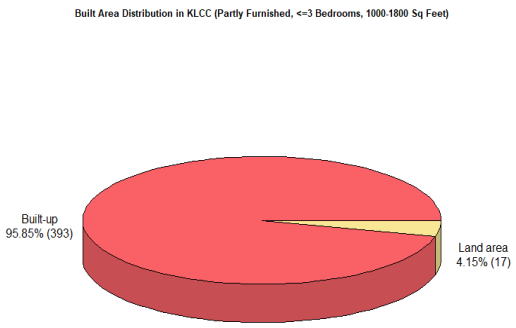


Figure 94-3.2.8.2.1 3D Pie Chart showing the relationship between built area and partly furnished houses in KLCC with price greater than 1 million with bedroom size <=3 and square feet size 1000-1800.

3.2.8.3 Analysis 2-8-3: Does built area have impact on partly furnished houses in KLCC with price greater than 1 million, have greater than 3 bedrooms and have square feet >1800.

```
# Create new subset for partly furnished houses with >3 bedrooms and square feet >1800
pf_klcc_gt1million_roomgt3_sqgt1800 <- Filtered_dataset %>%
  filter(Property_Furnishing %in% c("Partly Furnished"),
         Property_Price > 1000000,
         Property_Location == "Klcc",
         Property_Rooms > 3,
         Square_Feet_Category %in% c(">1800"))

pf_klcc_gt1million_roomgt3_sqgt1800$Built_Area <-
  as.factor(pf_klcc_gt1million_roomgt3_sqgt1800$Built_Area)

# Get values and labels
values_pf_gt3_sqgt1800 <- table(pf_klcc_gt1million_roomgt3_sqgt1800$Built_Area)
labels_pf_gt3_sqgt1800 <- names(values_pf_gt3_sqgt1800)

# Calculate percentages and frequencies
percentages_pf_gt3_sqgt1800 <- sprintf("%.2f%%", (values_pf_gt3_sqgt1800 / sum(values_pf_gt3_sqgt1800)) * 100)
frequencies_pf_gt3_sqgt1800 <- paste("(", values_pf_gt3_sqgt1800, ")", sep = "")

# Create 3D explode pie chart using plotrix
pie3D(values_pf_gt3_sqgt1800,
      col = c("#2F3C7E", "#FBEAEB"),
      border = "black",
      main = "Built Area Distribution in KLCC (Partly Furnished, >3 Bedrooms, >1800 Sq Feet)",
      labels = paste(labels_pf_gt3_sqgt1800, "\n", percentages_pf_gt3_sqgt1800, " ", frequencies_pf_gt3_sqgt1800, sep = ""),
      labelcex = 1,
      labelcol = "black")
```

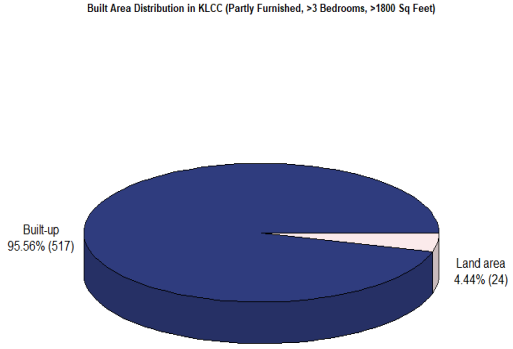


Figure 95-3.2.8.3.1 3D Pie Chart showing the relationship between built area and partly furnished houses in KLCC with price greater than 1 million with bedroom size >3 and square feet size >1800.

The method of plotting these 3D pie charts is already explained in 3.2.4 during the analysis of furnishing Status Distribution in KLCC location.

Conclusion for Built Area:

In these 3 analyses, we can clearly see that Built up stands a much higher percentage in all this 3-3D pie chart. This meant that the built-up has a higher impact compared to land area.

Final conclusion:

After all my testing and analysis that I have done on all these attributes (furnishing status, location, price, bedroom size, square feet and lastly built area), I came to 3 conclusions.

First conclusion is that 20.22% of houses in KLCC with price greater than 1 million are fully furnished, bedroom size of less than or equal to 3, between 1000 to 1800 square feet and built-up type.

Second conclusion is that 13.36% of houses in KLCC with price greater than 1 million are partly furnished, bedroom size of less than or equal to 3, between 1000 to 1800 square feet and built-up type.

Third conclusion is that 17.40% of houses in KLCC with price greater than 1 million are partly furnished, bedroom size of larger 3, greater than 1800 square feet and built-up type.

3.2.9 Extra Features:

	Explanation	First use at Analysis
1. T-test	To investigate the relationship between the furnishing status and price.	3.2.1
2. ANOVA Test	To investigate relationship between furnishing status and square feet category with price.	3.2.2
3. Violin Plot	geom_violin	3.2.2
4. theme_minimal (ggplot2 library)	theme function changes the visual appearance of the plot to a minimalistic manner.	3.2.3 & other
5. theme()	used to create additional theme customization.	3.2.3 & other
6. table()	to calculate the number of unique items in a vector or data frame column.	3.2.3 & other
7. prop.table()	to determine proportions (percentages) using the counts returned by the table() function.	3.2.3 & other
8. theme_economist() (ggthemes library)	set the visual appearance of your ggplot visuals to resemble the style commonly found in publications such as The Economist magazine.	3.2.3 & other
9. plotrix library	allow me to use the pie3D() function.	3.2.4 & 3.2.8
10. sprintf()	used to format and output character strings in a flexible way.	3.2.4
11. pie3D()	part of the plotrix package and is used to build a 3D exploding pie chart.	3.2.4 & 3.2.8
12. scale_fill_brewer(...)	used to set the color palette for the fill aesthetic in a plot.	3.2.2& 3.2.5&3.2.6
13. cut()	used to divide a numeric vector into different ranges.	3.2.7

3.3 Objective 3: To investigate the impact of Rooms on Price in KLCC (SIM SAU YANG TP065596)

This analysis **studies the relationship and differences** between the independent variables including room number, size, furnishing status, and built area and their **impact on price (>1M) and location (KLCC)** to **validate the hypothesis**.

Question to validate hypothesis: If KLCC and price >1M properties are selected, which rooms, size, furnishing, and built area category are more likely to be selected?

```r
# import library
library(dplyr)
library(ggplot2)
library(moments)
library(corrplot)
library(treemapify)
library(plotrix)
library(ggpubr)
library(vcd)
library(VennDiagram)
```

```r
# Get the data set
df <- Filtered_dataset

# Add columns to categorize the numeric value based on hypothesis Criteria
df <- df %>%
  mutate(IsKLCC =
           ifelse(Property_Location == "Klcc", "KLCC", "Non-KLCC")) %>%
  mutate(Price_Category =
           ifelse(Property_Price > 1000000, ">1M", "<=1M")) %>%
  mutate(Room_Category =
           ifelse(IsStudio == TRUE, "Studio", ifelse(Property_Rooms <= 3, "<=3", ">3"))) %>%
  mutate(Size_Category =
           case_when(Property_Size < 1000 ~ "<1000", Property_Size >= 1000 &
                       Property_Size <= 1800 ~ "1000-1800", Property_Size > 1800 ~ ">1800"))
```

Figure 96-3.3.1: Import Library and Prepared Dataset

Import used libraries. The new columns are added to categorize numeric columns.

3.3.1 Analysis 1-1: What is the difference between number of rooms with KL and KLCC?

```r
# Perform Wil coxon-Mann-Whitney test
wilcox.test(Property_Rooms ~ IsKLCC, data = df)
# Hypothesis 0: No difference in rooms between KL & KLCC
# Hypothesis 1: Difference in rooms between KL & KLCC
```

```
        Wilcoxon rank sum test with continuity correction

data:  Property_Rooms by IsKLCC
W = 48254726, p-value < 2.2e-16
alternative hypothesis: true location shift is not equal to 0
```

Figure 97-3.3.1.1: Wilcoxon-Mann Whitney (WMW) test on (Property_Rooms) by (IsKLCC)

The Wilcoxon rank sum test statistic (W) is 46297299 and the p-value is less than 2.2e-16, which suggests that there is a significant association, which the KL and KLCC have different distribution of the number of rooms. H0 is rejected.

```r
# Create array for percentage of KLCC & Rooms Category
value = c(
  round(nrow(df[df$IsKLCC == "KLCC" & df$Room_Category == "<=3", ])
        /nrow(df[df$IsKLCC == "KLCC", ])*100,2),
  round(nrow(df[df$IsKLCC == "KLCC" & df$Room_Category == ">3", ])
        /nrow(df[df$IsKLCC == "KLCC", ])*100,2),
  round(nrow(df[df$IsKLCC == "KLCC" & df$Room_Category == "Studio", ])
        /nrow(df[df$IsKLCC == "KLCC", ])*100,2),
  round(nrow(df[df$Room_Category == "<=3", ])/nrow(df)*100,2),
  round(nrow(df[df$Room_Category == ">3", ])/nrow(df)*100,2),
  round(nrow(df[df$Room_Category == "Studio", ])/nrow(df)*100,2))

multi_pie <- data.frame(
  Location = rep(levels(factor(df$IsKLCC)), each = 3),
  Rooms = rep(levels(factor(df$Room_Category)), time = 2),
  Value = value
)

# Reorder the levels of IsKLCC
multi_pie$Location[multi_pie$Location == "Non-KLCC"] <- "All Location"
multi_pie$Location <- factor(multi_pie$Location,
                             levels = c("All Location", "KLCC"))

# Multilevel Pie Chart: Proportion of Room Category by Location
ggplot(multi_pie, aes(x = Location, y = Value, fill = Rooms)) +
  geom_col() +scale_x_discrete(limits = c("", "All Location","KLCC")) +
  coord_polar("y")+ geom_text(aes(label = paste0(Value, "%")),
                              position = position_stack(vjust = 0.4),
                              size = 5) +
  labs(title = "Proportion of Room Category by Location",
       subtitle = "Level 1: All Location | Level 2: KLCC Only",
       fill = "Room Category") + theme_bw()
```

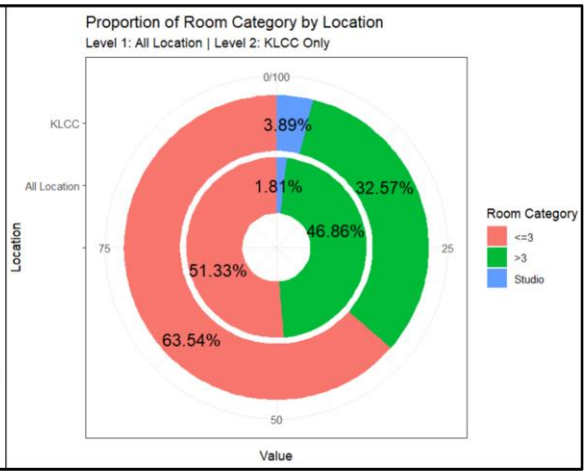


Figure 98-3.3.1.2: Multilevel Pie Chart for (Room_Category) by (IsKLCC)

The All Location pie shows 51.33% and 46.66% with the number of rooms <=3 and >3, which have similar proportions. In KLCC, the <=3 rooms percent is higher, which is 63.54% (61% exclude studio), which has a higher probability in hypothesis, while the >3 rooms are 46.86% (39% exclude studio), showing a difference in location, as indicated by the WMW test.

3.3.1 Analysis 1-2: What is the relationship between number of rooms and price?

```
# Exclude Studio
a1.df = df %>% filter(IsStudio == FALSE)

# Spearman's rank correlation coefficient
cor.test(a1.df$Property_Rooms,
         a1.df$Property_Price,
         method = "spearman",
         exact = FALSE)
# Hypothesis 0: No relationship between room & Price
# Hypothesis 1: Relationship between room & Price
```

```
        Spearman's rank correlation rho

data:  a1.df$Property_Rooms and a1.df$Property_Price
S = 4.5547e+12, p-value < 2.2e-16
alternative hypothesis: true rho is not equal to 0
sample estimates:
      rho
0.4828089
```

Figure 99-3.3.1.2: Spearman's rank correlation between (Property_Rooms) and (Property_Price)

The Spearman's correlation coefficient (rho) is 0.4828, suggesting a moderate positive monotonic relationship, which means that as the number of rooms increases, the price will tend to increase. The p-value is small (2.2e-16), which is less than the conventional significance level of 0.05, this strengthens that the correlation is unlikely to occur by random chance. The null hypothesis (H0) is rejected.

```
# Box Plot: Price for number of rooms room
ggplot(a1.df,
       aes(y = Property_Price,
           x = Property_Rooms,
           group = Property_Rooms)) +
  geom_boxplot(fill = "pink", color = "black") +
  labs(title = "Property Price by Number of Rooms",
       x = "Number of Rooms",
       y = "Property Price",
       fill = "Number of Rooms") +
  theme_minimal()
```

```
# Only KLCC
a1.df.klcc = a1.df[a1.df$IsKLCC == "KLCC", ]

# calculate mean & standard error
mean_prices <- tapply(a1.df.klcc$Property_Price, a1.df.klcc$Property_Rooms, mean)

sds <- tapply(a1.df.klcc$Property_Price, a1.df.klcc$Property_Rooms, sd)
counts <- table(a1.df.klcc$Property_Rooms)
standard_errors <- as.vector(sds / sqrt(counts))

room_prices_df <- data.frame(Rooms = names(mean_prices),
  Mean_Price = as.numeric(mean_prices), Standard_Error = standard_errors)

# Bar Chart: Mean price for number of room in KLCC
ggplot(room_prices_df, aes(x = factor(Rooms), y = Mean_Price, fill = factor(Rooms))) +
  geom_bar(stat = "identity", position = "dodge", color = "black") +
  geom_errorbar(aes(ymin = Mean_Price - Standard_Error, ymax = Mean_Price + Standard_Error),
                position = position_dodge(width = 0.8), width = 0.5, color = "red") +
  labs(title = "Mean Price for Number of Room in KLCC", x = "Rooms", y = "Mean Price") +
  scale_fill_manual(values = c("blue", "skyblue", "lightblue", "cyan")) +
  geom_text(aes(label = sprintf("%.2f", Mean_Price)),
            position = position_dodge(width = 0.8), vjust = -0.7)
```

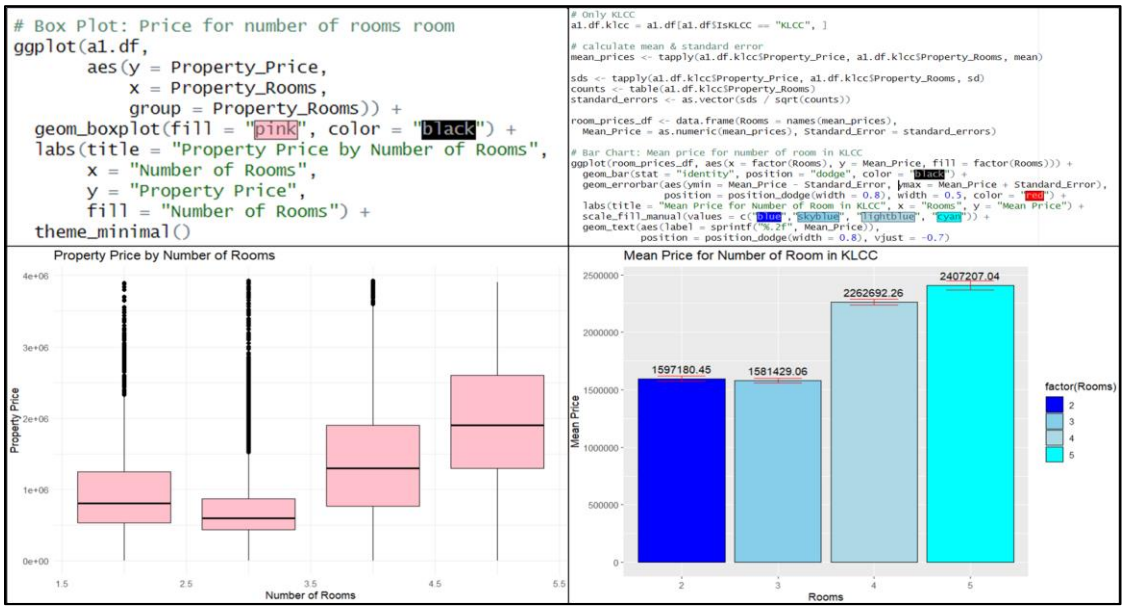


Figure 100-3.3.1.3: Box Plot of (Property_Price) by (Property_Rooms) & Bar Chart with Error Bar of (Mean Price) by (Property_Rooms) in KLCC

The box plot and the bar chart show as the number of rooms increases, there is a generally increasing trend in prices. While the IQR appears to widen with increasing no. rooms, the general price difference (range) is even greater. There are outliers observed in each category. There is a positive association, but the mean price of all room categories in KLCC is >1M, which suggests a lower impact on price in the hypothesis.

```
summary_table <- a1.df.klcc %>%
  group_by(Room_Category, Price_Category) %>%
  summarise(Counts = n()) %>%
  group_by(Room_Category) %>%
  mutate(Percentage = Counts / sum(Counts) * 100)

# Percentage Stacked Bar Chart: Percentage for Price Category by Room Category
ggplot(summary_table,
       aes(x = Room_Category,
           y = Percentage,
           fill = Price_Category,
           label = sprintf("%.2f%%", Percentage))) +
  geom_bar(stat = "identity",
           position = "fill") +
  geom_text(position = position_fill(vjust = 0.5),
            size = 7) +
  labs(title = "Percentage for Price Category by Room Category in KLCC",
       x = "Price Category",
       y = "Percentage",
       fill = "Room Category") +
  scale_y_continuous(
    labels = scales::percent_format(scale = 1)) +
  theme_minimal()
```

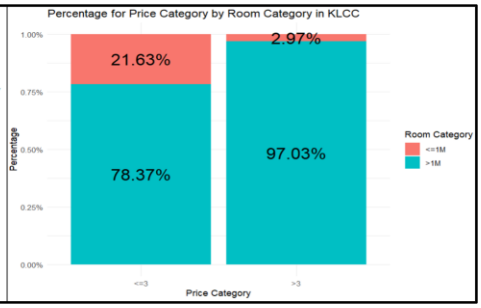


Figure 101-3.3.1.4: Percentage Stacked Bar Chart for (Price_Category) by (Room_Category)

The bar chart shows that there are 97.03% of rooms have a price >1M in no. rooms >3, higher than the no. rooms <=3 which is 78.37%. The larger proportion suggests selecting no. rooms >3 in hypothesis.

3.3.1 Analysis 1-3: Which Room Category presents the higher proportion intersection with price and KLCC?

```
# Calculate counts for room category &
summary_table <- a1.df.klcc %>%
  group_by(Room_Category, Price_Category) %>%
  summarise(Counts = n()) %>%
  mutate(Price_Category =
           paste0(Price_Category, "\n", Counts, "\n",
                  sprintf("%.2f%%", round(Counts/nrow(a1.df.klcc)*100,2))))

# Treemap: The proportion for Price Category by Room Category
ggplot(summary_table, aes(area = Counts,
                          fill = Room_Category,
                          label = Price_Category,
                          subgroup = Room_Category)) +
  geom_treemap() +
  geom_treemap_subgroup_border(colour = "white", size = 5) +
  geom_treemap_subgroup_text(place = "topleft",
                             colour = "green", size = 25) +
  geom_treemap_text(colour = "white", place = "centre",
                    size = 25, fontface = "bold") +
  scale_fill_manual(values =
                      c("<=3" = "brown",
                        ">3" = "purple")) +
  labs(title = "Proportion for Price Category by Room Category in KLCC")
```

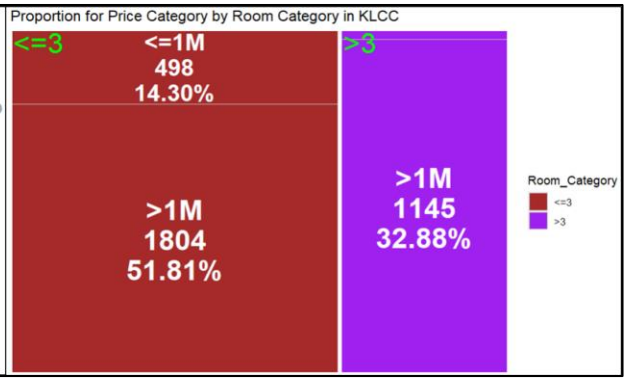


Figure 102-3.3.1.5: Treemap for (Price_Category) by (Room_Category) in KLCC

Through the testing on Location in KLCC, no. rooms <=3 is suggested, but the price testing suggests no. rooms >3 has higher impact for hypothesis. A cross-intersection of both data is conducted using [Percentage = (Room_Category % in KLCC) * (Price_Category in KLCC)] to determine which room category has higher weightage. The treemap shows the no. room <=3 with price >1M has a higher distribution (51.81%) in KLCC. Hence, No. rooms <=3 has larger distribution (61%, 1804/2949) in KLCC with price >1M for higher impact on hypothesis.

3.3.2 Analysis 2-1: What is the relationship between Size, Price, and Location?

```
# Select the Size, Price, Rooms Column
size_price_rooms <- df %>%
  select(Property_Rooms,
         Property_Price,
         Property_Size)

correlation_matrix <-
  cor(size_price_rooms,
      method="pearson")

# Correlation Plot: Relationship of Rooms, Price, Size
corrplot(correlation_matrix,
         method = "pie",
         type = "lower",
         order = "hclust",
         tl.cex = 0.7)
```

```
> correlation_matrix
              Property_Rooms Property_Price Property_Size
Property_Rooms      1.0000000      0.4544535     0.6649115
Property_Price      0.4544535      1.0000000     0.7175036
Property_Size       0.6649115      0.7175036     1.0000000
```

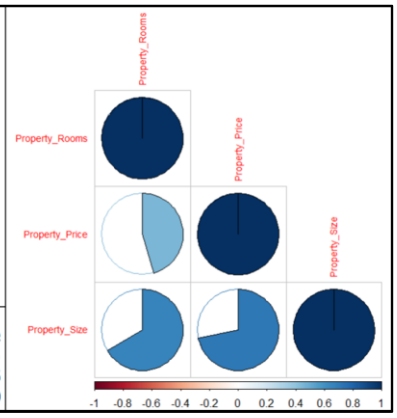


Figure 103-3.3.2.1: Pearson Correlation for (Property_Size), (Property_Price) & (Property_Room)

The Pearson Correlation suggests a high relationship with a correlation of 0.7175 between size and price. There is also a relationship with a correlation of 0.6649 observed between size and rooms.

```
# Conduct t.test for Size & KLCC/Non-KLCC
t.test(Property_Size ~ IsKLCC, data = a1.df)
```

```
# Hypothesis 0: No difference in Size between KL & KLCC
# Hypothesis 1: Difference in Size between KL & KLCC
```

```
        Welch Two Sample t-test

data:  Property_Size by IsKLCC
t = -1.1185, df = 4152.1, p-value = 0.2634
alternative hypothesis: true difference in means between
group KLCC and group Non-KLCC is not equal to 0
95 percent confidence interval:
 -48.29166  13.20645
sample estimates:
    mean in group KLCC mean in group Non-KLCC
              1620.771               1638.314
```

Figure 104-3.3.2.2: T.Test for (Size) & (IsKLCC)

The t-value -1.1185 shows that the mean Size in KLCC is smaller than the mean Size of non-KLCC. The p-value is 0.2634, which is >0.05, not having enough evidence to reject the null hypothesis, which suggests no difference in the mean Size between non-KLCC and KLCC.

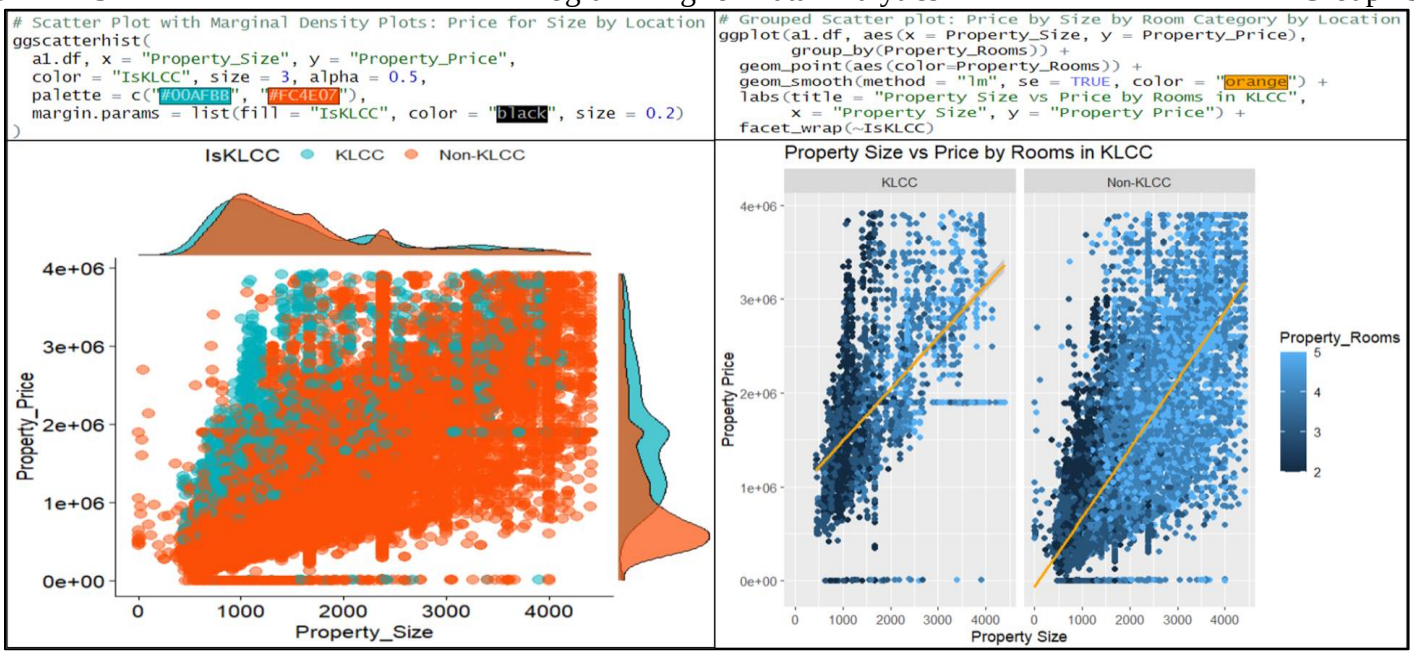


Figure 105-3.3.2.3: *ScatterPlot with Density of (Property_Price) for (Property_Size) by (IsKLCC) & Grouped ScatterPlot of (Property_Price) for (Property_Size) by (IsKLCC) by (Room_Category)*

The first scatterplot shows not much difference in the density of size in KLCC and non-KLCC, as suggested in the t-test, no difference between property size in KLCC and non-KLCC. The dots in KLCC are mostly distributed at higher prices. In the grouped scatterplot, according to the regression line, there is a relationship, as the size increases, the price also increases. Other than that, there is also a relationship between size and room, as the size increases, the no. rooms also increase.

3.3.2 Analysis 2-2: What is the distribution of Size? Which size category (<1000, 1000-1800, >1800) has a larger proportion?

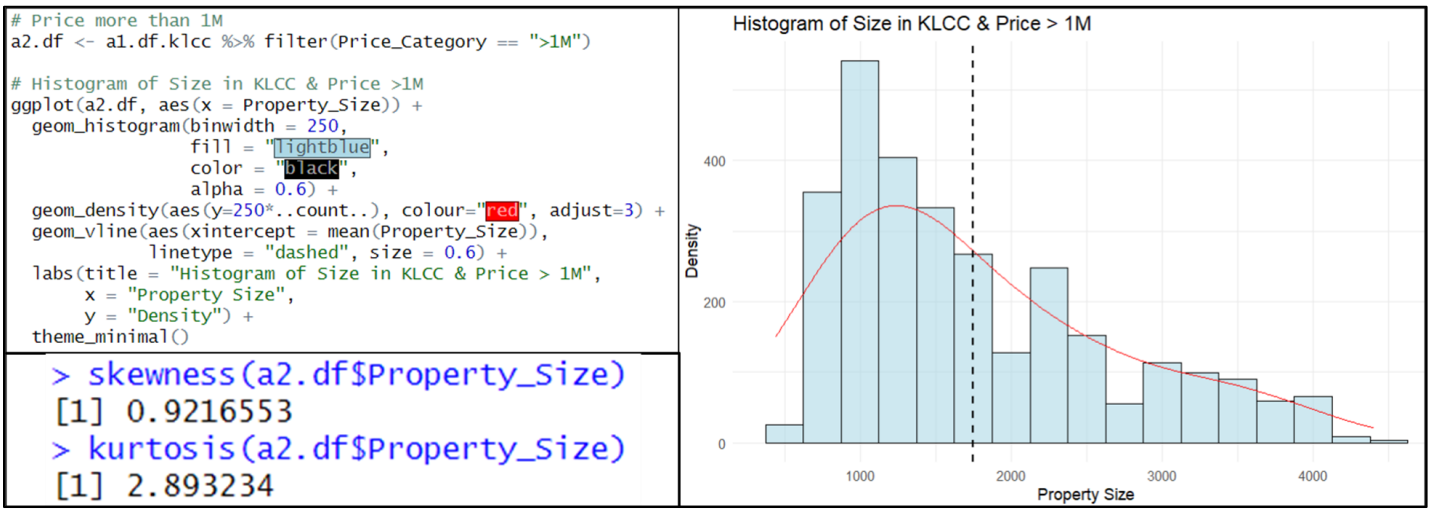


Figure 106-3.3.2.4: *Histogram of Size in in KLCC & Price >1M*

The histogram of size in KLCC and Price >1M shows a right-skewed distribution with skewness of 0.9216 and thin-tailed (Platykurtic) with kurtosis of 2.8932. Most of the records are in the size category of <1000 sqft and 1000-1800 sqft.

```r
# Group the Size Category by Room Category
summary_table <- a2.df %>%
  group_by(Room_Category, Size_Category) %>%
  summarise(Counts = n()) %>%
  mutate(Percentage = round(Counts/nrow(a2.df)*100,2)) %>%
  mutate(Label = paste0(Counts, "\n",sprintf("%.2f%%", Percentage)))
summary_table$Size_Category =
  factor(summary_table$Size_Category, levels = c(">1800", "1000-1800", "<1000"))

# Calculate row totals percentage
row_totals <- summary_table %>%
  group_by(Size_Category) %>%
  summarise(Counts = sum(Counts), Percentage = sum(Percentage))
# Calculate column totals percentage
col_totals <- summary_table %>%
  group_by(Room_Category) %>%
  summarise(Counts = sum(Counts), Percentage = sum(Percentage))

# Matrix Bubble Chart: Proportion of Size and Room Category in KLCC with Price > 1M
ggplot(summary_table,
       aes(x = Room_Category, y = Size_Category,
           colour = Room_Category, size = Counts)) +
  geom_point() +
  geom_text(aes(label = Label), colour = "black", size = 5) +
  geom_text(data = col_totals, aes(x = Room_Category, y = "Total",
                                   label = sprintf("%.2f%%", Percentage)),
            colour = "black", size = 4, vjust = -1) +
  geom_text(data = row_totals, aes(x = "Total", y = Size_Category,
                                   label = sprintf("%.2f%%", Percentage)),
            colour = "black", size = 4, hjust = 1) +
  scale_x_discrete(position = "top") +
  scale_size_continuous(range = c(18, 42)) +
  scale_color_brewer(palette = "Set2") +
  labs(title = "Proportion of Size & Rooms in KLCC & Price > 1M",
       x = "Room Category", y = "Size Category") +
  theme(legend.position = "none", panel.background = element_blank(),
        panel.grid = element_blank(), axis.ticks = element_blank())
```

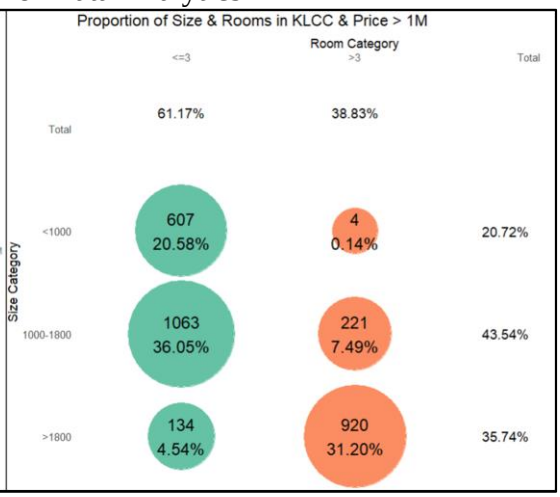


*Figure 107-3.3.2.*5: *Bubble Chart for Distribution of (Size_Category) by (Room_Category)*

Within the properties in KLCC and Price >1M, 43.54% of the properties have sizes within 1000-1800 sqft., as stated in the histogram. While the larger room has a larger size, leading to the largest bubble in size of >1800 sqft with >3 rooms (31.20%).

```r
# Calculate percentage in room category <=3
size_count <- table(a2.df$Size_Category[a2.df$Room_Category == "<=3"])
percentage_counts <-
  round((size_count / sum(size_count)) * 100, 2)
percentage_counts_labels <-
  paste0(percentage_counts, "%", "\n", paste0(size_count))

# 3D Pie Chart: Percentage for Size Category in Room <=3, Price >1M, in KLCC
pie3D(size_count, labels = percentage_counts_labels,
      main = "Percentage for Size Category in
      Room <=3, Price >1M, in KLCC", col = factor(size_count),
      height = 0.15, radius = 1, theta = 0.7)
legend("right", legend = names(size_count),
       fill = factor(size_count),
       title = "Size Category", cex = 0.8)
```

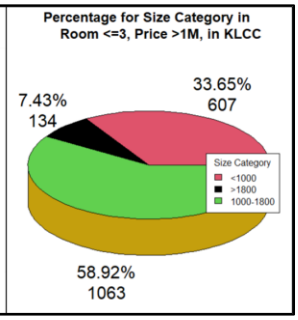


*Figure 108-3.3.2.*6*: 3D Pie Chart for Distribution of (Size_Category) with (Rooms_Category <=3)*

Within the <=3 room category properties, the size between 1000-1800 sqft has the largest proportion (58.92%), which has a higher chance of impact according to the hypothesis.

3.3.3 Analysis 3-1: Is there difference between Furnishing status with price and location?

```r
# Hypothesis a0: No difference in price between Property_Furnishing
# Hypothesis a1: Difference in price between Property_Furnishing
# Hypothesis b0: No difference in price between IsKLCC
# Hypothesis b1: Difference in price between IsKLCC
# Hypothesis c0: No difference in price between interaction of Property_Furnishing & IsKLCC
# Hypothesis c1: Difference in price between  interaction of Property_Furnishing & IsKLCC
> # ANOVA testing: Price, Furnishing, Location
> anova_result <- aov(Property_Price ~ Property_Furnishing * IsKLCC, data = df)
> anova_result
Call:
   aov(formula = Property_Price ~ Property_Furnishing * IsKLCC,
    data = df)

Terms:
                Property_Furnishing       IsKLCC Property_Furnishing:IsKLCC    Residuals
Sum of Squares         1.478750e+14 1.387268e+15                1.037874e+14 2.553896e+16
Deg. of Freedom                   2            1                           2        38211

Residual standard error: 817536.9
Estimated effects may be unbalanced
> summary(anova_result)
                             Df    Sum Sq   Mean Sq F value Pr(>F)
Property_Furnishing           2 1.479e+14 7.394e+13  110.62 <2e-16 ***
IsKLCC                        1 1.387e+15 1.387e+15 2075.61 <2e-16 ***
Property_Furnishing:IsKLCC    2 1.038e+14 5.189e+13   77.64 <2e-16 ***
Residuals                 38211 2.554e+16 6.684e+11
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> plot(anova_result)
```

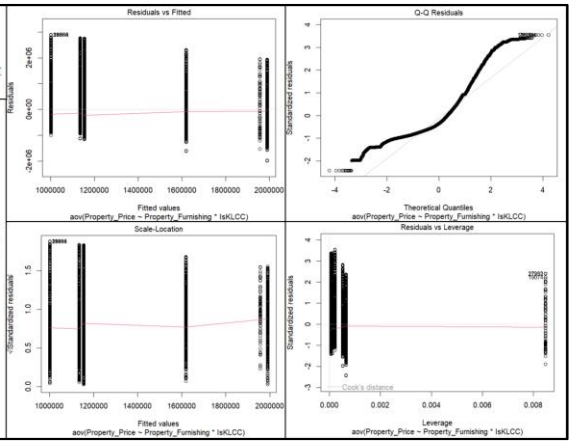
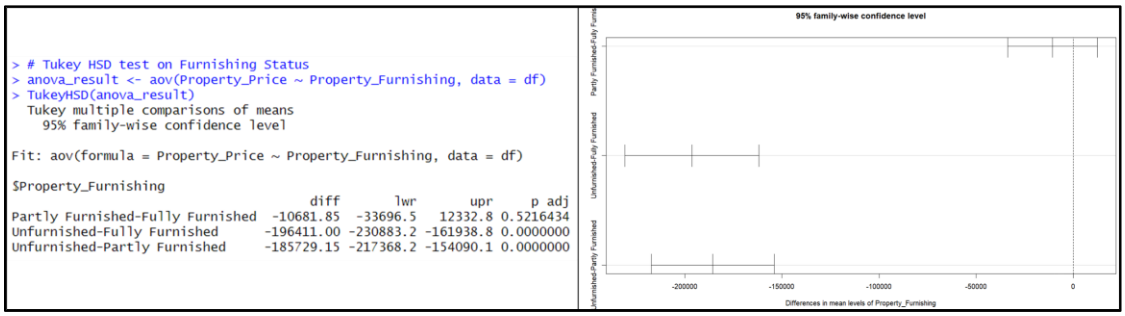


Figure 109-3.3.3.1: ANOVA testing for (Property_Price), (Property_Furnishing) & (IsKLCC)

The ANOVA test shows the independent variables Property_Furnishing, IsKLCC, and their interactions have the same p-values which are less than 2e-16 (Less than a significance level of 0.05), indicating both variables have a

significant impact on price, suggesting all the hypotheses could be rejected. The diagnostic plots have the residual mean line near 0, showing that no large outliers that would lead to research bias. This QQ plot shows there are deviations in both variables, slightly different from homoscedasticity.



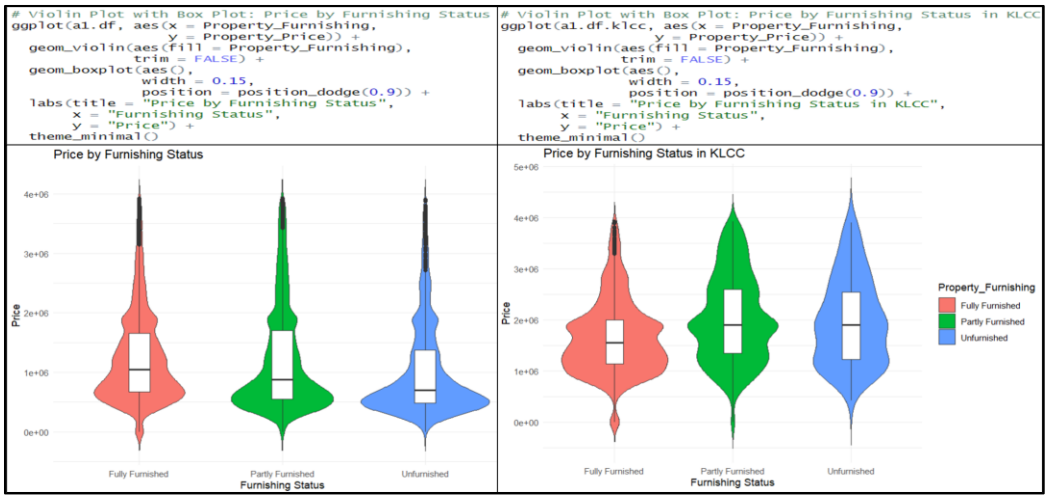
```
> # Tukey HSD test on Furnishing Status
> anova_result <- aov(Property_Price ~ Property_Furnishing, data = df)
> TukeyHSD(anova_result)
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = Property_Price ~ Property_Furnishing, data = df)

$Property_Furnishing
                                    diff       lwr       upr     p adj
Partly Furnished-Fully Furnished  -10681.85  -33696.5   12332.8 0.5216434
Unfurnished-Fully Furnished      -196411.00 -230883.2 -161938.8 0.0000000
Unfurnished-Partly Furnished     -185729.15 -217368.2 -154090.1 0.0000000
```


Figure 110-3.3.3.2: Tukey HSD Plot of (Property_Furnishing) to Imapct on Price

The p-value of partly vs. fully is 0.5216 (indicating no difference in mean price between partly and fully), while the p-value of unfurnished vs. fully and unfurnished vs. partly is 0.0000 (there are differences between unfurnished with fully and with partly on mean price).

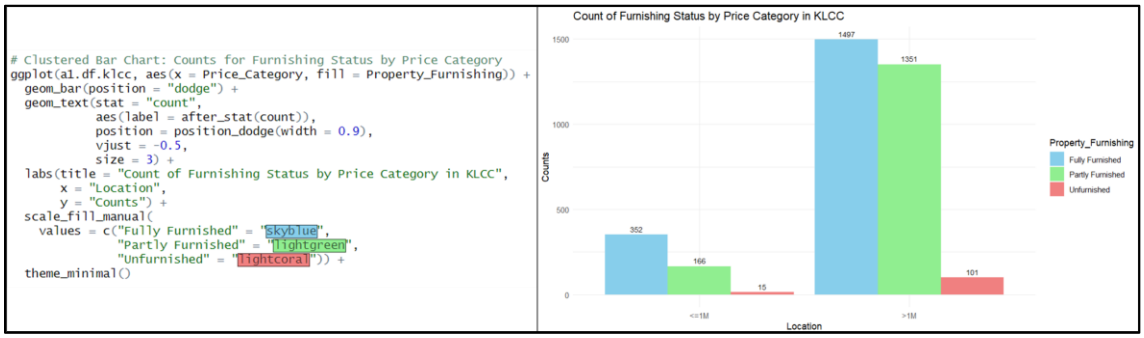


```
# Violin Plot with Box Plot: Price by Furnishing Status
ggplot(a1.df, aes(x = Property_Furnishing,
                  y = Property_Price)) +
  geom_violin(aes(fill = Property_Furnishing),
              trim = FALSE) +
  geom_boxplot(aes(),
               width = 0.15,
               position = position_dodge(0.9)) +
  labs(title = "Price by Furnishing Status",
       x = "Furnishing Status",
       y = "Price") +
  theme_minimal()
```

```
# Violin Plot with Box Plot: Price by Furnishing Status in KLCC
ggplot(a1.df.klcc, aes(x = Property_Furnishing,
                       y = Property_Price)) +
  geom_violin(aes(fill = Property_Furnishing),
              trim = FALSE) +
  geom_boxplot(aes(),
               width = 0.15,
               position = position_dodge(0.9)) +
  labs(title = "Price by Furnishing Status in KLCC",
       x = "Furnishing Status",
       y = "Price") +
  theme_minimal()
```


Figure 111-3.3.3.3: Violin and Box Plot of (Property_Price) by (Property_Furnishing)

Through the violin and box plot, the violin shapes fully and partly in all locations are slightly similar. Comparing the unfurnished to others, there is a difference in the density as observed in the violin shape is the result of ANOVA and Tukey test. While in KLCC, partly furnished and unfurnished properties have overall higher prices, but three categories have overall prices>1M, having less impact on price in hypothesis.



```
# Clustered Bar Chart: Counts for Furnishing Status by Price Category
ggplot(a1.df.klcc, aes(x = Price_Category, fill = Property_Furnishing)) +
  geom_bar(position = "dodge") +
  geom_text(stat = "count",
            aes(label = after_stat(count)),
            position = position_dodge(width = 0.9),
            vjust = -0.5,
            size = 3) +
  labs(title = "Count of Furnishing Status by Price Category in KLCC",
       x = "Location",
       y = "Counts") +
  scale_fill_manual(
    values = c("Fully Furnished" = "skyblue",
               "Partly Furnished" = "lightgreen",
               "Unfurnished" = "lightcoral")) +
  theme_minimal()
```


Figure 112-3.3.3.4: Clustered Bar Chart of (Property_Furnishing) by (IsKLCC)

There is a difference between locations and price, where the fully furnished has the highest property distribution in both category prices in KLCC, and (1497, 50.76%) in price >1M.

3.3.3 Analysis 3-2: How is the distribution of Furnishing status in KLCC and Price > 1M?

```
> table(a2.df$Size_Category, a2.df$Property_Furnishing,
+       a2.df$Room_Category)
, ,  = <=3

           Fully Furnished Partly Furnished Unfurnished
  <1000                362              239          6
  >1800                 52               72         10
  1000-1800            621              408         34

, ,  = >3

           Fully Furnished Partly Furnished Unfurnished
  <1000                  1                3          0
  >1800                342              541         37
  1000-1800            119               88         14
```

```
# Insert 3D table data into array
three_way_array <-
  array(c(362, 52, 621, 239, 72, 408, 6, 10, 34, 1,
          342, 119, 3, 541, 88, 0, 37, 14),
        dim = c(3, 3, 2),
        dimnames = list(Size_Category =
                          c("<1000", ">1800", "1000-1800"),
                        Property_Furnishing =
                          c("Fully Furnished",
                            "Partly Furnished", "Unfurnished"),
                        Room_Category = c("<=3", ">3")))

# Convert to a three-way contingency table
three_way_table <- as.table(three_way_array)

# Mosaic Plot: Size by Furnishing by Room
mosaic(three_way_table, shade = TRUE, legend = TRUE)
```

Figure 113-3.3.3.5: Mosaic Plot of (Property Furnishing), (Size_Category), & (Room_Category)

The mosaic plot shows the properties in 3 dimensions. Through observation, the fully furnished and partly furnished seem to have similar large areas. Further analysis is conducted to determine which furnishing status with most impact on the hypothesis.

```
# Get Room <=3
a3.df <- a2.df %>% filter(Room_Category == "<=3")

# Get table data for Size Category and Furnishing
summary_data <- as.data.frame(
  table(a3.df$Property_Furnishing, a3.df$Size_Category))
colnames(summary_data) <-
  c("Property_Furnishing", "Size_Category", "Counts")
summary_data$Size_Category <-
  factor(summary_data$Size_Category,
         levels = c("<1000", "1000-1800", ">1800"))
summary_data$Furnishing_Size <-
  paste(summary_data$Property_Furnishing,
        summary_data$Size_Category, sep = " : ")

# Lollipop Chart: Counts for Furnishing Status by
#                 Size Category with Rooms <=3, Price >1M in KLCC
ggdotchart(summary_data, x = "Furnishing_Size", y = "Counts",
           title = "Counts for Furnishing Status by
              Size Category with Rooms <=3, Price >1M in KLCC",
           color = "Size_Category",
           palette = c("#00AFBB", "#E7B800", "#FC4E07"),
           sorting = "asc", sort.by.groups = TRUE,
           group = "Size_Category",
           add = "segments",
           add.params = list(color = "lightgray", size = 3),
           dot.size = 5,
           ggtheme = theme_pubclean()) +
  font("x.text", size = 8, vjust = 0.5) +
  labs(x="Furnishing Status") +
  geom_text(aes(label = Counts),
            position = position_dodge(width = 0.8),
            vjust = -0.7)
```

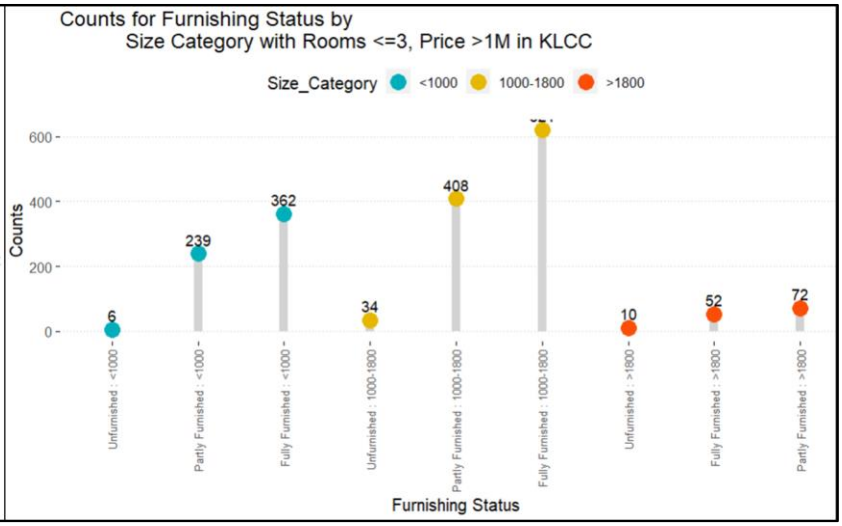


Figure 114-3.3.3.6: Lollipop Chart of Counts for (Property_Furnishing) by (Size_Category)

The lollipop chart sorts the counts for furnishing status by size. Although the partly furnished properties are the most in the size >1800 sqft, the counts are too small to impact on hypothesis. While within the size 1000-1800 sqft, the fully furnished properties have the most records (58.42%, 621/1063), indicating a larger chance of impact on the hypothesis to fulfill the criteria of price >1M and KLCC.

3.3.4 Analysis 4-1: Is there a difference between Built_Area with Location and Price?

```
# Hypothesis 0: No difference in Built_Area and IsKLCC
# Hypothesis 1: Difference in Built_Area and IsKLCC

contingency_table <- table(df$Built_Area, df$IsKLCC)
contingency_table

           KLCC Non-KLCC
Built-up   3439    25168
Land area   184     9426
```

```
> # Perform Chi-square test
> chi_square_result <- chisq.test(contingency_table)
> chi_square_result

        Pearson's Chi-squared test with Yates' continuity correction

data:  contingency_table
X-squared = 855.1, df = 1, p-value < 2.2e-16
```

Figure 115-3.3.4.1: Chi-square Test of (Built_Area) & (IsKLCC)

The p-value of the Chi-Square test is 2.2e-16, which is less than the significance level of 0.05, suggesting a significant association between Built_Area and IsKLCC. The null hypothesis is rejected.

```
# Stacked Bar Chart of Count for Built Area by IsKLCC
ggplot(df, aes(x = Built_Area,
               fill = IsKLCC)) +
  geom_bar(position = "stack") +
  geom_text(stat = "count",
            aes(label = stat(count)),
            position = position_stack(vjust = 0.5)) +
  labs(title = "Count for Built Area by IsKLCC",
       x = "Built Area",
       y = "Count") +
  theme_minimal()
```

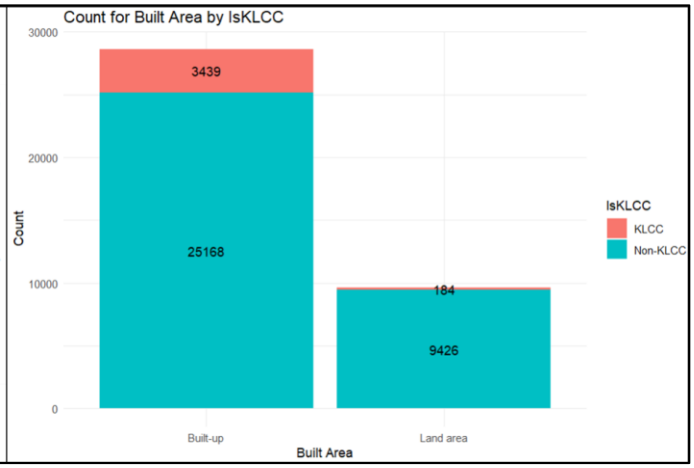


Figure 116-3.3.4.2: Stacked Bar Chart of Counts for (Built_Area) by (IsKLCC)

There is a difference between built area and location. In KLCC, only 184 (5.08%) properties are land area and 3439 (94.92%) properties are built-up, which has a higher ratio compared to built-up properties in non_KLCC.

```
# Conduct t.test for Built Area & Price
t.test(Property_Price ~ Built_Area, data = df)
```

```
# Hypothesis 0: No difference in Price between Built-Up & Land
# Hypothesis 1: Difference in Price between Built-Up & Land
```

```
        Welch Two Sample t-test

data:  Property_Price by Built_Area
t = -34.781, df = 14459, p-value < 2.2e-16
alternative hypothesis: true difference in means between
group Built-up and group Land area is not equal to 0
95 percent confidence interval:
 -390353.8 -348702.8
sample estimates:
 mean in group Built-up mean in group Land area
                1098791                 1468320
```

Figure 117-3.3.4.3: T-Test for (Built_Area) & (Price_Category)

The T-Test is applied to Built_Area and price, showing the p-value of 2.2e-16, which is hugely less than the significance level of 0.05, suggesting that there is a significant association between Built_Area and price. The null hypothesis is rejected.

```
mean_values <- a1.df.klcc %>%
  group_by(Built_Area) %>%
  summarize(mean_price =
              mean(Property_Price, na.rm = TRUE))

# Box Plot with Mean for price of Built Area in KLCC
ggplot(a1.df.klcc, aes(x = Built_Area,
                       y = Property_Price)) +
  geom_boxplot(notch = TRUE,
               fill = "yellow") +
  geom_point(data = mean_values,
             aes(x = Built_Area,
                 y = mean_price),
             shape = 18,
             size = 6,
             color = "#FC4E07") +
  labs(title = "Property Prices by Built Area in KLCC",
       x = "Built Area",
       y = "Property Price") +
  theme_minimal()
```

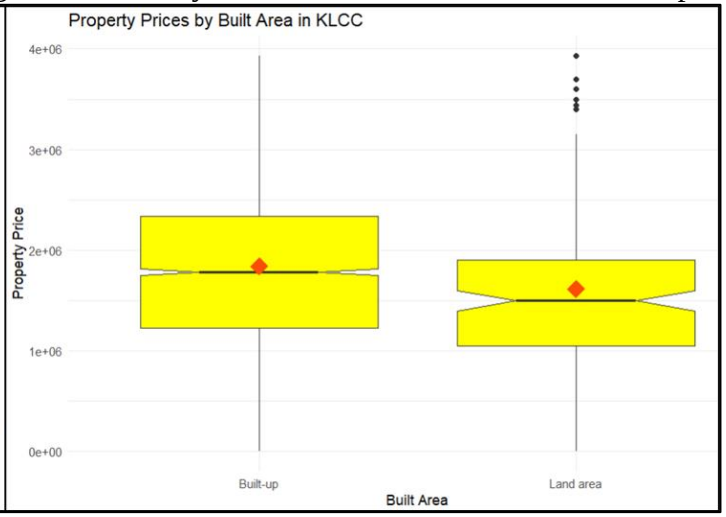
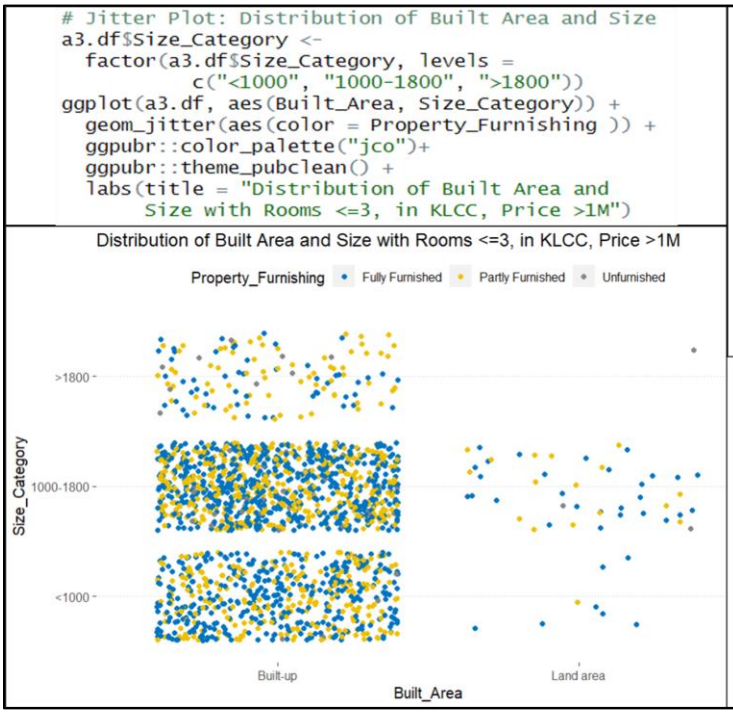


Figure 118-3.3.4.4: Box Plot with Mean for (Property_Price) of (Built_Area) in KLCC

The box plot shows the difference between the price of built-up and land area in KLCC. The built-up properties in KLCC have a higher price compared to the land area in all quartiles and mean. However, 1st quartile (75% of records) of both built areas has a price >1M, having less impact on the hypothesis.

3.3.4 Analysis 4-2: What is the distribution of built area?

```
# Jitter Plot: Distribution of Built Area and Size
a3.df$Size_Category <-
  factor(a3.df$Size_Category, levels =
           c("<1000", "1000-1800", ">1800"))
ggplot(a3.df, aes(Built_Area, Size_Category)) +
  geom_jitter(aes(color = Property_Furnishing )) +
  ggpubr::color_palette("jco")+
  ggpubr::theme_pubclean() +
  labs(title = "Distribution of Built Area and
       Size with Rooms <=3, in KLCC, Price >1M")
```




```
type_count <- table(a2.df$Built_Area)
percentage_counts <- round((type_count / sum(type_count)) * 100, 2)
percentage_counts_labels <- paste0(percentage_counts, "%", "\n", paste0(type_count))

# Pie Chart: Built Area in KLCC and Price >1M
pie(type_count, labels = percentage_counts_labels,
    main = "Built Area in whole dataset", radius = 0.9,
    col = c("skyblue","lightcoral"), clockwise = TRUE)


a4.df = a3.df %>%
  filter(Size_Category == "1000-1800" & Property_Furnishing == "Fully Furnished")

type_count <- table(a4.df$Built_Area)
percentage_counts <- round((type_count / sum(type_count)) * 100, 2)
percentage_counts_labels <- paste0(percentage_counts, "%", "\n", paste0(type_count))

# Pie Chart: Built Area with Room <=3, 1000-1800 sqft,
#          Fully Furnished, Price>1M in KLCC
pie(type_count, labels = percentage_counts_labels,
    main = "Built Area with Room <=3, 1000-1800sqft,
    Fully FUrnished, Price>1M in KLCC", radius = 0.9,
    col = c("skyblue","lightcoral"), clockwise = TRUE)
legend("topright", legend = names(type_count), fill = c("skyblue","lightcoral"),
       title = "Built Area", cex = 1)
```

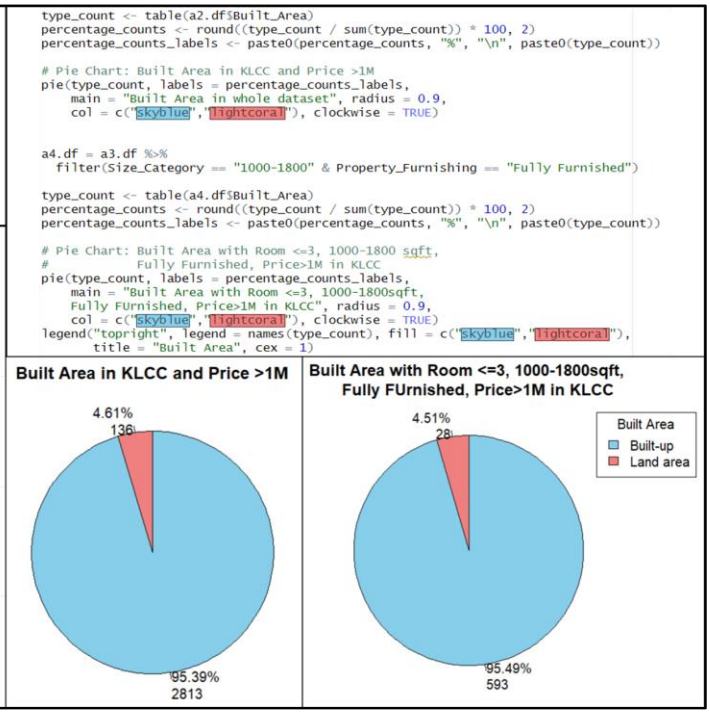


Figure 119-3.3.4.5: Jitter Plot of (Built_Area) by (Size_Category) & Pie Chart for Distribution of (Built_Area) in KLCC with Price > 1M

There are huge differences in the distribution of built-up and land-area properties with more than 95% of properties being built-up in KLCC. Therefore, the built-up properties have a higher impact on the hypothesis.

3.3.5 Conclusion: Validation of Hypothesis

Indep. / Dep.	Location → KLCC in >1M	Price → >1M in KLCC	Higher Impact In
Rooms_Category	<=3 has a higher proportion (61.17%)	<=3 & >3 have a mean price >1M	Location
Size_Category	Most sizes in 1000-1800 (43.54%)	↑ size, ↑ price	Both
Property_Furnishing	Fully Furnished has the most proportion (50.76%)	All categories have an overall price >1M	Location
Built_Area	The built-up proportion is hugely higher (95.39%)	All categories have an overall price >1M	Location

Table 1-3.3.5.1: The impact of independent variables on the dependent variable in hypothesis

The table concludes the impact and answers the question for the hypothesis:

1. If the properties are in KLCC, what category of each variable is more likely to appear?
2. If the property's price >1M, what category of each variable is more likely to appear?
3. If the properties are in KLCC and price >1M, what category of each variable is more likely to appear?

```
# Determine the area, intersection
venn <- a2.df %>%
  select(Room_Category, Size_Category, Property_Furnishing, Built_Area)
area1 = sum(venn$Room_Category=="<=3")
area2 = sum(venn$Size_Category=="1000-1800")
area3 = sum(venn$Property_Furnishing=="Fully Furnished")
area4 = sum(venn$Built_Area=="Built-up")
n12 = sum(venn$Room_Category=="<=3"&venn$Size_Category=="1000-1800")
n13 = sum(venn$Room_Category=="<=3"&venn$Property_Furnishing=="Fully Furnished")
n14 = sum(venn$Room_Category=="<=3"&venn$Built_Area=="Built-up")
n23 = sum(venn$Size_Category=="1000-1800"&venn$Property_Furnishing=="Fully Furnished")
n24 = sum(venn$Size_Category=="1000-1800"&venn$Built_Area=="Built-up")
n34 = sum(venn$Property_Furnishing=="Fully Furnished"&venn$Built_Area=="Built-up")
n123 = sum(venn$Room_Category=="<=3"&venn$Size_Category=="1000-1800"
           &venn$Property_Furnishing=="Fully Furnished")
n124 = sum(venn$Room_Category=="<=3"&venn$Size_Category=="1000-1800"
           &venn$Built_Area=="Built-up")
n134 = sum(venn$Room_Category=="<=3"&venn$Property_Furnishing=="Fully Furnished"
           &venn$Built_Area=="Built-up")
n234 = sum(venn$Size_Category=="1000-1800"&venn$Property_Furnishing=="Fully Furnished"
           &venn$Built_Area=="Built-up")
n1234 = sum(venn$Room_Category=="<=3"&venn$Size_Category=="1000-1800"
            &venn$Property_Furnishing=="Fully Furnished"&venn$Built_Area=="Built-up")
left = sum(venn$Room_Category!="<=3"&venn$Size_Category!="1000-1800"
           &venn$Property_Furnishing!="Fully Furnished"&venn$Built_Area!="Built-up")
# 4 Set Venn Diagram
grid.newpage()
draw.quad.venn(area1 = area1, area2 = area2, area3 = area3, area4 = area4, n12 = n12,
               n13 = n13, n14 = n14, n23 = n23, n24 = n24, n34 = n34,
               n123 = n123, n124 = n124, n134 = n134, n234 = n234, n1234 = n1234,
               fill = c("#007AC2FF", "#EFC000FF", "#868686FF", "#CD534CFF"),
               lty = "blank",
               category =
                 c("Rooms <=3", "Size 1000-1800", "Fully Furnished", "Built-up"))
grid.text(label = "Number of Properties in KLCC with Price >1M", x = 0.5, y = 0.98,
          just = "top", gp = gpar(fontsize = 16, fontface = "bold"))
grid.text(label = left, x = 0.17, y = 0.22, just = "top")
grid.text(label = paste0("Total: ", nrow(venn)), x = 0.5, y = 0.06, just = "bottom")
```

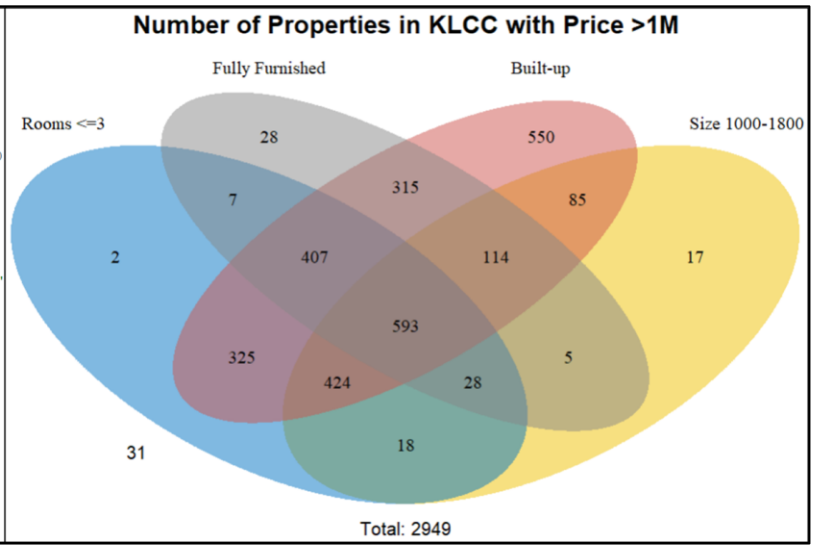


Figure 120-3.3.5.2: Venn Diagram of (Room <=3), (1000-1800 sqft), (Fully Furnished) & (Built-up)

```
> round(593/2949*100, 2)
[1] 20.11
```

Figure 121-3.3.5.3: Percentage

The Venn diagram shows that in KLCC with a price >1M, most (20.11%, 593/2949) properties are **rooms <=3, size 1000-1800 sqft, fully furnished, and built-up**. It is also observed that most of the properties are built-up (95.39%, 2813/2949), having the highest impact compared to other independent variables, followed by rooms <=3 (61.17%, 1804/2949), fully furnished (50.76%, 1497/2949), and size 1000-1800 sqft (43.54%, 1284/2949). In conclusion, the hypothesis is incorrect regarding the room category variables.

3.3.6 Additional Features

Progress	Features
Analysis 1-1	• **wilcox.test():** Compare the distribution of two paired or unpaired groups, especially when the data may not follow a normal distribution. • **geom_col():** Creates bar plots to compare the values of different categories. • **coord_polar():** Converts Cartesian coordinates to polar coordinates to enable the creation of circular plots or pie charts. • **labs():** Customizes plot labels, improving the interpretability in visualization
Analysis 1-2	• **cor.test():** Conducts correlation tests to assess the strength and significance of relationships between variables. • **tapply():** Applies function over subsets to facilitate data summaries. • **geom_bar(position = "fill"):** Creates a stacked bar plot with proportions to provide a clear view of the relative frequencies. **geom_errorbar():** Error bars to a plot to visualize uncertainty or variability.
Analysis 1-3	• **geom_treemap():** Creates a treemap plot, an effective way to display hierarchical and nested data structures.
Analysis 2-1	• **cor(method="spearman"):** Correlation robust to non-linear relationships. • **corrplot():** Plot correlation to provide a visual representation of matrices. • **t.test():** Conducts t-tests to assess differences in means between groups. • **ggscatterhist():** Creates scatter plots with marginal histograms to give insights into the univariate distribution of variables. • **geom_smooth():** Draw a smoothed line to identify trends or patterns in data.
Analysis 2-2	• **geom_density():** Kernel density for representation of the distribution. • **geom_vline(): V**ertical lines to highlight specific points or thresholds. • **skewness():** Calculates skewness, providing insights into the asymmetry of a distribution. • **kurtosis():** Calculates kurtosis, measuring the "tailedness" of a distribution. • **legend():** Adds or modifies legends to improve the clarity of plots.
Analysis 3-1	• **aov():** Performs analysis of variance to compare means in multiple groups. • **TukeyHSD():** Conducts Tukey's honestly significant difference test for post-hoc analysis after ANOVA. • **geom_violin():** Violin plot for insights into the distribution of data across different categories. • **geom_bar(position = "dodge"):** Side-by-side bar plots for comparison
Analysis 3-2	• **mosaic():** Mosaic plot for the joint distribution of categorical variables. • **ggdotchart():** Creates a dot chart, to display lollipop chart
Analysis 4-1	• **chisq.test():** Chi-square test for association between categorical variables. • **geom_bar(position="stack"):** Stacked bar plot for the cumulative counts
Analysis 4-2	• **geom_jitter():** Jitter plot to provide a clearer view of individual data points.
Conclusion	• **draw.quad.venn():** Venn diagram to see the overlap between multiple sets. • **grid.text():** Adds text to a grid plot to facilitate annotation of complex visualizations.

3.4 Objective 4: To determine the impact of property size (sq. ft.) on property price in KLCC (Lim Wen Hann TP065443)

3.4.1 Analysis 4-1: What is the Overall Relationship between Property Size and Price in the Whole Dataset?

```
#Linear Regression Analysis
Filtered_analysis <- data.frame(Filtered_dataset$Property_Siz
                                Filtered_dataset$Property_Pri
names(Filtered_analysis) <- c("Property_Size","Property_Price

Filtered_analysis_model <- lm(Property_Price ~ Property_Size,
                              data = Filtered_analysis)
Filtered_analysis_model

summary(Filtered_analysis_model)
```

```
> summary(Filtered_analysis_model)

Call:
lm(formula = Property_Price ~ Property_Size, data = Filtered_analysis)

Residuals:
     Min       1Q   Median       3Q      Max
-2881664  -342896  -124074   243809  3021355

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   40891.963   6456.856   6.333 2.43e-10 ***
Property_Size   711.068      3.531 201.366  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 587400 on 38215 degrees of freedom
Multiple R-squared:  0.5148,    Adjusted R-squared:  0.5148
F-statistic: 4.055e+04 on 1 and 38215 DF,  p-value: < 2.2e-16
```

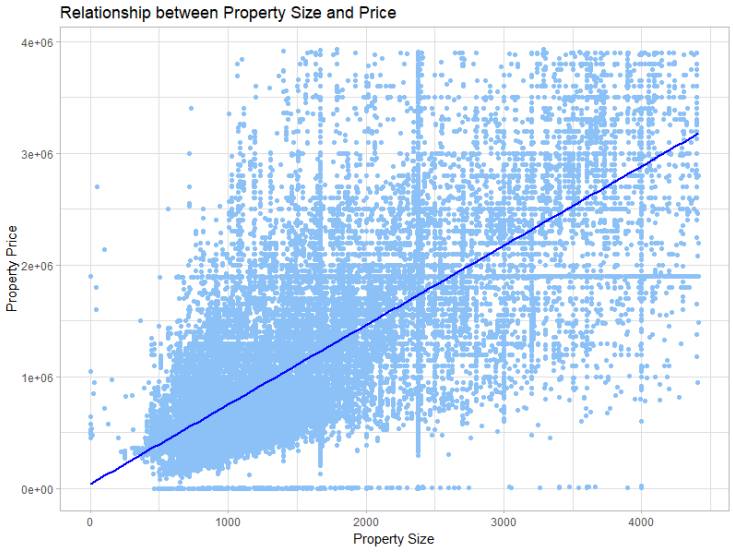


Figure 122-3.4.1.1 Linear Regression Analysis for Property Size & Price in Whole Dataset

This linear regression analysis shows the relationship between property size and price of all residential properties in Kuala Lumpur. It reveals a significant association between both the variables for the entire dataset. The coefficient for property size is RM 711.06, indicating that on average, for each unit square feet increase in property size, its corresponding price will be increased by RM 711.06. There is also an extremely low p-value ($p < 2.2e-16$), showing that the model is statistically significant. The adjusted R-squared value of 0.5148 suggests that approximately 51.48% of variability in price is caused by property size.

3.4.2 Analysis 4-2: What is the Distribution of Size of Residential Properties in Kuala Lumpur?

```
#Create Histogram to view frequencies of different property sizes

whole_dataset_histogram <-
  ggplot(Filtered_dataset ,aes(x = Property_Size)) +
  geom_histogram(binwidth = 150, colour ="black",aes(fill=..count..)) +
  scale_fill_gradient("Count", low="#8BC1F7",high = "#0066CC") +
  theme_light() +
  labs(x = "Property Size (sq ft.)", y = "Number of Properties",
       title="Distribution of Property Sizes in Whole Dataset\n") +
  theme(plot.title = element_text(hjust = 0.5,
                                  margin = margin(b = -0.1, t = 0.3, unit = "cm")),
        legend.position="none",
        axis.title.x = element_text(margin = margin(t= 0.3, b = 0.3, unit = "cm")),
        axis.title.y = element_text(margin = margin(l = 0.1, r = 0.2, unit = "cm")))
whole_dataset_histogram
```

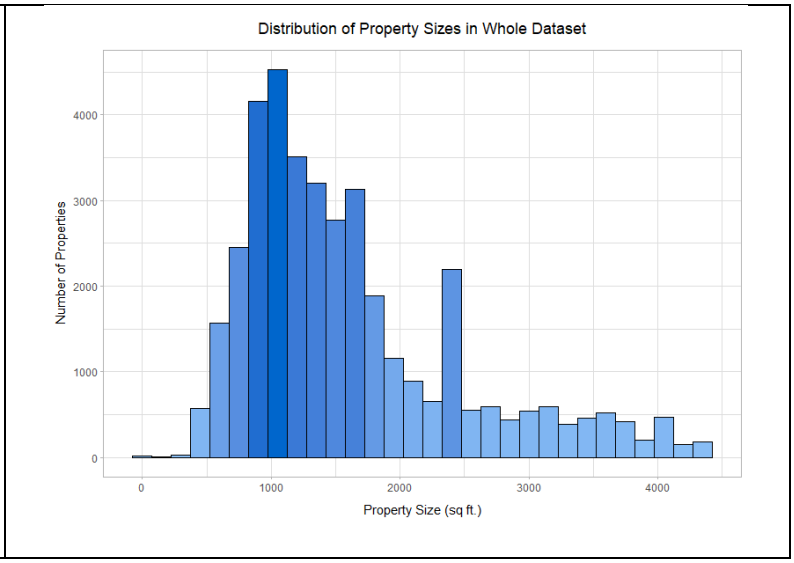
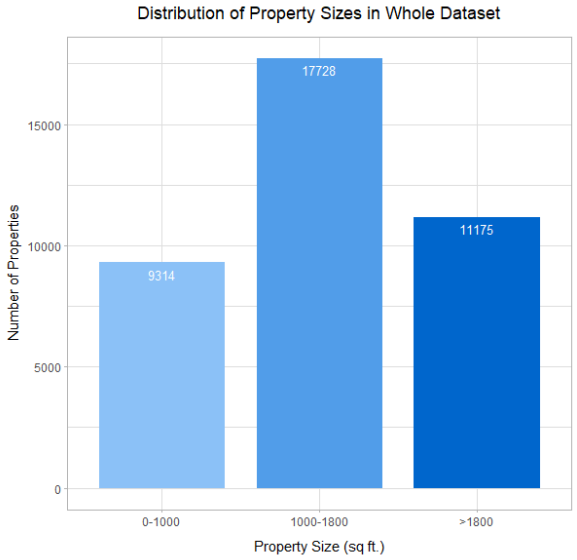
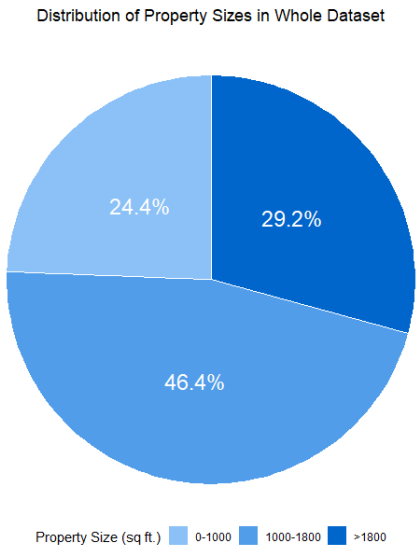


Figure 123-3.4.2.1 Frequency Distribution of Property Sizes in Whole Dataset

The above histogram depicts the frequency distribution of property sizes for residential properties in the entire Kuala Lumpur dataset. This histogram shows a right-skewed distribution, where the majority of residential properties are around 1000 square feet in size and being the most common property size. There is a slight peak around the 2500 square feet range due to the outliers being changed to the mean property size during the data validation process. To facilitate analysis, the data has been divided into three different categories, each with its corresponding range: 0 – 1000 sq ft, 1000 – 1800 sq ft, and > 1800 sq ft.






```
#Create Pie and Bar Charts for Distribution of Property Sizes in Whole Dataset
ggplot(whole_dataset_pie, aes(x="", y=y_axis, fill=x_axis)) +
  geom_bar(stat="identity", width=1, color="white") +
  coord_polar("y", start=0) +
  theme_void() +
  theme(plot.title = element_text(hjust = 0.5, margin = margin(b = -0.5, t = 0.3, unit = "cm")),
        legend.position="bottom",
        legend.margin = margin(t = -0.5, b = 0.3, unit = "cm")) +
  labs(fill="Property Size (sq ft.)", title="Distribution of Property Sizes in Whole Dataset\n") +
  geom_text(aes(label = percentage),position = position_stack(vjust = 0.5),
            color = "white", size=6) +
  scale_fill_manual(values=c("#88C1F7","#519DE9","#0066CC"))
```

```
whole_dataset_barchart<-
  ggplot(whole_dataset_pie, aes(x=x_axis,y=y_axis, fill = factor(y_axis))) +
  geom_bar(stat="identity", width = 0.8) +
  geom_text(aes(label = y_axis), vjust=1.6, color="white", size=3.5) +
  labs(x = "Property Size (sq ft.)", y = "Number of Properties",
       title="Distribution of Property Sizes in Whole Dataset\n") +
  theme_light() +
  theme(plot.title = element_text(hjust = 0.5, margin = margin(b = -0.1, t = 0.3, unit = "cm")),
        legend.position="none",
        axis.title.x = element_text(margin = margin(t= 0.3, b = 0.3, unit = "cm")),
        axis.title.y = element_text(margin = margin(l = 0.1, r = 0.2, unit = "cm"))) +
  scale_fill_manual(values=c("#88C1F7","#0066CC","#519DE9"))
whole_dataset_barchart
```

Figure 124-3.4.2.2 Bar Chart & Pie Chart for Distribution of Property Sizes in Whole Dataset

The above figures illustrate the distribution of property sizes in Kuala Lumpur after dividing property size into three different categories in both a bar and pie chart as well as the R syntax used. In Kuala Lumpur, there are a total of 17728 residential properties falling within the size range of 1000 – 1800 sq ft., accounting for 46.4% of the total properties surveyed. The next category, consisting of properties larger than 1800 sq ft, represents 29.2% of the total, with a total count of 11,175. The smallest category, with properties ranging from 0 to 1000 sq ft, accounts for 24.4% of the total, consisting of only 9314 properties.

3.4.3 Analysis 4-3: What is the Distribution of Size of Residential Properties in KLCC, Kuala Lumpur?

```
KLCC_dataset_lolipop<-
  ggplot(KLCC_dataset_pie,aes(x=x_axis,y_axis)) +
  geom_segment( aes(x=x_axis, xend=x_axis, y=0, yend=y_axis), color = "black",
                size = 1, alpha = 0.7) +
  geom_point(size = 10, color = KLCC_dataset_pie$custom_color) +
  geom_label(aes(x_axis, y_axis , label = signif(y_axis)),  colour = "black",
             nudge_x = 0.35, size = 4) +
  labs(x = "Property Size (sq ft.)", y = "Number of Properties",
       title="Distribution of Property Sizes in KLCC\n") +
  coord_flip() +
  theme_light() +
  theme(panel.grid.major.x = element_blank(), axis.ticks.x = element_blank(),
        plot.title = element_text(hjust = 0.5,
                                  margin = margin(b = -0.1, t = 0.3, unit = "cm")),
        legend.position="none",
        axis.title.x = element_text(margin = margin(t= 0.3, b = 0.3, unit = "cm")),
        axis.title.y = element_text(margin = margin(l = 0.1, r = 0.2, unit = "cm"))) +
  ylim(c(0, max(KLCC_dataset_pie$y_axis) * 1.2))
KLCC_dataset_lolipop
```

```
KLCC_dataset_barchart<-
  ggplot(KLCC_dataset_pie, aes(x=x_axis,y=y_axis, fill = factor(y_axis))) +
  geom_bar(stat="identity", width = 0.8) +
  geom_text(aes(label = y_axis), vjust=1.6, color="white", size=3.5) +
  labs(x = "Property Size (sq ft.)", y = "Number of Properties",
       title="Distribution of Property Sizes in KLCC\n") +
  theme_light() +
  theme(plot.title = element_text(hjust = 0.5,
                                  margin = margin(b = -0.1, t = 0.3, unit = "cm")),
        legend.position="none",
        axis.title.x = element_text(margin = margin(t= 0.3, b = 0.3, unit = "cm")),
        axis.title.y = element_text(margin = margin(l = 0.1, r = 0.2, unit = "cm"))) +
  scale_fill_manual(values=c("#009596","#A2D9D9","#73C5C5"))
KLCC_dataset_barchart
```

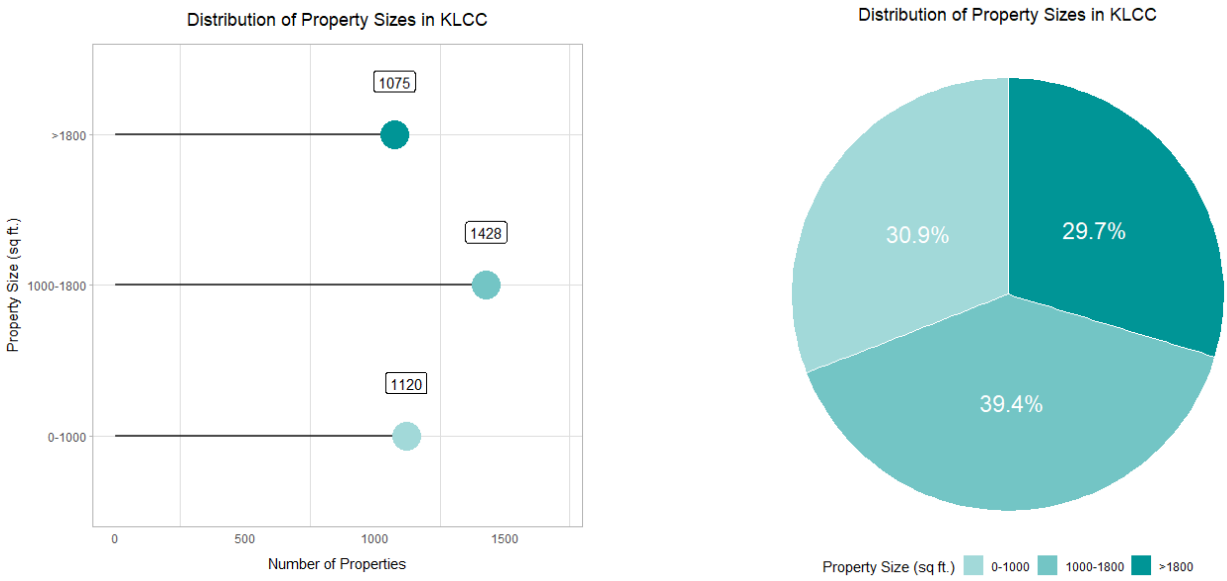
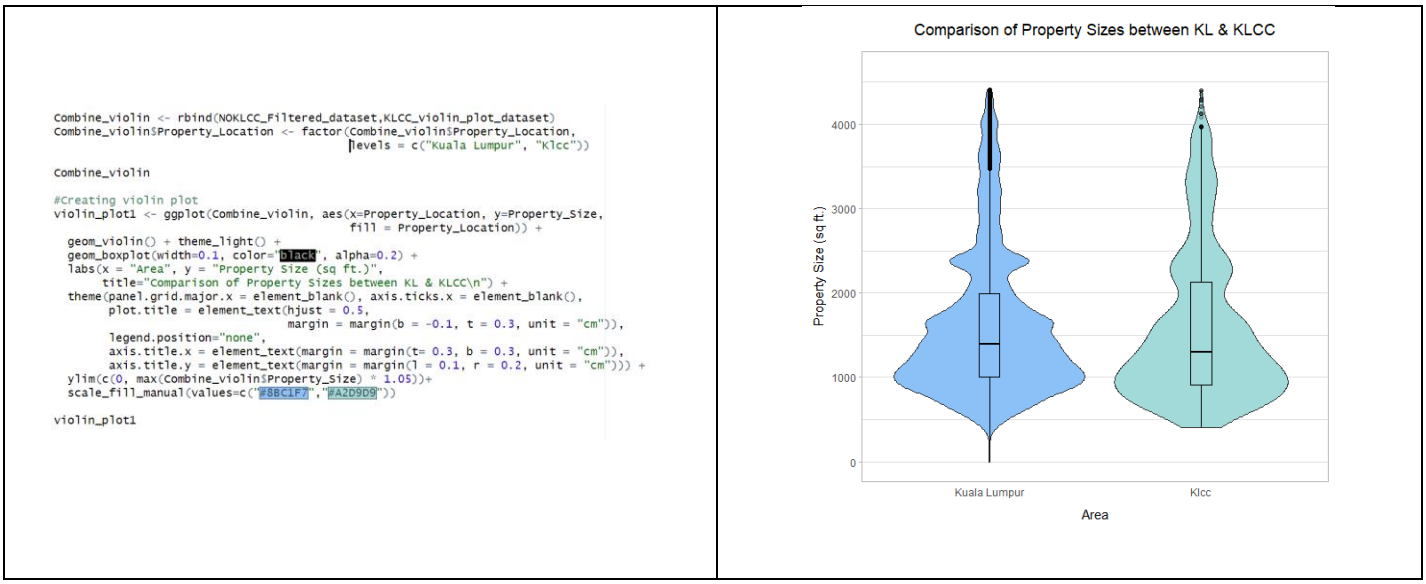


Figure 125- 3.4.3.1 Lollipop Plot & Pie Chart for Distribution of Property Sizes in KLCC

In this section, a horizontal lollipop chart is utilized in place of the bar chart to replace the bar chart to introduce more variety in the data visualizations. As we focus our analysis specifically on the KLCC region, we observe a trend towards more balanced percentages in the distribution of property sizes. The 1000-1800 sq. ft. range maintains its lead with 39.4%, representing 1428 residential properties in KLCC. However, the other size ranges are closely trailing behind, with 30.9% (1120 properties) for the 0-1000 sq ft. category and 29.7% (1075 properties) for the >1800 sq ft. category.



```
Combine_violin <- rbind(NOKLCC_Filtered_dataset,KLCC_violin_plot_dataset)
Combine_violin$Property_Location <- factor(Combine_violin$Property_Location,
                                           levels = c("Kuala Lumpur", "Klcc"))

Combine_violin

#Creating violin plot
violin_plot1 <- ggplot(Combine_violin, aes(x=Property_Location, y=Property_Size,
                                           fill = Property_Location)) +
  geom_violin() + theme_light() +
  geom_boxplot(width=0.1, color="black", alpha=0.2) +
  labs(x = "Area", y = "Property Size (sq ft.)",
       title="Comparison of Property Sizes between KL & KLCC\n") +
  theme(panel.grid.major.x = element_blank(), axis.ticks.x = element_blank(),
        plot.title = element_text(hjust = 0.5,
                                  margin = margin(b = -0.1, t = 0.3, unit = "cm")),
        legend.position="none",
        axis.title.x = element_text(margin = margin(t= 0.3, b = 0.3, unit = "cm")),
        axis.title.y = element_text(margin = margin(l = 0.1, r = 0.2, unit = "cm"))) +
  ylim(c(0, max(Combine_violin$Property_Size) * 1.05))+
  scale_fill_manual(values=c("#8BC1F7","#A2D9D9"))

violin_plot1
```


Figure 126-3.4.3.2 Violin Plot with Boxplot for Comparison of Property Sizes between Kuala Lumpur & KLCC

When comparing the distribution of property sizes in KLCC and Kuala Lumpur using a violin plot with boxplot, it is obvious that there is less variation in property sizes within the KLCC region compared to the broader Kuala Lumpur dataset. The boxplot indicated that there are more extreme values in the Kuala Lumpur dataset compared to KLCC. We can hypothesize that the residential properties in KLCC are more evenly distributed across different size ranges.

3.4.4 Analysis-4.4: What is the Relationship between Property Size & Price in KLCC?

```
ggplot(KLCC_dataset_cor,aes(x = Property_Size, y = Property_Price))+
  geom_point(shape = 21, fill = '#A2D9D9', color = 'white',
             size = 3, alpha = 0.8)+
  sm_statCorr(color = '#009596', corr_method = 'pearson',
              linetype = 'dashed') +
  labs(x = "Property Size (sq ft.)", y = "Price",
       title = "Scatter Plot with Pearson correlation coefficient")+
  theme_light()

correlation_result <- cor.test(KLCC_dataset_cor$Property_Size,
                               KLCC_dataset_cor$Property_Price)
```

```
        Pearson's product-moment correlation

data:  KLCC_dataset_cor$Property_Size and KLCC_dataset_cor$Property_Price
t = 48.293, df = 3621, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.6056860 0.6453163
sample estimates:
      cor
0.625905
```

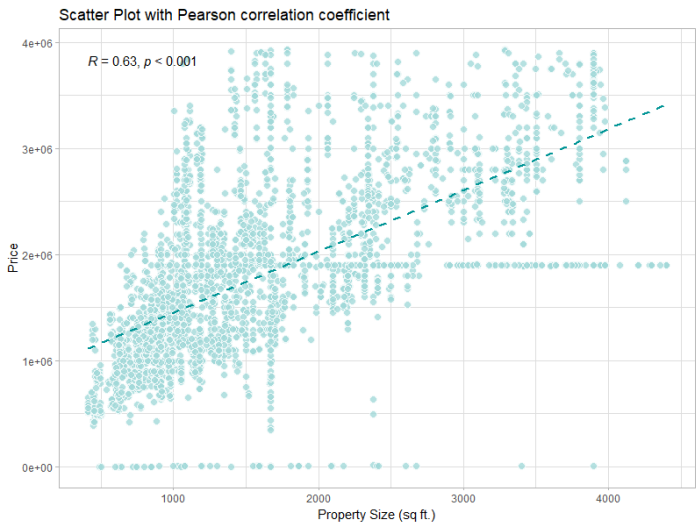


Figure 127-3.4.4.1 Scatter Plot with Pearson correlation coefficient for Property Size & Price in KLCC

According to the scatter plot above, the Pearson correlation coefficient (R) is 0.63, indicating a moderate positive linear relationship between the size and price of properties in KLCC. The statistical significance of this correlation was confirmed by the t-test, which yielded a t-value of 48.293 and a p-value of less than 2.2e-16, suggesting that the observed correlation is highly unlikely to occur naturally. With these results, the null hypothesis can be rejected in favor of the alternative hypothesis, as there is significant correlation between property size and property price in the KLCC dataset.

ANOVA Test

```
> KLCC_dataset_ANOVA_result <- aov(Property_Price ~ Property_Size_Range,
> summary(KLCC_dataset_ANOVA_result)
                     Df    Sum Sq   Mean Sq F value Pr(>F)
Property_Size_Range    2 9.623e+14 4.812e+14    1179 <2e-16 ***
Residuals           3620 1.478e+15 4.082e+11
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Tukey’s HSD Test

```
> posthoc <- glht(KLCC_dataset_ANOVA_result, linfct = mcp(Property_Size_Range = "Tukey"))
> summary(posthoc)

        Simultaneous Tests for General Linear Hypotheses

Multiple Comparisons of Means: Tukey Contrasts


Fit: aov(formula = Property_Price ~ Property_Size_Range, data = KLCC_dataset_ANOVA)

Linear Hypotheses:
                          Estimate Std. Error t value Pr(>|t|)
1000-1800 - 0-1000 == 0   735544      25500   28.84   <2e-16 ***
>1800 - 0-1000 == 0      1319808      27279   48.38   <2e-16 ***
>1800 - 1000-1800 == 0    584263      25798   22.65   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
(Adjusted p values reported -- single-step method)
```

Figure 128-3.4.4.2 ANOVA and Tukey Test for Property Price Data Grouped by Size in KLCC

After dividing the property sizes into three different categories, the ANOVA test can be used to determine whether there are any major differences in mean prices across the size ranges. There is a significant effect of property size range on the price ($F(2, 3620) = 1179$, $p < 0.001$). As for the Tukey’s test, big differences in property prices were found across the size ranges: 0-1000 square feet compared to 1000-1800 square feet ($t(3620) = 28.84$, $p < 2e\text{-}16$), 0-1000 square feet compared to above 1800 square feet ($t(3620) = 48.38$, $p < 2e\text{-}16$), and above 1800 square feet compared to 1000-1800 square feet ($t(3620) = 22.65$, $p < 2e\text{-}16$). The results of this Tukey test can allow us to reject the null hypothesis. The alternative hypothesis is accepted, where there are indeed significant differences in property prices among the different size ranges in the KLCC dataset.

```
ggplot(KLCC_dataset_ANOVA, aes(x = Property_Size_Range, y = Property_Price,
                               fill = Property_Size_Range)) +
  geom_boxplot() +
  labs(x = "Property Size Range", y = "Property Price",
       title = "Box Plot of Property Prices in KLCC by Size Range") +
  theme_light() +
  theme(plot.title = element_text(hjust = 0.5,
                                  margin = margin(b = 0.2, t = 0.3, unit = "cm")),
        legend.position="bottom",
        axis.title.x = element_text(margin = margin(t= 0.3, unit = "cm")),
        axis.title.y = element_text(margin = margin(l = 0.1, r = 0.2, unit = "cm")))+
  scale_fill_manual(values = c("#A2D9D9","#73C5C5","#009596"))
```

Figure 129-3.4.4.3 Box Plot for ANOVA Test Visualization

3.4.5 Analysis-4.5: What is the Distribution of Size of Residential Properties in KLCC, Kuala Lumpur when Price > RM 1,000,000?


```
KLCC_dataset_barchart<-
  ggplot(KLCC_dataset_pie, aes(x=x_axis,y=y_axis, fill = factor(y_axis))) +
  geom_bar(stat="identity", width = 0.8) +
  geom_text(aes(label = y_axis), vjust=1.6, color="white", size=3.5) +
  labs(x = "Property Size (sq ft.)", y = "Number of Properties",
       title="Distribution of Property Sizes in KLCC\n") +
  theme_light() +
  theme(plot.title = element_text(hjust = 0.5,
                                  margin = margin(b = -0.1, t = 0.3, unit = "cm")),
        legend.position="none",
        axis.title.x = element_text(margin = margin(t= 0.3, b = 0.3, unit = "cm")),
        axis.title.y = element_text(margin = margin(l = 0.1, r = 0.2, unit = "cm"))) +
  scale_fill_manual(values=c("#009596","#A2D9D9","#73C5C5"))
KLCC_dataset_barchart
```

```
#Create Stacked Bar Chart for Distribution of Property Sizes in KLCC with Price Variable
KLCC_dataset_stacked <- ggplot(KLCCprice_datasetE, aes(x=x_axis, y = value, fill = newcolor, size = status)) +
  geom_bar(stat = "identity", width = 0.8) +
  geom_text(aes(label = value), vjust=1.6, color="white", size=3.5, position = "stack") +
  labs(x = "Property Size (sq ft.)", y = "Number of Properties",
       title="Distribution of Property Sizes in KLCC\n with Price", size = "Price") +
  scale_fill_manual(
    values = c(`#009596` = "#009596",
               `#73C5C5` = "#73C5C5",
               `#A2D9D9` = "#A2D9D9",
               darkgrey = "darkgrey"),guide = FALSE) +
  theme_light() +
  theme(plot.title = element_text(hjust = 0.5, margin = margin(b = 0.2, t = 0.3, unit = "cm")),
        legend.position="bottom",axis.title.x = element_text(margin = margin(t= 0.3, unit = "cm")),
        axis.title.y = element_text(margin = margin(l = 0.1, r = 0.2, unit = "cm"))) +
  guides(size = guide_legend(title = "Price",
                             title.position = "left",
                             title.theme = element_text(size = 12, face = "bold", margin = margin(t=-1)),
                             label.theme = element_text(size = 10),
                             override.aes = list(fill = c("darkgrey","#73C5C5"))))
KLCC_dataset_stacked
```

Figure 130-3.4.5.1 Bar Charts for Distribution of Property Sizes in KLCC

The first bar chart depicts the distribution of Property Sizes in KLCC. It is then transformed into a stacked bar chart where the colored regions show the residential properties of each size range that have a price greater than RM 1,000,000 while the gray regions show those that do not fulfill the criteria. In the 0 – 1000 sq ft. size range, a significant portion of residential properties (453 of 1120) have a price lower than RM 1,000,000, indicating that smaller properties are cheaper in price. Conversely, a wide majority of residential properties meet the criteria of having a price greater than RM 1,000,000 in the other two size ranges.

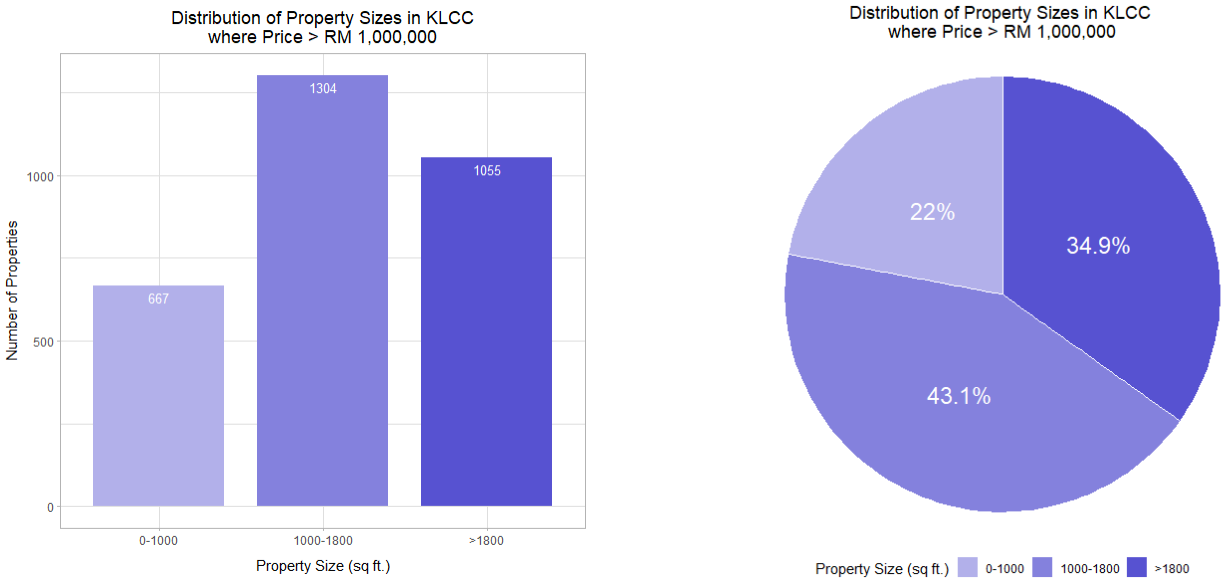


Figure 131-3.4.5.2 Bar & Pie Charts for Distribution of Property Sizes in KLCC where Price > RM 1,000,000 (R Code not shown)

The above bar and pie charts depict that within the subset of properties priced above RM 1,000,000, a notable portion - 43.1%, comprise of residential properties within the 1000 -1800 sq ft. size range, totaling 1304 units. The >1800 sq ft. size range also demonstrates significant representation, accounting for 34.9% of the properties priced above RM 1,000,000, totaling 1055 units.

We will proceed with the subsequent stage of analysis focusing on both these size ranges. In order to streamline the explanation in the next section, the dataset comprising properties within the 1000 - 1800 sq ft. size range will be designated as Dataset A, while the dataset containing properties exceeding 1800 sq ft. will be labeled as Dataset B. The dataset which has a size range of 0 – 1000 square feet will not be used.

3.4.6 Analysis-4.6: Exploring the Distribution of Residential Property Sizes in KLCC, Kuala Lumpur for Properties with Price > RM 1,000,000 in Both Datasets

```
overall_furnishing_barchart <-
  ggplot(overall_furnishing_dataset, aes(fill=color_value, y=no, x=size)) +
  geom_bar(position="dodge", stat="identity", width = 0.9) +
  geom_text(aes(label = no), position = position_dodge(width = 0.9),vjust = 1.6, color = "white") +
  labs(x = "Property Size (sq ft.)", y = "Number of Properties",
       title="Distribution of Property Furnishing Status in KLCC\nwhere Price > RM 1,000,000",
       fill = "Furnishing Status") +
  theme_light() +
  theme(plot.title = element_text(hjust = 0.5, margin = margin(b = 0.2, t = 0.3, unit = "cm")),
        legend.position="bottom",
        axis.title.x = element_text(margin = margin(t= 0.3,b = - 0.1, unit = "cm")),
        axis.title.y = element_text(margin = margin(l = 0.1, r = 0.2, unit = "cm")),
        legend.box.background = element_rect(color = "grey", size = 1),
        legend.title = element_text(size = 10)) +

  scale_fill_manual(values= c(`1` = "#F9E79E",
                              `2` = "#F1C40F",
                              `3` = "#F7DC6F",
                              `4` = "#F4B678",
                              `5` = "#EC7A08",
                              `6` = "#EF9234"),
                    breaks = c("1", "4", "2", "5", "3","6"),
                    labels = c("Fully Furnished","Fully Furnished", "Partly Furnished",
                               "Partly Furnished","Unfurnished", "Unfurnished")) +
  guides(fill = guide_legend(title.position = "top", title.hjust = 0.5))

overall_furnishing_barchart
```

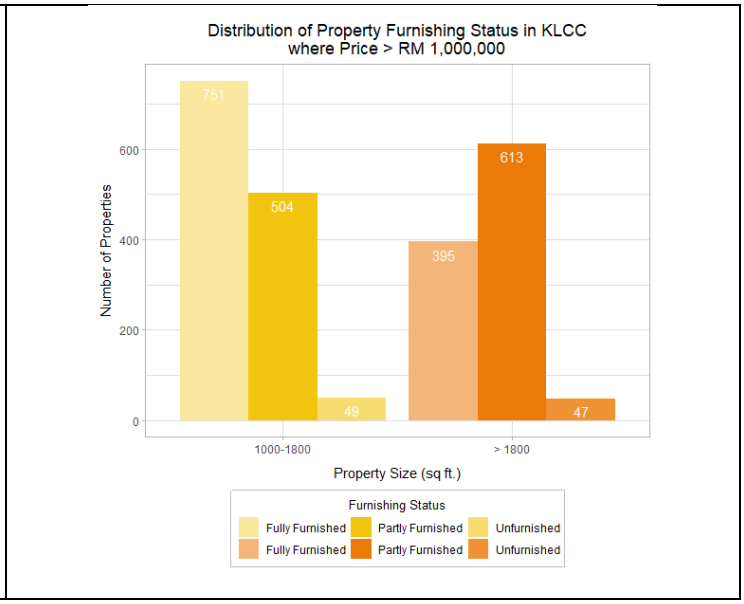


Figure 132-3.4.6.1 Clustered Bar Chart for Distribution of Furnishing Status in KLCC where Price > RM 1,000,000 for both datasets

Based on the clustered bar chart, in Dataset A, a considerable number of properties (751 units) with Fully Furnished status meet the aforementioned criteria. Meanwhile, within Dataset B, the prevailing furnishing status is Partly Furnished, encompassing 631 residential properties.

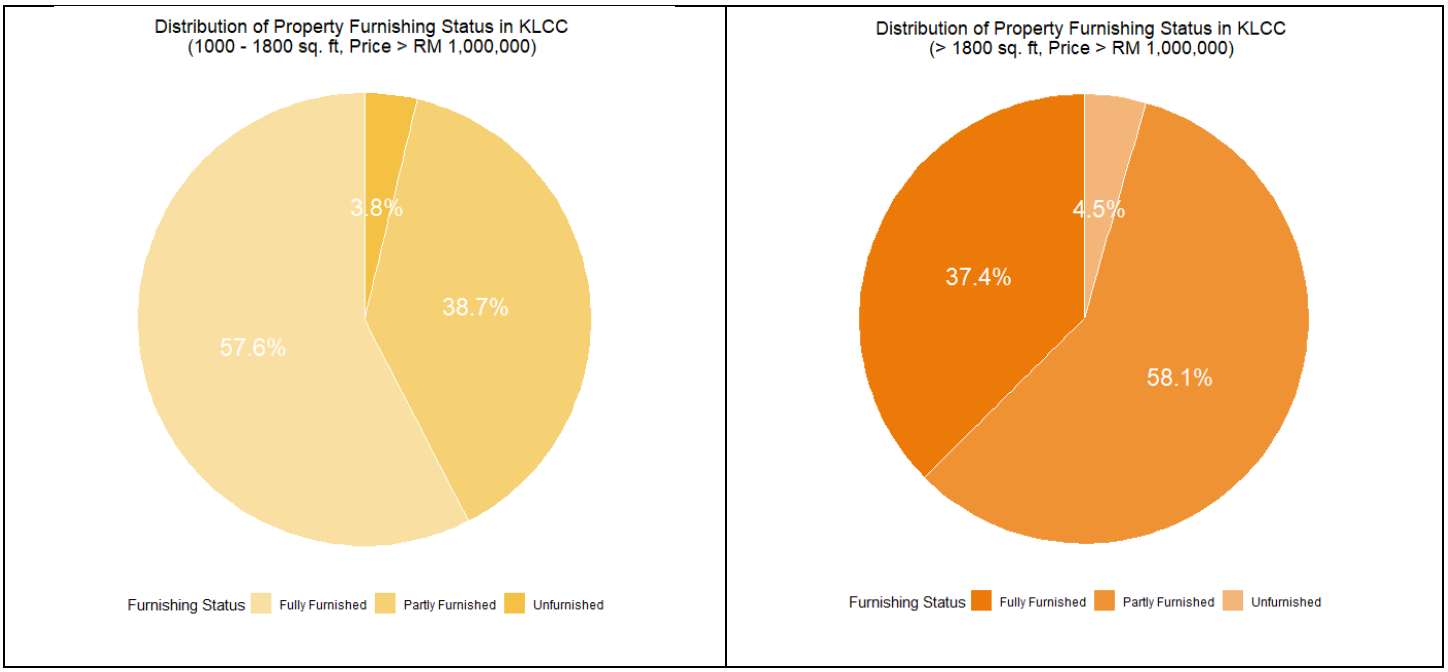


Figure 133-3.4.6.2 Pie Charts for Furnishing Status Distribution in Both Datasets (R code not shown)

In Dataset A, 57.6 % of the properties within the size range of 1000 – 1800 sq ft. and priced more than RM 1,000,000 are fully furnished. However, on the contrary, 58.1 % of the properties greater than 1800 sq ft. and price more than RM 1,000,000 are partly furnished. This suggests that furnishing status, with the exception of Unfurnished properties which charted low on both pie charts, does not greatly impact the price of residential properties in KLCC.

3.4.7 Analysis-4.7: Exploring the Distribution of Built Area Types in KLCC, Kuala Lumpur for Properties with Price > RM 1,000,000 in Both Datasets

```
Overall_built_area_barchart <-
  ggplot(overall_built_area_dataset, aes(fill=color_value, y=no, x=data)) +
  geom_bar(position="stack", stat="identity", width = 0.8) +
  geom_text(aes(label = no),vjust = 1.025, color = "white") +
  labs(x = "Datasets", y = "Number of Properties",
       title="Distribution of Property Built Area Type in KLCC\nwhere Price > RM 1,000,000",
       fill = "Type of Built Area") +
  theme_light() +
  theme(plot.title = element_text(hjust = 0.5,
                                  margin = margin(b = 0.2, t = 0.3, unit = "cm")),
        legend.position="none",
        axis.title.x = element_text(margin = margin(t= 0.3,b = 0.4, unit = "cm")),
        axis.title.y = element_text(margin = margin(l = 0.1, r = 0.2, unit = "cm"))) +
  scale_fill_manual(values= c(`1` = "#abebc6",
                              `2` = "#58d68d",
                              `3` = "#AED581",
                              `4` = "#88C34A"),guide = FALSE)
Overall_built_area_barchart
```

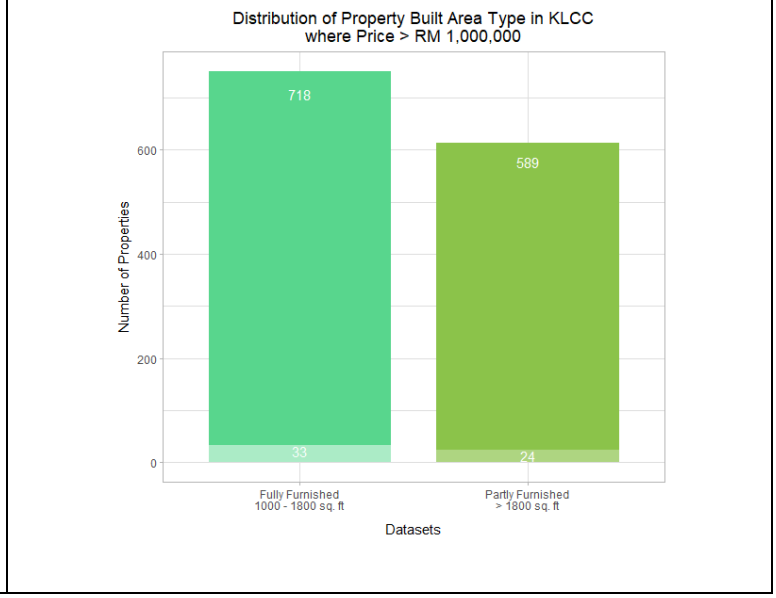


Figure 134-3.4.7.1 Stacked Bar Chart for Distribution of Built Area in KLCC where Price > RM 1,000,000 for both datasets

The stacked bar chart above highlights that an exceptionally high proportion of properties in both datasets are built-up properties. This trend is consistent across Dataset A and Dataset B, indicating a prevalent preference for built-up properties in the KLCC area. Built-up properties account for 95.6% (718 units) and 96.1% (589 units) in Dataset A and B respectively. Land area properties account for a mere 4.4% (33 units) and 3.9% (24 units) in Dataset A and B, respectively, as revealed in the pie chart below.

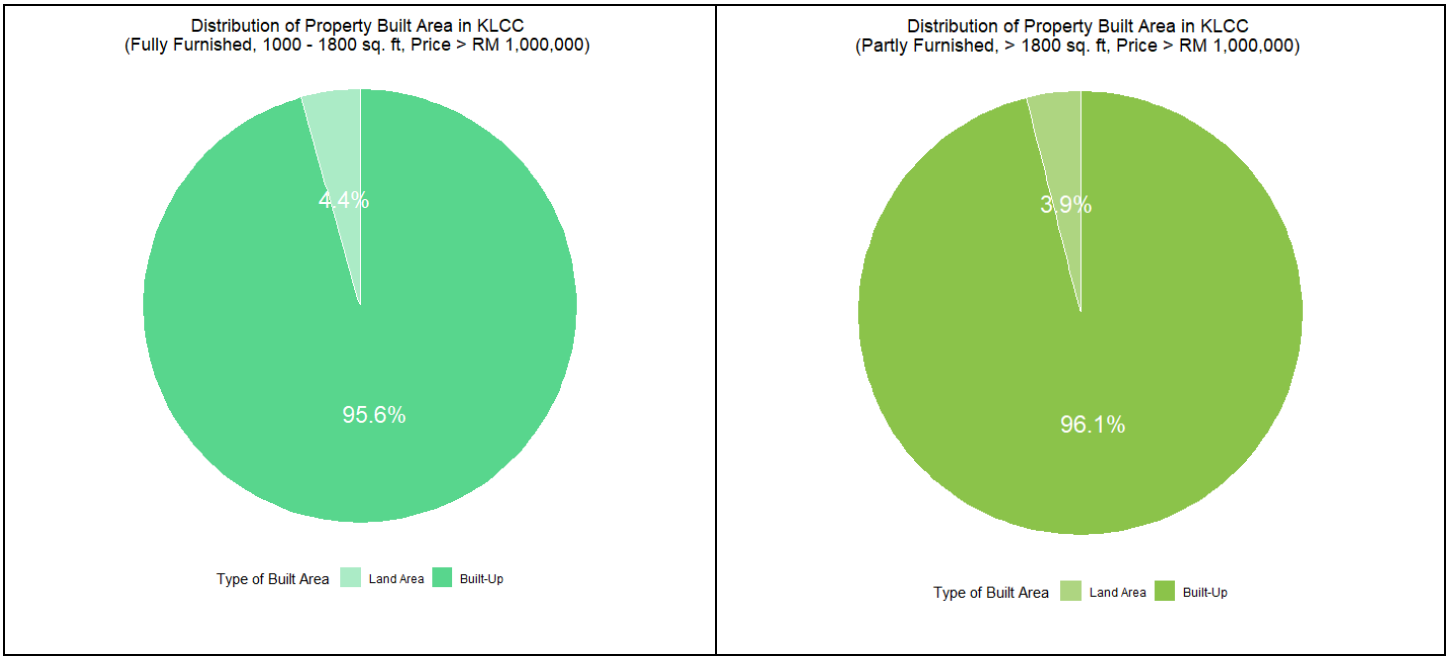


Figure 135-3.4.7.2 Pie Charts for Built Area Type distribution for Both Datasets (R code not shown)

3.4.8 Analysis-4.8: Exploring the Distribution of Room Count in KLCC, Kuala Lumpur for Properties with Price > RM 1,000,000 in Both Datasets

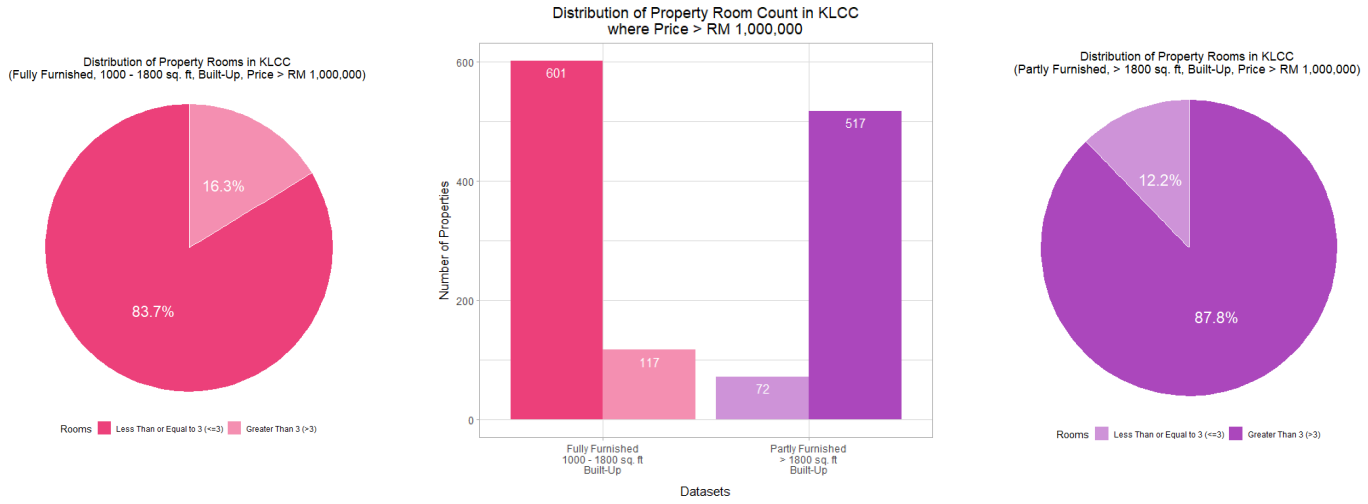
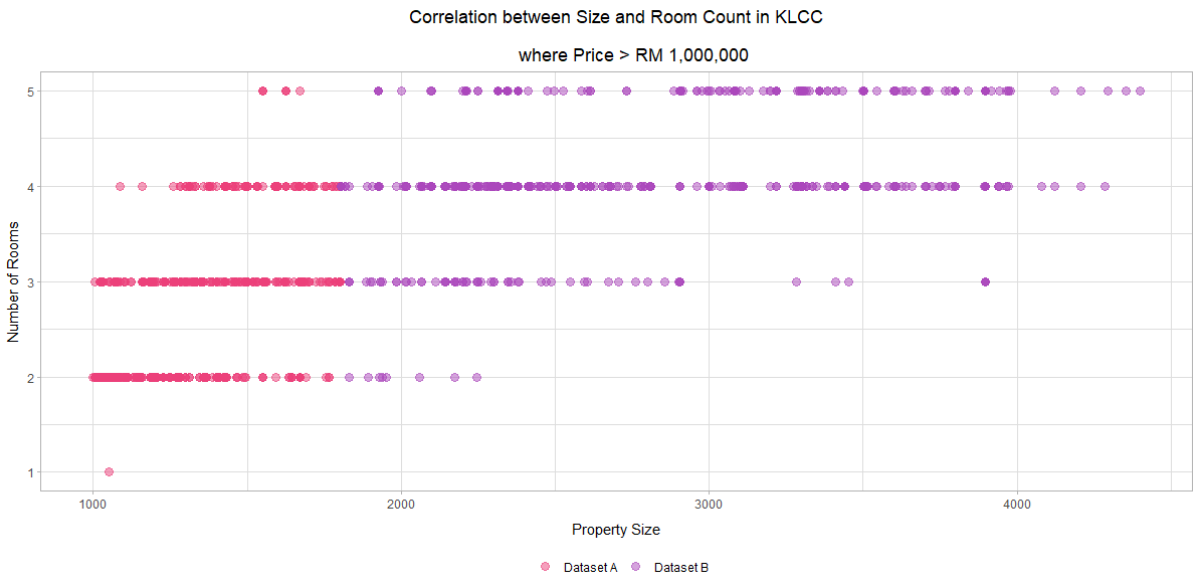


Figure 136-3.4.8.1 Pie Charts & Stacked Bar Chart for Room Count Distribution in Both Datasets (R code not shown)

The charts above present contrasting outcomes for both datasets. In Dataset A, properties with three or fewer rooms are more prevalent, comprising 83.7% (601 units) of the dataset. However, Dataset B is composed of properties with more than 3 rooms, accounting for 87.8% (517 units) of the dataset. The relationship between the room count and property size will be explained in the next section.

3.4.9 Analysis-4.9: Investigating the Correlation between Size & Room Count in KLCC in Both Datasets




```
Scatter_plot <-
  ggplot(Combined_scatter, aes(x=Property_Size, y=Property_Rooms,
                               color = dataset)) + geom_point(size = 3, alpha = 0.5) +
  labs(x = "Property Size", y = "Number of Rooms", title="Correlation between Size and Room Count in KLCC\nwhere Price > RM 1,000,000", color = "") +
  theme_light() +
  theme(plot.title = element_text(hjust = 0.5,margin = margin(b = 0.2, t = 0.3, unit = "cm")), legend.position="bottom",axis.title.x = element_text(margin = margin(t= 0.3,b = - 0.1
        axis.title.y = element_text(margin = margin(l = 0.1, r = 0.2, unit = "cm"))) +
  scale_color_manual(values= c("Dataset A" = "#ec407a","Dataset B" = "#AB47BC"))
```

Figure 137-3.4.9.1 Scatter Plot for Correlation between Size & Room in Both Datasets

The scatter plot above reveals that there is a clear linear relationship between property size and number of rooms. As property size increases, there is a corresponding increase in the number of rooms. Given that Dataset A primarily consists of smaller properties compared to Dataset B, it follows that a significant portion of properties meeting the criteria would have three or fewer rooms.

3.4.10 Conclusion

Among the total 3026 residential properties in KLCC that are priced greater than RM 1,000,000:

1. 19.86% of Fully Furnished built-up properties within the square feet of 1000 – 1800 range and featuring three or fewer rooms are priced greater than RM 1,000,000 in KLCC as evident in Dataset A.
2. 17.09% of Partly Furnished built-up properties exceeding 1800 square feet and featuring more than three rooms are priced greater than RM 1,000,000 in KLCC as evident in Dataset B.

In summary, while both conclusions differ from the initial hypothesis, it provides various insights into the relationship between the property attributes and price exceeding RM 1,000,000 in KLCC. The variables property size, built type greatly affects the property price, while furnishing status and number of rooms have little to no impact on the property price.

3.4.11 Additional Features

Analysis	Feature	Function
4.1.1	lm()	To fit linear regression models to data.
	geom_smooth(method = "lm")	Uses a linear regression model to generate smooth line in scatterplot.
	geom_smooth(se = FALSE)	Disables display of confidence intervals.
	theme_light()	Set light-coloured themes for plot appearance
4.2.1	element_text(hjust = 0.5)	Centers the plot title horizontally.
	margin	Adjusts the margins.
	legend.position	Sets the position of legends.
	axis.title.x	Customise x-axis labels and position.
	axis.title.y	Customise y-axis labels and position.
4.2.2	coord_polar("y", start=0)	Transforms the Cartesian coordinate to polar, with the y-axis mapped to radial distance from the center.
	theme_void()	Renders a blank canvas for custom plotting.
	geom_text()	Adds text annotations to a plot.
	position= position_stack(vjust = 0.5)	Aligns stacked elements with the specified vertical justification.
4.3.1	geom_segment()	Used to add line segments to a plot
	alpha = 0.7	Set transparency of graphical elements.
	label = signif(y_axis)	Assigns labels to elements on the y-axis
	nudge_x = 0.35	Moves elements horizontally by 0.35 units,
	coord_flip()	Transposes the plot.
	ylim()	Sets the limits of the y-axis in a plot
4.3.2	rbind()	Combine vectors, matrices, or data frames by row
	geom_violin()	Creates a violin plot.
4.4.1	library(smplot2)	Simplifies data visualization process.
	sm_statCorr()	Compute and visualize correlation statistics between variables
	corr_method = 'pearson'	Uses Pearson correlation coefficient to compute correlations between variables.
	linetype = 'dashed'	Sets the line type to be dashed in the smplot2 package.
	cor.test()	Perform correlation tests to assess correlation between two numeric variables.
4.4.2	aov()	Perform analysis of variance (ANOVA) tests.
	library(multcomp)	Provides functions for multiple comparison procedures.
	Glht()	To perform general linear hypothesis tests.
	linfct=mcp(colname= "Tukey")	Used to specify the type of multiple comparison procedure (MCP) to be applied, which is Tukey Test in this example.
4.4.3	Same as 4.2.1	
4.5.1	tidyr::pivot_longer()	To convert multiple columns (variables) into two columns.

	ifelse()	Perform conditional operations.
	position = "stack"	To stack overlapping graphical elements.
	guide = FALSE	Hide display of legend or guide.
	guides()	To customize the appearance and behavior of legends or guides.
	override.aes	To override the default aesthetics settings for specific elements.
4.5.2	Same as 4.2.1	
4.6.1	rep()	Replicate elements of a dataset by certain amount of times.
	position="dodge"	Position overlapping graphical elements side by side.
	position=position_dodge(width = 0.9)	Applies dodging to the position of elements.
	legend.box.background	Customizes the background of the legend box.
	element_rect()	Customize the appearance of rectangular elements in a plot.
	breaks	Specifies the locations where breaks should occur on the axis scale
4.6.2	Same as 4.2.1	
4.7.1	Same as 4.5.1	
4.7.2	Same as 4.2.1	
4.8.1	Same as 4.2.1 & 4.6.1	
4.9.1	Same as 4.2.1	

4.0 Conclusion

4.1 Overall discussion

All of our members reached three different conclusions throughout our own analysis. After our discussion, we decided to choose two of the conclusions with the greatest and closest percentages as our final conclusion. The first conclusion that we came up with is that approximately 20% of houses in KLCC priced above 1 million are fully furnished, built-up, between 1000 and 1800 square feet in size, and have fewer than or equal to three bedrooms. This conclusion differs from our hypothesis at the independent variable Property_Rooms. The second conclusion we reached is that approximately 17% of houses in KLCC priced above 1 million are partly furnished, built-up, with a property size bigger than 1800 square feet and more than three bedrooms. This conclusion differs from our hypothesis at the independent variable Property_Furnishing and Property_Size According to this two final conclusion, the independent variable Property_Rooms is statistically significant on fully furnished houses in KLCC with prices greater than one million. Meanwhile, the Property_Size independent variable has a significant impact on the partly furnished properties in KLCC that cost more than 1 million. On the other hand, the Property_Rooms are statistically insignificant to the Partly Furnished properties in KLCC. As a final conclusion, it is clear that our hypothesis is wrong, and we had came up with two final conclusions as follows:

1. Around 20% of Fully Furnished Built-up properties with less than or equal to 3 rooms and 1000 – 1800 square feet has a price greater than RM 1 million in KLCC.
2. Around 17% of Partly Furnished Built-up properties with more than 3 rooms and size greater than 1800 square feet have a price greater than RM 1 million in KLCC.

4.2 Recommendation

As for recommendations to stakeholders, it is recommended that property dealers segment their offerings based on the attributes that drive pricing. Important attributes to consider for residential properties in KLCC are Built-Type and Size as they significantly impact the property price. Meanwhile, investors looking to break through and enter into the competitive real estate market in KLCC should leverage insights from the data to identify lucrative investment opportunities in the future. They should closely monitor residential properties that align with the desired attributes as they might be good for future investment.

4.3 Limitation and future direction

One slight limitation faced by the team members is the lack of experience in performing statistical tests on the datasets. The team members are required to look for external sources and perform in-depth research on these foreign topics in order to conduct statistical testing and find out the relationships between the variables.

Another limitation faced by the team is the time constraint to complete this detailed data analysis focusing on residential property price trends. The intricate nature of the analysis, combined with the need for thoroughness and accuracy, demands significant time investment. With the predefined deadline set so close, it understandably is hard for the team to maximize the quality of the analysis, as the team might require sacrificing quality for speed in certain areas.

5.0 Workload Matrix

Component	Student 1: Ooi Chong Ming	Student2: Yeoh Zi Qing Bryan	Student 3: Sim Sau Yang	Student 4: Lim Wen Hann	Total
1.0 Introduction	**25%**	**25%**	**25%**	**25%**	**100%**
2.0 Data Preparation	**25%**	**25%**	**25%**	**25%**	**100%**
2.1 Data Import	**25%**	**25%**	**25%**	**25%**	**100%**
2.2 Data Cleaning	**25%**	**25%**	**25%**	**25%**	**100%**
2.3 Data Validation	**25%**	**25%**	**25%**	**25%**	**100%**
3.0 Data analysis	**25%**	**25%**	**25%**	**25%**	**100%**
3.1 Objective 1	**100%**	**-**	**-**	**-**	**100%**
3.2 Objective 2	**-**	**100%**	**-**	**-**	**100%**
3.3 Objective 3	**-**	**-**	**100%**	**-**	**100%**
3.4 Objective 4	**-**	**-**	**-**	**100%**	**100%**
4.0 Conclusion	**25%**	**25%**	**25%**	**25%**	**100%**

Word Count : 14112 words

Page Count : 59 pages