PRINCESS NOURAH UNIVERSITY

DEPARTMENT OF COMPUTER AND INFORMATION SCIENCES

Masters of Data Science *First Term 2023*

Houses Price

Statistical Data Analysis 2023

This Report is Submitted in Complete Fulfillment of the Statistical Data Analysis Project

Date:26/11/2023

Author:

Sawsan Daban 445009481

Alaa Alsharekh 445009444

Supervisor:

Dr. Seham Bas. Meshoul

ACKNOWLEDGEMENTS

We express our deep gratitude to the Almighty Allah for giving us an opportunity to successfully finish modeling and simulating and write this report.

We would like to express our gratitude and appreciation to Princess Nourah Bint Abdulrahman University for giving us the opportunity to continue learning and achieve new things in our life, and for their remarkable services they gave us.

We would like to express our gratitude to Dr. Seham Bas. Meshoul for her kindness and unequivocal support.

We would like to thank our friends for helping us realize what We are missing from our lives. We thank them for their interesting discussions and help. Without them, We would not be here.

Finally, We would like to thank our family for their unwavering support throughout our lives, pushing us to always do our best.

EXECUTIVE SUMMARY

This report includes information about the problem statement of the project, goals and objectives, data summary, data preparation, and analysis.

Contents

List of Figures

List of Tables

Introduction

1.1 Problem Statement

The dataset provided is a dataset of house prices. It includes information about the house area, house age, distance to the nearest metro station, number of convenience stores, latitude and longitude, and the price of unit area.

1.2 Goals And Objectives

- Gain a good understanding of the different level of measurements, data types.
- Understand data using graphical displays.
- Gain a good understanding of descriptive statistics through univariate and bivariate analysis.
- Gain a good understanding of inferential statistics
- Select and apply the suitable summary statistics.
- Select and apply the suitable hypothesis testing method.

Data Preparation

2.1 Explore Data

```r
# Install and load necessary packages
install.packages("ggplot2")
install.packages("dplyr")
library(ggplot2)
library(dplyr)
```

Figure 2.1: Import the necessary packages

```r
# Load the dataset
data <- read.csv("data.csv", header = TRUE)
```

Figure 2.2: Importing and Reading the Data

```r
# View the first few rows of the dataset
head(data)
```

A data.frame: 6 x 8

	No	House.Area	House.Age	Nearest.Metro.Station	Number.of.Convenience.Stores	latitude	longitude
	<int>	<chr>	<dbl>	<dbl>	<int>	<dbl>	<dbl>
1	1	Medium	32.0	84.87882	10	24.98298	121.5402
2	2	Medium	19.5	306.59470	9	24.98034	121.5395
3	3	Medium	13.3	561.98450	5	24.98746	121.5439
4	4	Large	13.3	561.98450	5	24.98746	121.5439
5	5	Large	5.0	390.56840	5	24.97937	121.5425
6	6	Large	7.1	2175.03000	3	24.96305	121.5125

Figure 2.3: Displaying the first 6 rows of the Data

The dataset includes information about the house area, house age, distance to the nearest metro station, number of convenience stores, latitude and longitude, and the price of unit area.

2.2 Data Summary

```r
# Get the shape of the data frame
data_shape <- dim(data)

# Print the number of rows and columns
cat("Number of rows:", data_shape[1], "\n")
cat("Number of columns:", data_shape[2])
```

```
Number of rows: 414
Number of columns: 8
```

Figure 2.4: Get the shape of the data frame

```r
# Get the summary of the dataset
summary(data)
```

```
       No          House.Area           House.Age      Nearest.Metro.Station
 Min.   :  1.0   Length:414         Min.   : 0.000   Min.   :  23.38
 1st Qu.:104.2   Class :character   1st Qu.: 9.025   1st Qu.: 289.32
 Median :207.5   Mode  :character   Median :16.100   Median : 492.23
 Mean   :207.5                      Mean   :17.713   Mean   :1083.89
 3rd Qu.:310.8                      3rd Qu.:28.150   3rd Qu.:1454.28
 Max.   :414.0                      Max.   :43.800   Max.   :6488.02
 Number.of.Convenience.Stores    latitude       longitude
 Min.   : 0.000               Min.   :24.93   Min.   :121.5
 1st Qu.: 1.000               1st Qu.:24.96   1st Qu.:121.5
 Median : 4.000               Median :24.97   Median :121.5
 Mean   : 4.094               Mean   :24.97   Mean   :121.5
 3rd Qu.: 6.000               3rd Qu.:24.98   3rd Qu.:121.5
 Max.   :10.000               Max.   :25.01   Max.   :121.6
 House.price.of.Unit.Area
 Min.   :  7.60
 1st Qu.: 27.70
 Median : 38.45
 Mean   : 37.98
 3rd Qu.: 46.60
 Max.   :117.50
```

Figure 2.5: Get the summary of the dataset

The dataset 2.2 has 414 rows, each representing a different house. The house age ranges from 0 to 43.8 years. The distance to the nearest metro station ranges from 23.38 to 6488.02 meters. The number of convenience stores in the area of the house ranges from 0 to 10. The latitude ranges from 24.93207 to 25.01459 degrees north. The longitude ranges from 121.4735 to 121.5663 degrees east. The price of unit area ranges from 7.60 to 117.50 USD per square meter 2.5.

```r
# Get the data types of each variable
str(data)
```

```
'data.frame':   414 obs. of  8 variables:
 $ No                        : int  1 2 3 4 5 6 7 8 9 10 ...
 $ House.Area                : chr  "Medium" "Medium" "Medium" "Large" ...
 $ House.Age                 : num  32 19.5 13.3 13.3 5 7.1 34.5 20.3 31.7 17.9 ...
 $ Nearest.Metro.Station     : num  84.9 306.6 562 562 390.6 ...
 $ Number.of.Convenience.Stores: int  10 9 5 5 5 3 7 6 1 3 ...
 $ latitude                  : num  25 25 25 25 25 ...
 $ longitude                 : num  122 122 122 122 122 ...
 $ House.price.of.Unit.Area  : num  37.9 42.2 47.3 54.8 43.1 32.1 40.3 46.7 18.8 22.1 ...
```

Figure 2.6: Get the data types of each variable

```r
# Get all unique values in a House.Area column
unique_house_area_values <- unique(data$House.Area)

# Print the unique values
cat("Unique House Area values:", unique_house_area_values)
```

```
Unique House Area values: Medium Large Small
```

Figure 2.7: Get all unique values in a House.Area column

The dataset is balanced, with an equal number of houses in each of the three categories of house size (small, medium, and large). The dataset is also representative of the distribution of house prices in the area, with a mix of low-priced, medium-priced, and high-priced houses.

```r
#Get the highest unit price row
highest_unit_price_row <- data[order(-data$House.price.of.Unit.Area, decreasing = TRUE), , drop = FALSE][1,]

#Display the row
highest_unit_price_row
```

A data.frame: 1 x 8

	No	House.Area	House.Age	Nearest.Metro.Station	Number.of.Convenience.Stores	latitude	longitude
	<int>	<chr>	<dbl>	<dbl>	<int>	<dbl>	<dbl>
114	114	Small	14.8	393.2606	6	24.96172	121.5381

Figure 2.8: Get the highest unit price row

House number 114 has the highest house price of unit area.

2.3 Data Preprocessing

Data preprocessing is a crucial step in the data analysis pipeline aimed at cleaning, organizing, and transforming raw data into a format suitable for analysis. These steps are crucial to ensure the accuracy and reliability of statistical analyses. It serves several steps:

- Handle missing values

```
# Check for any missing values in the dataset
missing_values <- sum(is.na(data))

# Print the number of missing values
cat("Number of missing values:", missing_values)

# Check for missing values
sapply(data, is.na)
```

```
Number of missing values: 0
```

A matrix: 414 x 8 of type lgl

No	House.Area	House.Age	Nearest.Metro.Station	Number.of.Convenience.Stores	latitude	longitude	Hous
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALS
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALS
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALS
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALS
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALS
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALS
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALS
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALS
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALS
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALS
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALS
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALS
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALS
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALS
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALS
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALS
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALS
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALS
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALS

Figure 2.9: Check for any missing values in the dataset

- Handle duplication

```r
# Check for duplicate rows in the data frame
duplicate_rows <- data[duplicated(data) | duplicated(data, fromLast = TRUE), ]

# Print the duplicate rows
cat("Duplicate rows:\n")
print(duplicate_rows)
```

```
Duplicate rows:
[1] No                          House.Area
[3] House.Age                   Nearest.Metro.Station
[5] Number.of.Convenience.Stores latitude
[7] longitude                   House.price.of.Unit.Area
<0 rows> (or 0-length row.names)
```

Figure 2.10: Check for duplicate rows in the dataset

There are no missing values nor duplicated rows in the dataset.

- Handle outliers

```r
# Check for outliers
boxplot(data$House.price.of.Unit.Area)
```

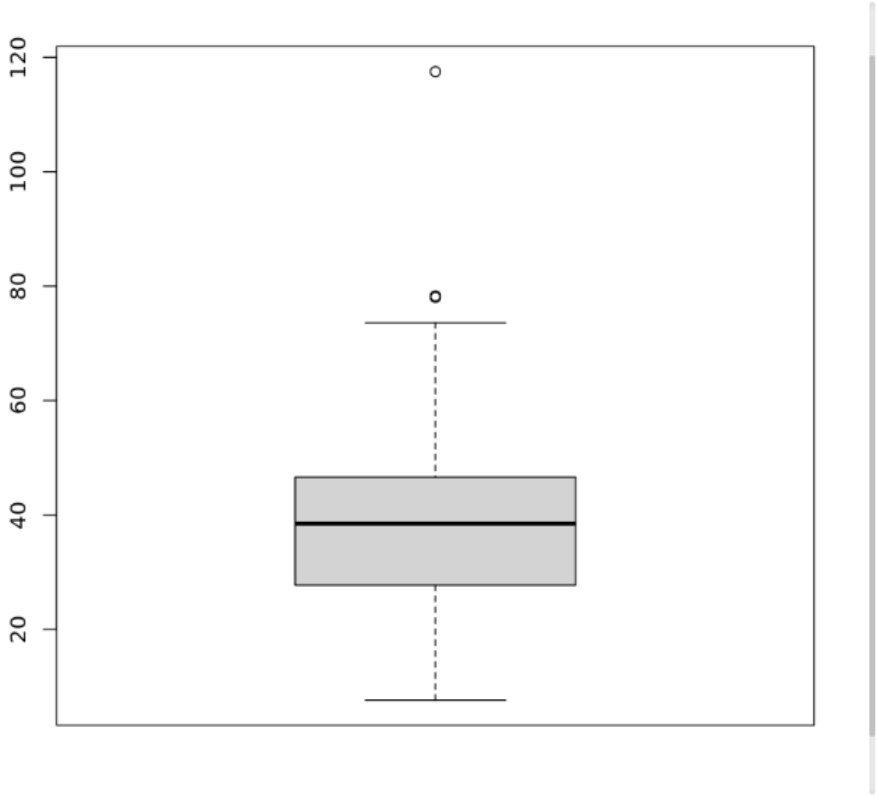


Figure 2.11: Check for outliers

Based on the boxplot in figure 2.11 we see that there are outliers in the House.price.of.Unit.Area variable.

```r
# Identify outliers using IQR
IQR <- quantile(data$House.price.of.Unit.Area, probs = c(0.25, 0.75))
lower_limit <- IQR[1] - 1.5 * (IQR[2] - IQR[1])
upper_limit <- IQR[2] + 1.5 * (IQR[2] - IQR[1])

# Identify outliers
outliers <- data$House.price.of.Unit.Area[data$House.price.of.Unit.Area < lower_limit
                                          | data$House.price.of.Unit.Area > upper_limit]

# Remove outliers
data_without_outliers <- data[!(data$House.price.of.Unit.Area %in% outliers), ]
```

Figure 2.12: Identify outliers using IQR

```r
# Check for outliers
boxplot(data_without_outliers$House.price.of.Unit.Area)
```

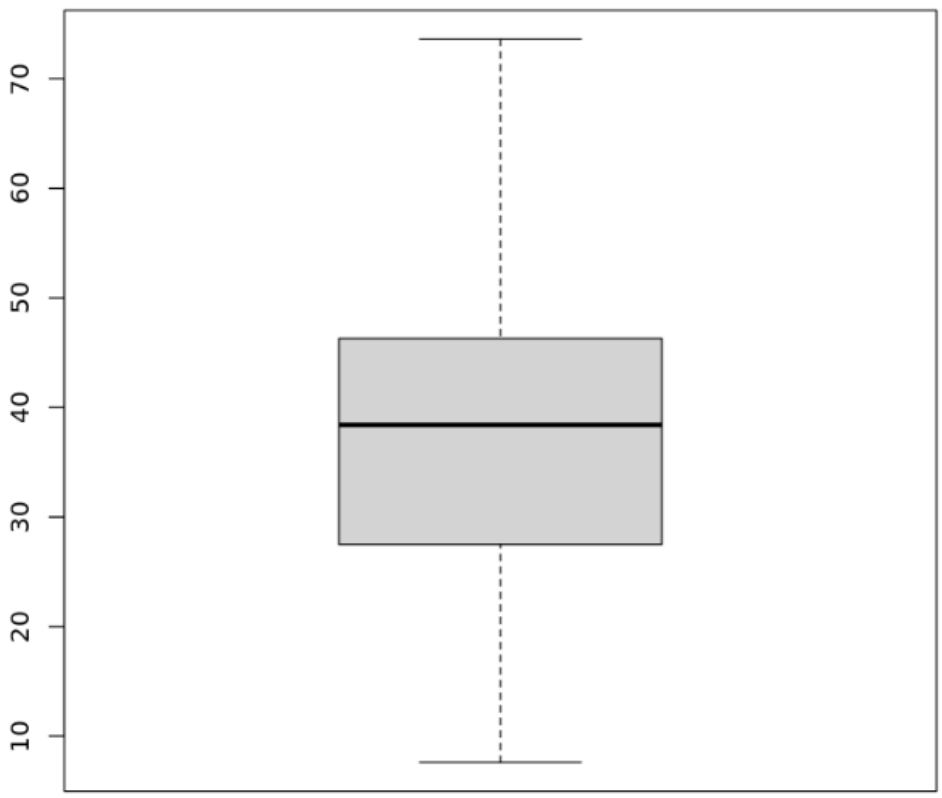


Figure 2.13: Check for outliers

- Handle data type errors

```r
# Check for data type errors
str(data_without_outliers)
```

```
'data.frame':   411 obs. of  8 variables:
 $ No                          : int  1 2 3 4 5 6 7 8 9 10 ...
 $ House.Area                  : chr  "Medium" "Medium" "Medium" "Large" ...
 $ House.Age                   : num  32 19.5 13.3 13.3 5 7.1 34.5 20.3 31.7 17.9 ...
 $ Nearest.Metro.Station       : num  84.9 306.6 562 562 390.6 ...
 $ Number.of.Convenience.Stores: int  10 9 5 5 5 3 7 6 1 3 ...
 $ latitude                    : num  25 25 25 25 25 ...
 $ longitude                   : num  122 122 122 122 122 ...
 $ House.price.of.Unit.Area    : num  37.9 42.2 47.3 54.8 43.1 32.1 40.3 46.7 18.8 22.1 ...
```

Figure 2.14: Check for data type errors

There is one categorical column that is House.Area which contains three possible values, therefore to handle it we encoded the column to three possible numbers (1: small, 2: medium, 3:large).

- Data Encoding (Encode categorical variables)

```r
# Create a factor variable for the house size category
data_without_outliers$House.Area <- as.factor(data_without_outliers$House.Area)
```

```r
# Encode categorical variables
data_without_outliers <- data_without_outliers %>%
  mutate(House.Area.Encoded = as.numeric(House.Area))
```

Figure 2.15: Encode categorical variables

Analysis

3.1 Univariate Analysis

Table 3.1: Skewness Values

Skewness Value	Approximate Interpretation
Less than -1	Strongly skewed to the left
Between -1 and -0.5	Moderately skewed to the left
Between -0.5 and 0.5	Approximately normal
Between 0.5 and 1	Moderately skewed to the right
More than 1	Strongly skewed to the right

3.1.1 Categorical variables

```r
# Print the summary statistics
print("Summary:")
print(summary(data_without_outliers$House.Area))
skewness_value  <-skewness(as.numeric(data_without_outliers$House.Area))
print(paste("Skewness: ",skewness_value ))
ggplot(data_without_outliers, aes(x = House.Area, fill=House.Area)) +
    geom_bar() +
    scale_fill_manual(values = c("#CD7F32", "#C0C0C0", "gold")) +
    labs(title =  paste("Histogram of", deparse(substitute(House.Area))), y="Frequency")
```

Figure 3.1: Univiriate Analysis Categorical Variables Function

House Area

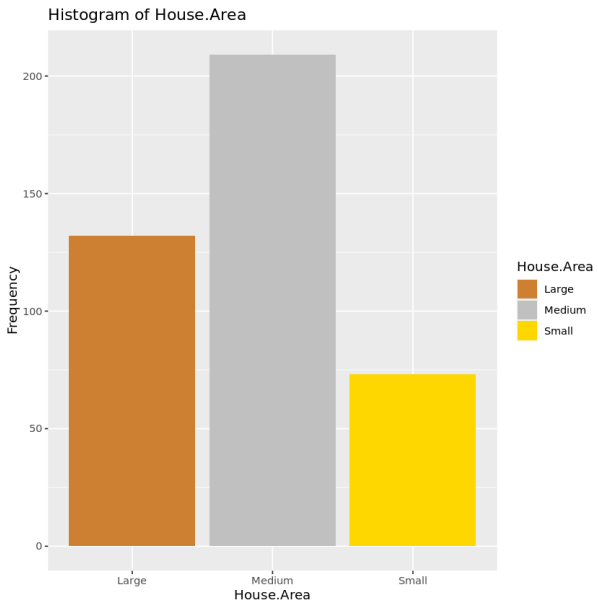


Figure 3.2: Univiriate Analysis For House Area

According to the figure 3.9, the House.Area variable is approximately normally distributed.

The skewness value is 0.18, which means that the distribution of the data is approximately normally distributed.

3.1.2 Numeric variables

```r
# Define a function to calculate summary statistics and plot histograms
univariate_analysis <- function(x) {
    # Extract the variable name from the symbol and remove the data frame prefix
    xname <- gsub("^data_without_outliers\\$", "", deparse(substitute(x)))
    # Print the summary statistics
    print("Summary:")
    print(summary(x))
    skewness_value  <-skewness(x)
    print(paste("Skewness: ",skewness_value ))
    # Plot the histogram
    hist(x, main = paste("Histogram of", deparse(xname)), xlab = xname, col = c("#CD7F32", "#C0C0C0", "gold"))
}
```

Figure 3.3: Univiriate Analysis Numeric Variables Function

House Age

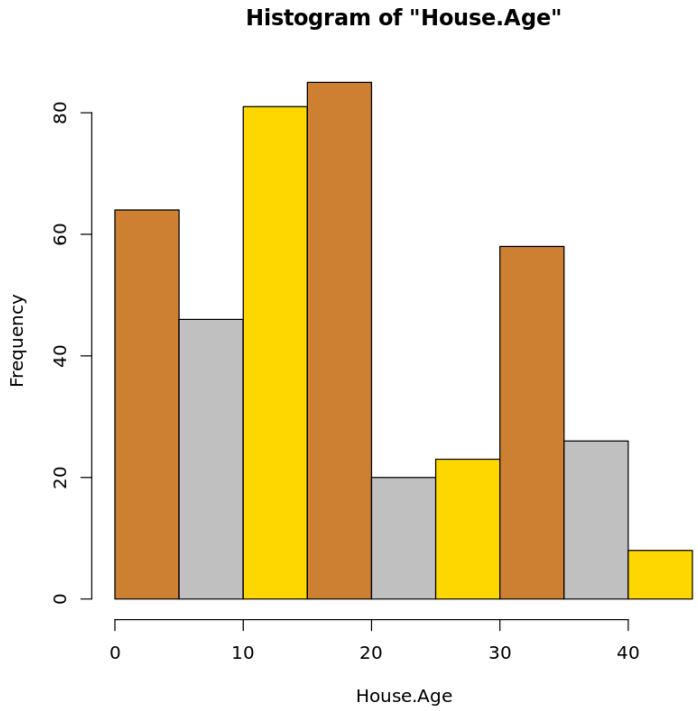


Figure 3.4: Univiriate Analysis For House Age

According to the figure 3.4, the House.Age variable is approximately normally distributed.

The skewness value is 0.38, which means that the distribution is approximately normally distributed.

Nearest Metro Station

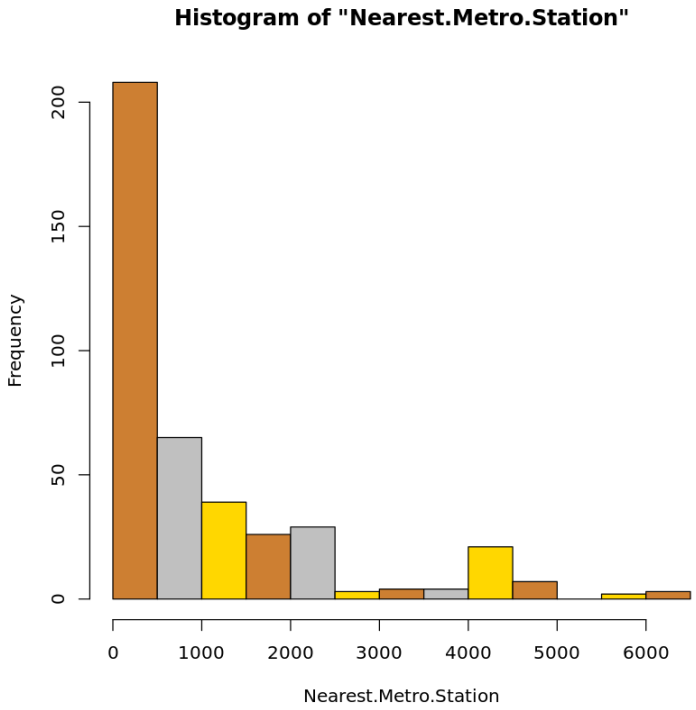


Figure 3.5: Univiriate Analysis For Nearest Metro Station

According to the figure 3.5, the Nearest.Metro.Station variable is skewed right.

The skewness value is 1.87, which means that the distribution is skewed right.

Number Of Convenience Stores

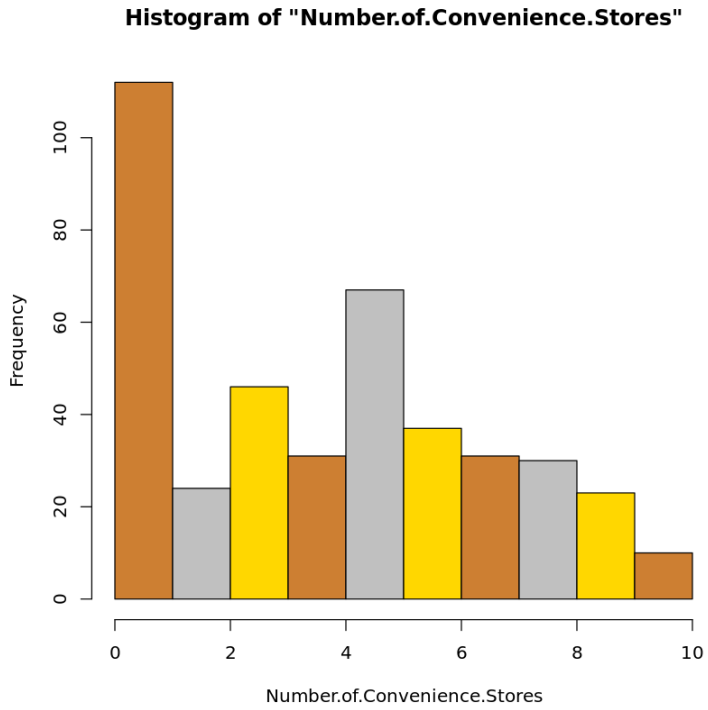


Figure 3.6: Univiriate Analysis For Number Of Convenience Stores

The skewness value of figure 3.6 is 0.15, which means that the distribution is approximately normally distributed.

Latitude

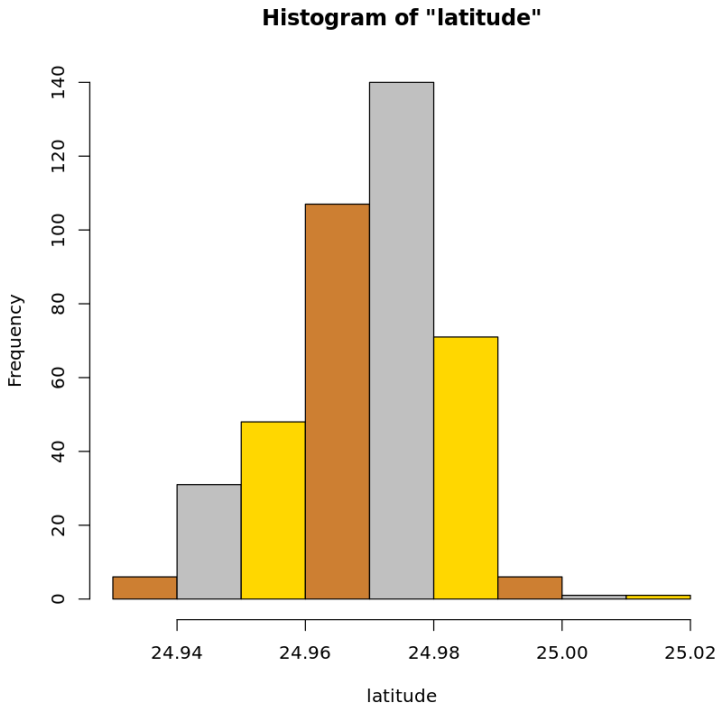


Figure 3.7: Univiriate Analysis For Latitude

According to the figure 3.7, the latitude variable is approximately normally distributed.

The skewness value is -0.42, which means that the distribution is approximately normally distributed.

Longitude

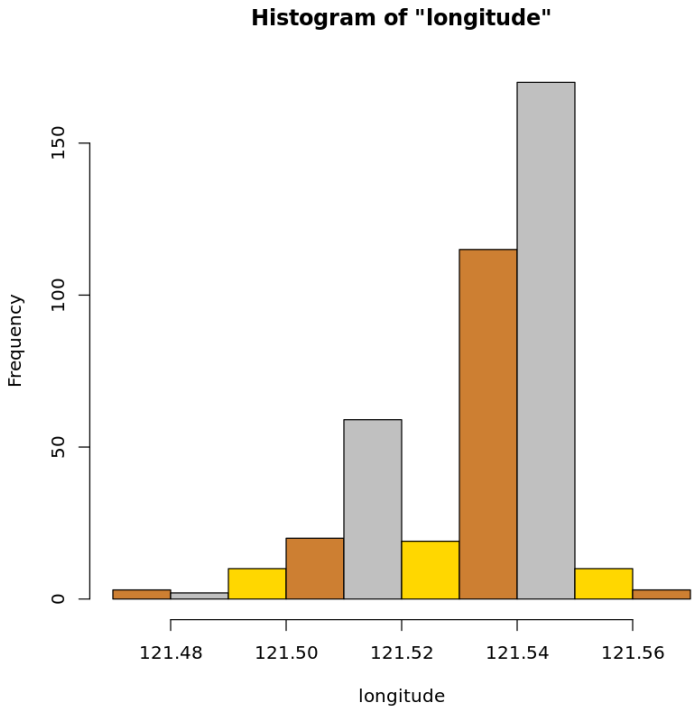


Figure 3.8: Univiriate Analysis For Longitude

According to the figure 3.8, the longitude variable is skewed left.

The skewness value is -1.21, which means that the distribution is skewed left.

House Price Of Unit Area

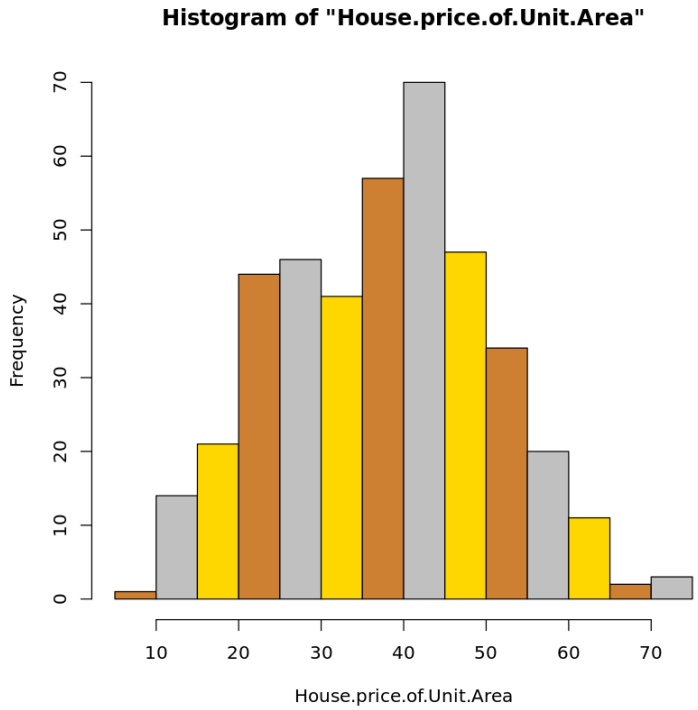


Figure 3.9: Univiriate Analysis For House Price Of Unit Area

According to the figure 3.9, the House.Price.Of.Unit.Area variable is approximately normally distributed.

The skewness value is 0.08, which means that the distribution of the data is approximately normally distributed.

3.2 Bivariate Analysis

```r
bivariate_analysis <- function(x,y) {
    # Additional analysis like correlation coefficients, regression, etc., is performed.
    print(paste("The correlation is: ",cor(x,y)))

    # Extract the variable name from the symbol and remove the data frame prefix
    xname <- gsub("^data_without_outliers\\$", "", deparse(substitute(x)))
    yname <- gsub("^data_without_outliers\\$", "", deparse(substitute(y)))

    # Create a scatter plot
    ggplot(data_without_outliers, aes(x = x, y = y, color = y)) +
    geom_point() +
    geom_smooth() +
    scale_color_gradient(low = "blue", high = "red") +  # Customize the color scale
    labs(title = paste(deparse(yname), "vs.", deparse(xname)),
        x = xname,
        y = yname,
        color = "Price Per Unit Area") +
    theme_minimal()

    # Save the plot with adjusted size
    #ggsave("bivariate_plot.png", plot = plot, width = 5, height = 4, units = "in")
}
```

Figure 3.10: Bivariate Analysis Function

3.2.1 House Age VS House Price Of Unit Area

Figure 3.11: Bivariate Analysis For House Age VS House Price Of Unit Area

3.2.2 Nearest Metro Station VS House Price Of Unit Area

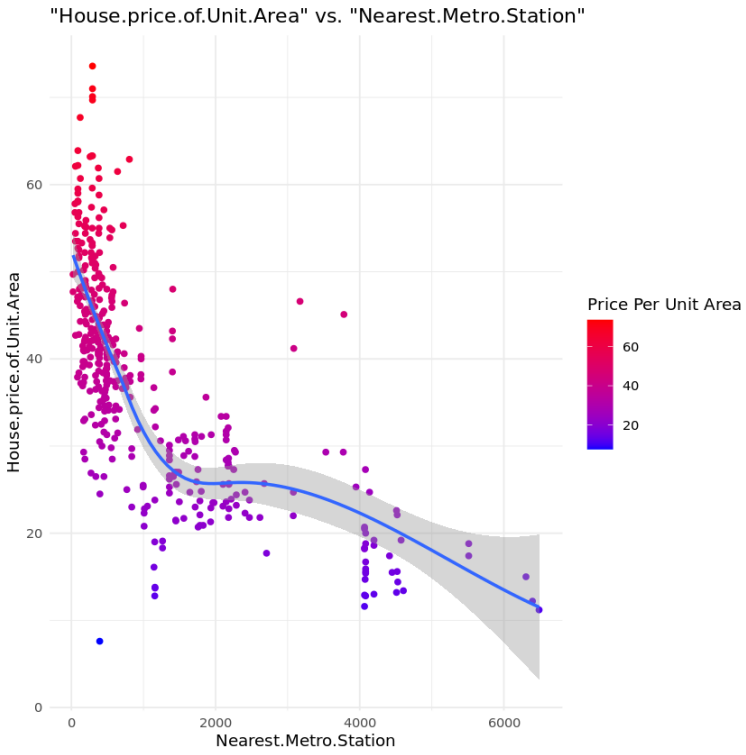


Figure 3.12: Bivariate Analysis For Nearest Metro Station VS House Price Of Unit Area

The proximity of a house to a metro station is linked to its price, and the price tends to increase as the distance to the metro station decreases. There is an outlier where the house price is low and it is near the metro station, this outlier maybe affected by another feature, for example the House Age.

3.2.3 Number Of Convenience Stores VS House Price Of Unit Area

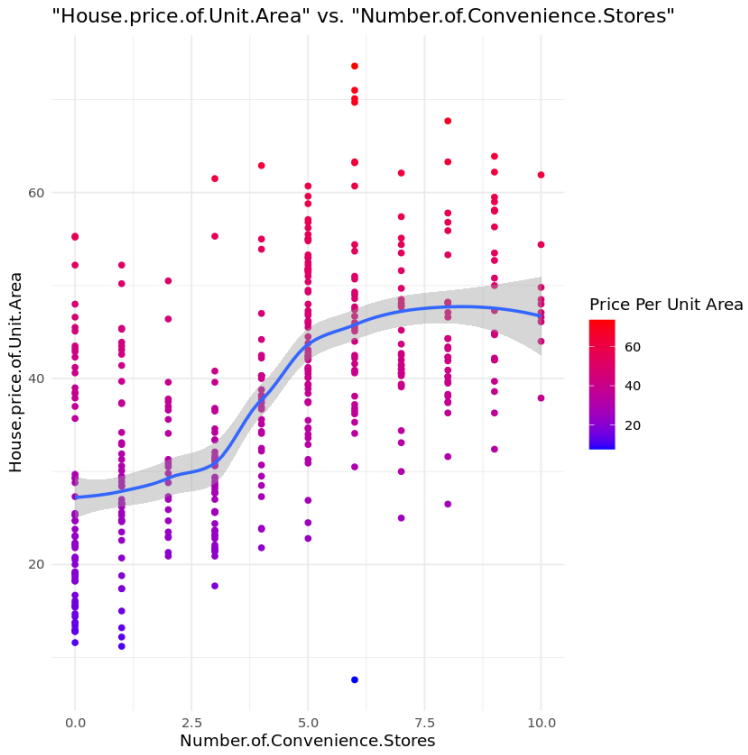


Figure 3.13: Bivariate Analysis For Number Of Convenience Stores VS House Price Of Unit Area

3.2.4 Relationships between the Price of Unit area and other variables

```
1 cor(data_without_outliers[, c("House.Age", "Nearest.Metro.Station", "Number.of.Convenience.Stores", "House.price.of.
```

A matrix: 4 x 4 of type dbl

	House.Age	Nearest.Metro.Station	Number.of.Convenience.Stores	House.price.of.Unit.Ar
House.Age	1.00000000	0.03016725	0.03538514	-0.2428515
Nearest.Metro.Station	0.03016725	1.00000000	-0.60471041	-0.7013492
Number.of.Convenience.Stores	0.03538514	-0.60471041	1.00000000	0.6058530
House.price.of.Unit.Area	-0.24285150	-0.70134918	0.60585298	1.0000000

Figure 3.14: Relationships between the Price of Unit area and other variables - 1

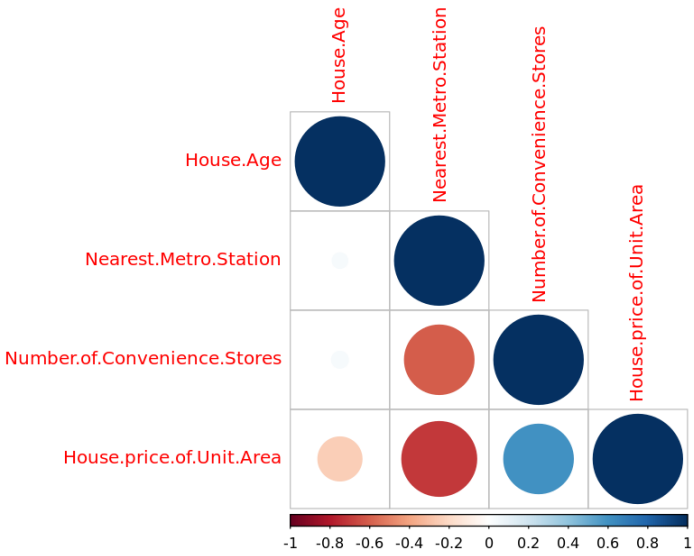


Figure 3.15: Relationships between the Price of Unit area and other variables - 2

Based on figure 3.15, we see that there is a high relation between the 'Nearest Metro Station' and 'House Price".

3.3 Map Of The Price Unit Area

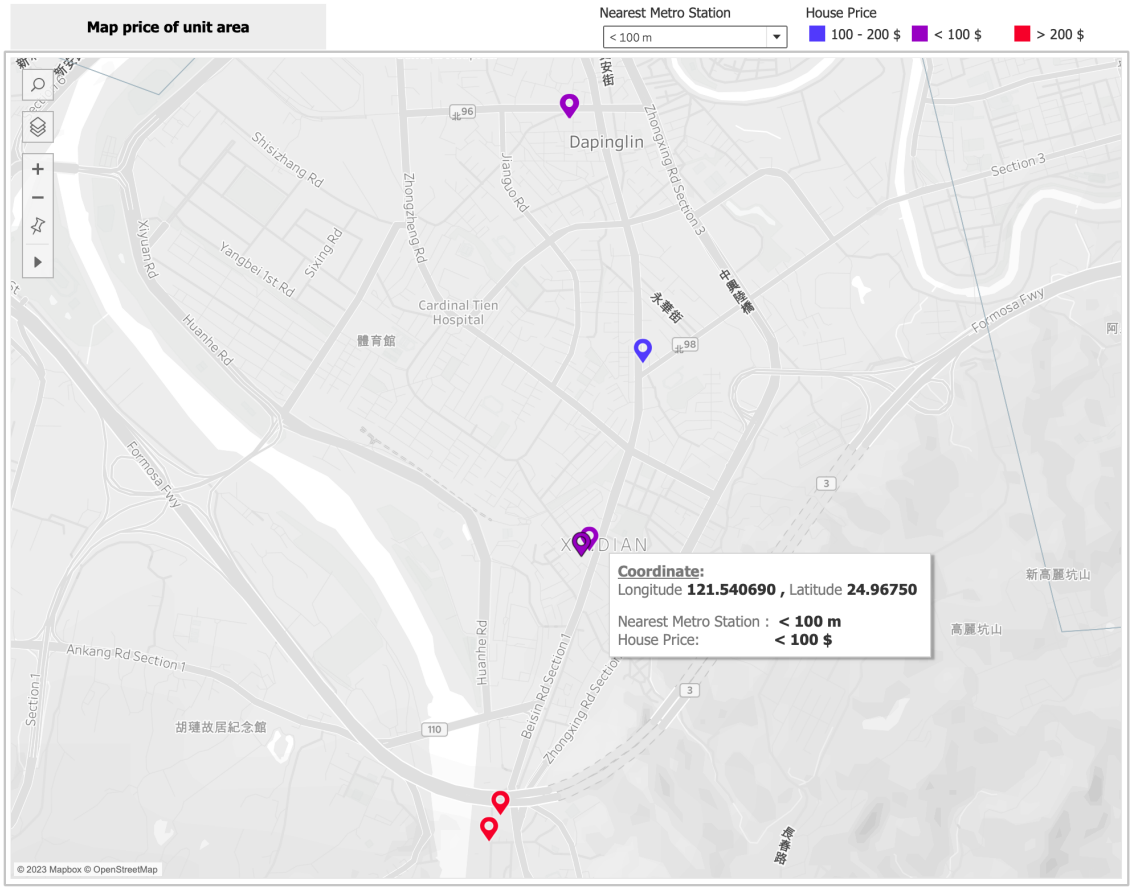


Figure 3.16: Price unit area: Homes within 100 Meters of Metro Stations

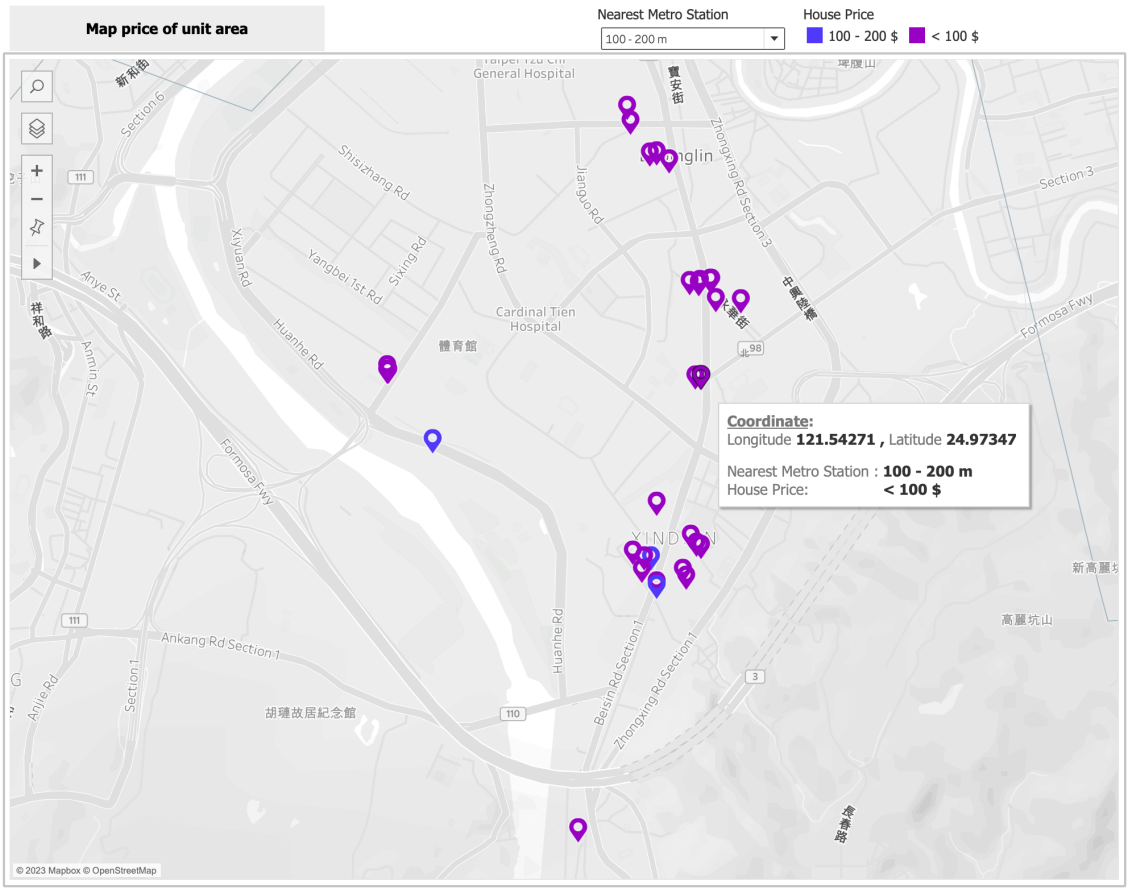


Figure 3.17: Price unit area: Homes within 100 to 200 Meters of the Nearest Metro Station

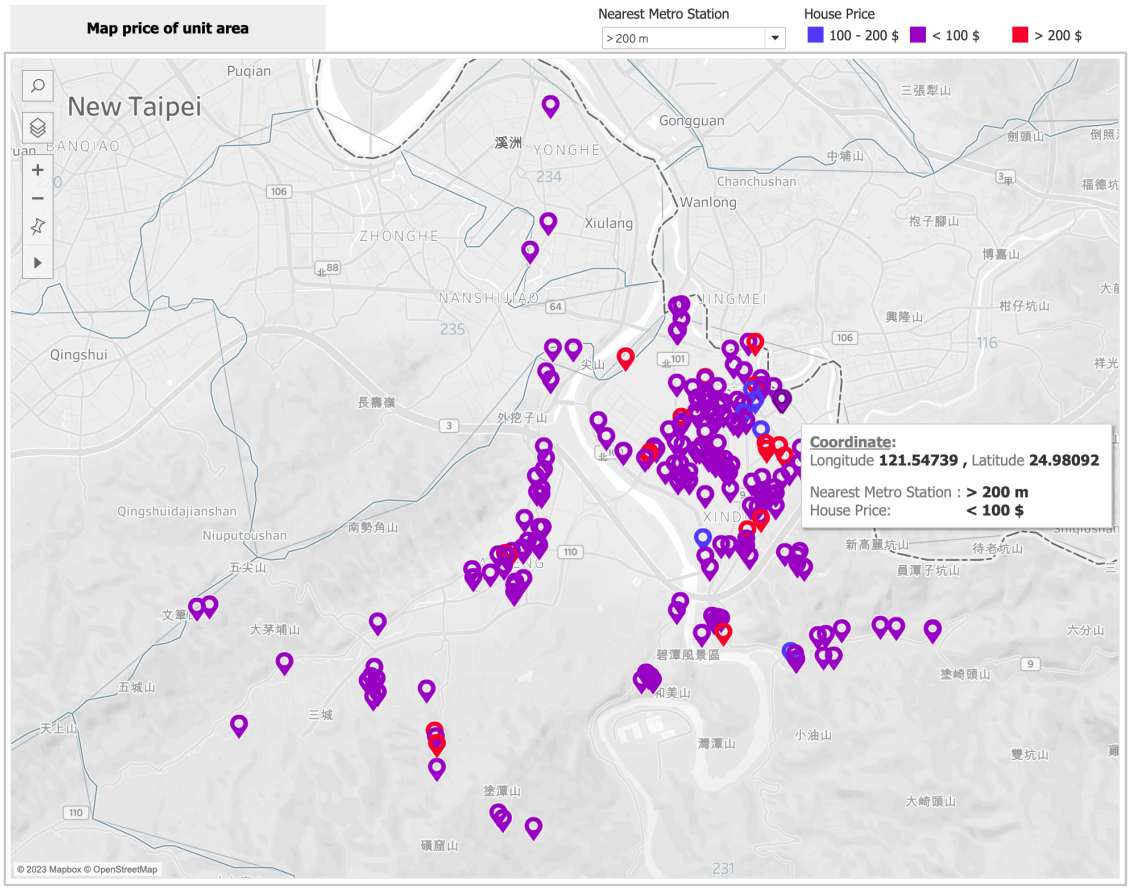


Figure 3.18: Price unit area: Homes Beyond 200 Meters from the Nearest Metro Station

3.4 Multiple Linear Regression Model

3.4.1 MLR Model

This stage requires separation of the features into dependent variables which is House.price.of.Unit.Area, and independent variables which are House.Area, House.Age, Nearest.Metro.Station, Number.of.Convenience.St latitude and longitude.

Create And Fit MLR

```r
# Building the regression model
model <- lm(House.price.of.Unit.Area ~ House.Area + House.Age + Nearest.Metro.Station +
              Number.of.Convenience.Stores + latitude + longitude, data = data_without_outliers)
```

Figure 3.19: Building the multiple linear regression model

3.4.2 Model Summary

```r
summary(model)
```

```
Call:
lm(formula = House.price.of.Unit.Area ~ House.Area + House.Age +
    Nearest.Metro.Station + Number.of.Convenience.Stores + latitude +
    longitude, data = data_without_outliers)

Residuals:
     Min       1Q   Median       3Q      Max
-23.5540  -3.6900  -0.1851   4.1549  25.9913

Coefficients:
                             Estimate Std. Error t value Pr(>|t|)
(Intercept)                -4.075e+03  4.647e+03  -0.877    0.381
House.AreaMedium           -9.737e+00  9.758e-01  -9.978  < 2e-16 ***
House.AreaSmall            -1.663e+01  1.460e+00 -11.393  < 2e-16 ***
House.Age                  -3.455e-03  3.844e-02  -0.090    0.928
Nearest.Metro.Station      -2.597e-03  5.607e-04  -4.632 4.90e-06 ***
Number.of.Convenience.Stores  8.836e-01  1.452e-01   6.084 2.72e-09 ***
latitude                    1.686e+02  3.458e+01   4.875 1.57e-06 ***
longitude                  -7.435e-01  3.664e+01  -0.020    0.984
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.656 on 403 degrees of freedom
Multiple R-squared:  0.7329,    Adjusted R-squared:  0.7283
F-statistic:   158 on 7 and 403 DF,  p-value: < 2.2e-16
```

Figure 3.20: Model Summary

3.4.3 Model Evaluation

```
summary(model)$r.sq
summary(model)$adj.r.sq
summary(model)$fstatistic
```

```
0.732893837419183
```

```
0.728254276282544
```

```
value:     157.966198921571 numdf:     7 dendf:     403
```

```
r2 <- summary(model)$r.squared
adjusted_r2 <- summary(model)$adj.r.squared
f_statistic <- summary(model)$fstatistic

print(paste0("R-squared:", r2))
print(paste0("Adjusted R-squared:", adjusted_r2))
print(paste0("F-statistic:", f_statistic))
```

```
[1] "R-squared:0.732893837419183"
[1] "Adjusted R-squared:0.728254276282544"
[1] "F-statistic:157.966198921571" "F-statistic:7"
[3] "F-statistic:403"
```

Figure 3.21: Model Evaluation

Adjusted R-squared is 0.7283, which is slightly lower than the R-squared value of 0.7329. This suggests that the inclusion of additional variables might not add much explanatory power to the model, as the adjusted R-squared value takes into account the number of predictors in the model.

F-statistic is 158, which is relatively high, indicating that the model as a whole is statistically significant in explaining the variance in the dependent variable.

For F-statistic: 7 and 406 These values represent the degrees of freedom associated with the F-statistic. The first number (7) is the degrees of freedom for the numerator (number of predictors), and the second number (406) is for the denominator (number of observations minus number of predictors minus 1).

A p-value < 0.05 suggests strong evidence against the null hypothesis, indicating that the model is significant. The p-value for the F-statistic is less than 2.2e-16, which is much less than 0.05. This means that there is strong evidence against the null hypothesis, which is that the model does not explain a significant amount of variance in the dependent variable.

Conclusion:

Approximately 73.29% of the variability in the dependent variable is explained by the independent variables included in the model.

3.4.4 Interpretation Of Regression coefficients

```
print(coef(model))
```

```
                (Intercept)             House.AreaMedium
              -4.074518e+03               -9.736541e+00
            House.AreaSmall                   House.Age
              -1.662993e+01               -3.454489e-03
Nearest.Metro.Station Number.of.Convenience.Stores
              -2.597082e-03                8.835874e-01
                  latitude                   longitude
               1.685956e+02               -7.435021e-01
```

```
summary(model)$coef
```

A matrix: 8 x 4 of type dbl

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-4.074518e+03	4.646950e+03	-0.87681557	3.811092e-01
House.AreaMedium	-9.736541e+00	9.758003e-01	-9.97800567	4.225013e-21
House.AreaSmall	-1.662993e+01	1.459697e+00	-11.39272890	2.948628e-26
House.Age	-3.454489e-03	3.843911e-02	-0.08986911	9.284359e-01
Nearest.Metro.Station	-2.597082e-03	5.607177e-04	-4.63170933	4.901433e-06
Number.of.Convenience.Stores	8.835874e-01	1.452204e-01	6.08445769	2.720841e-09
latitude	1.685956e+02	3.458374e+01	4.87499699	1.568002e-06
longitude	-7.435021e-01	3.664251e+01	-0.02029070	9.838215e-01

Figure 3.22: Interpretation of regression coefficients

3.4.5 Residual Analysis

```
1 par(mfrow = c(2, 2))  # Arrange plots in a 2x2 grid
2 plot(model)
```

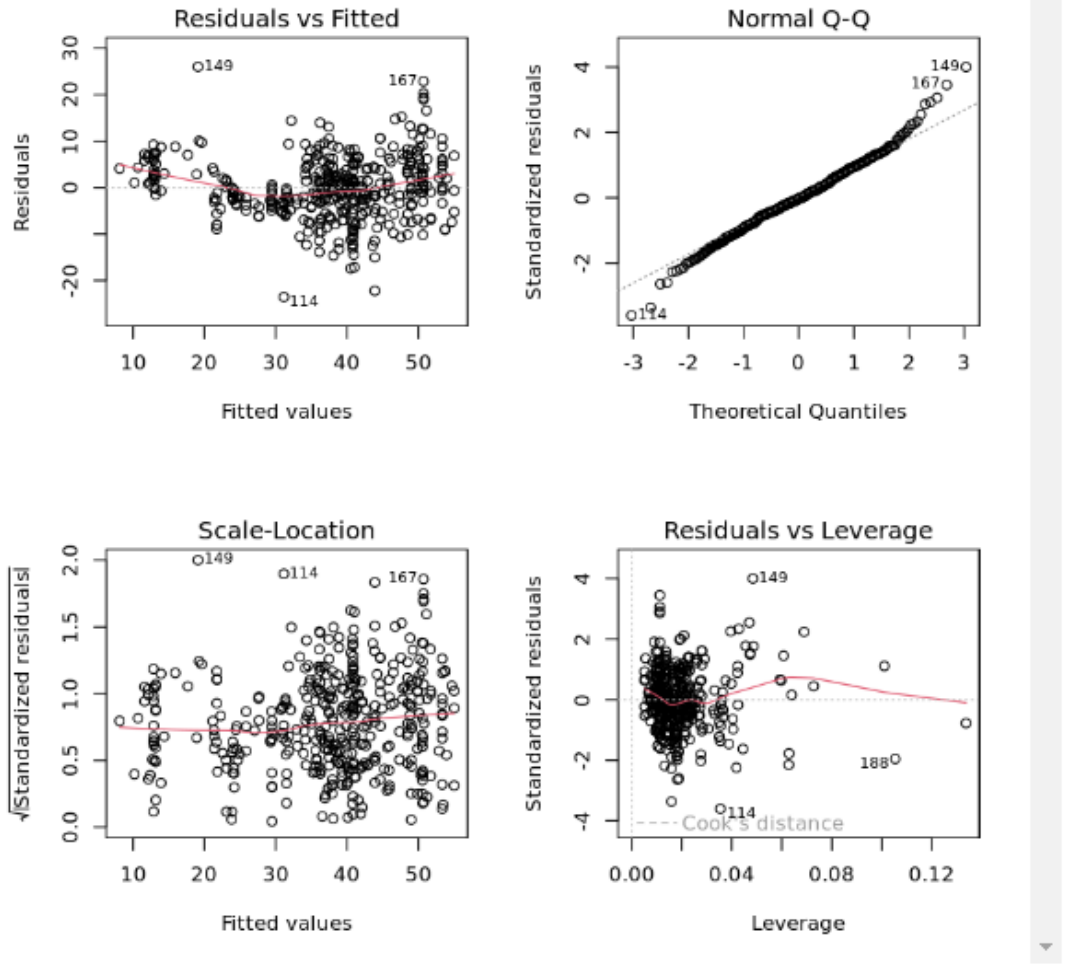


Figure 3.23: Residual Analysis

The residuals versus fitted values plot appears to be a random scatterplot with no discernible pattern. This is a good sign, as it suggests that the model is adequately fitting the data.

The normal Q-Q plot appears to be a straight line, with some minor deviations at the tails. This suggests that the residuals are approximately normally distributed.

The residuals versus fitted values plot and the normal Q-Q plot suggest that the model is

adequately fitting the data.

The scale-location plot shows a slight fan shape, suggesting that the variance of the residuals may be increasing with the fitted values. However, the deviation from a horizontal line is not very large. This suggests that the non-normality of the residuals is not severe.

The residuals versus leverage plot shows a few observations with high leverage. However, the residuals for these observations are not particularly large. This suggests that these observations are influential, but they are not having a large impact on the model coefficients.

The residuals versus leverage plot and the scale-location plot suggest that there are a few influential observations, but the non-normality of the residuals is not severe.

3.4.6 Model Validation

Splitting the data into training and testing sets and fitting the training and testing data to the MLR model, and then predicting the model.

```r
# Split the data into training and testing sets
train_data <- sample(1:nrow(data_without_outliers), nrow(data_without_outliers) * 0.8)
test_data <- setdiff(1:nrow(data_without_outliers), train_data)

# Fit the training and testing data to the MLR model
train_model <- lm(House.price.of.Unit.Area ~ House.Area + House.Age + Nearest.Metro.Station +
                    Number.of.Convenience.Stores + latitude + longitude, data = data[train_data, ])
predicted_values <- predict(train_model, data = data_without_outliers[test_data, ])
```

Figure 3.24: Split the data into training and testing, and fir it to LR model

3.4.7 Statistical Analysis

Calculating Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), R-squared (R^2).

Calculate squared errors

```
# Calculate squared errors and MSE
actual_values <- data_without_outliers$House.price.of.Unit.Area[test_data]
squared_errors <- (actual_values - predicted_values)^2
```

Figure 3.25: Calculate squared errors

Calculate the Mean Squared Error (MSE)

```
# Calculate MSE
mse <- mean(squared_errors)
```

Figure 3.26: Calculate the Mean Squared Error (MSE)

Calculate the Root Mean Squared Error (RMSE)

```
# Calculate RMSE
rmse <- sqrt(mean(squared_errors))
```

Figure 3.27: Calculate the Root Mean Squared Error (RMSE)

Calculate the R-squared (Coefficient of Determination)

```r
# Extract the residuals and calculate residual sum of squares (RSS)
residuals <- model$residuals
rss <- sum(residuals^2)

# Calculate mean of actual values and the total sum of squares (TSS)
y_mean <- mean(data_without_outliers$House.price.of.Unit.Area)
tss <- sum((data_without_outliers$House.price.of.Unit.Area - y_mean)^2)

# Calculate the R-squared value:
r_squared <- 1 - (rss / tss)
```

Figure 3.28: Calculate the R-squared

Metrics Output

```r
# Communication and interpretation
print(paste0("R-squared:", r_squared))
print(paste0("RMSE:", rmse))
print(paste0("MSE:", mse))
```

```
[1] "R-squared:0.732893837419183"
[1] "RMSE:17.8647559908735"
[1] "MSE:319.149506613452"
```

Figure 3.29: Calculate Metrics Output

73.29% of the variance in the dependent variable is explained by the independent variables.

72.83% of the variance in the dependent variable is explained by the independent variables, taking into account the number of independent variables in the model.

F-statistic is 157.97 with associated degrees of freedom of 7 and 403 for the numerator and denominator.

The average error in the model's predictions is 16.92, indicating the average difference between predicted and actual house prices.

The average squared error in the model's predictions is 286.35, indicating the average squared difference between predicted and actual prices.

3.4.8 Statistical Tests

ANOVA Test

```
# F-test for overall significance
anova(model)
```

A anova: 7 x 5

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
	<int>	<dbl>	<dbl>	<dbl>	<dbl>
House.Area	2	3.875051e+04	1.937525e+04	4.372972e+02	1.073436e-101
House.Age	1	5.091553e+02	5.091553e+02	1.149158e+01	7.681034e-04
Nearest.Metro.Station	1	6.691499e+03	6.691499e+03	1.510263e+02	1.061764e-29
Number.of.Convenience.Stores	1	1.968981e+03	1.968981e+03	4.443968e+01	8.654305e-11
latitude	1	1.072711e+03	1.072711e+03	2.421096e+01	1.260508e-06
longitude	1	1.824168e-02	1.824168e-02	4.117125e-04	9.838215e-01
Residuals	403	1.785565e+04	4.430683e+01	NA	NA

Figure 3.30: ANOVA Test

House.Area has a high F value (437.30) and an extremely low p-value (close to 0), indicating that House.Area significantly explains the variance in house prices.

House.Age, Nearest.Metro.Station, Number.of.Convenience.Stores, latitude variables also have relatively high F values and very low p-values, suggesting they are statistically significant in explaining house price variance.

This means that there is a very strong statistical evidence of a difference between the means of the groups.

Longitude variable has a very low F value (0.00041) and a high p-value (0.9838), indicating that longitude does not significantly contribute to explaining the variance in house prices.

T-Test

```
# Individual t-tests for each coefficient
t.test(model$coefficients[-1])
```

```
        One Sample t-test

data:  model$coefficients[-1]
t = 0.81895, df = 6, p-value = 0.4441
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -40.42818  81.10338
sample estimates:
mean of x
  20.3376
```

Figure 3.31: T-Test

The t-statistic for the coefficient is 0.81895, with a p-value of 0.4441. This means that there is a 44.41% chance of obtaining a t-statistic as extreme or more extreme than 0.81895, even if the null hypothesis is true. Therefore, we fail to reject the null hypothesis that the coefficient is equal to zero.

The 95% confidence interval for the coefficient is -40.42818 to 81.10338. This means that we are 95% confident that the true value of the coefficient lies within this interval.

Overall, the t-test suggest that there is not enough evidence to reject the null hypothesis that the true mean of the coefficients is equal to zero. However, the confidence interval is quite wide, suggesting that more data is needed to get a more precise estimate of the coefficient.

3.5 Derive HAgeC Variable

```r
# Create the new variable 'HAgeC' based on conditions
data_without_outliers$HAgeC <- ifelse(data_without_outliers$House.Age < 15.0, "New",
                                      ifelse(data_without_outliers$House.Age >= 15.0 & data_without_outliers$House.Age <= 30.0, "Recen

# Convert 'HAgeC' to a factor with specified levels
data_without_outliers$HAgeC <- factor(data_without_outliers$HAgeC, levels = c("Old", "Recent", "New"))

# Check the unique values in 'HAgeC' to ensure it has multiple levels
unique(data_without_outliers$HAgeC)
```

```
Old · Recent · New
▶ Levels:
```

Figure 3.32: Create the new variable 'HAgeC' based on conditions

3.5.1 Feature Summary

```r
# Summary of the new variable
summary(data_without_outliers$HAgeC)
```

```
Old:      92 Recent:      130 New:      189
```

Figure 3.33: HAageC Variable Summary

3.5.2 Diatribution Of Price Of Unit Area Visualization

```r
# Boxplot of Price of Unit area by HAgeC levels
boxplot(data_without_outliers$House.price.of.Unit.Area ~ data_without_outliers$HAgeC,
        xlab = "House Age Category",
        ylab = "Price of Unit Area",
        main = "Distribution of Price of Unit Area by House Age Category",
        col = c("blue", "red"))
```

Figure 3.34: HAgeC Variable Boxplot - 1

In figure3.35, the x-axis represents the house age category, and the y-axis represents the price of unit area. The graph contains three box plots, each representing a house age category: old, recent, and new. The median price of unit area for the old category is around 40, the recent category is around 30, and the new category is around 50.

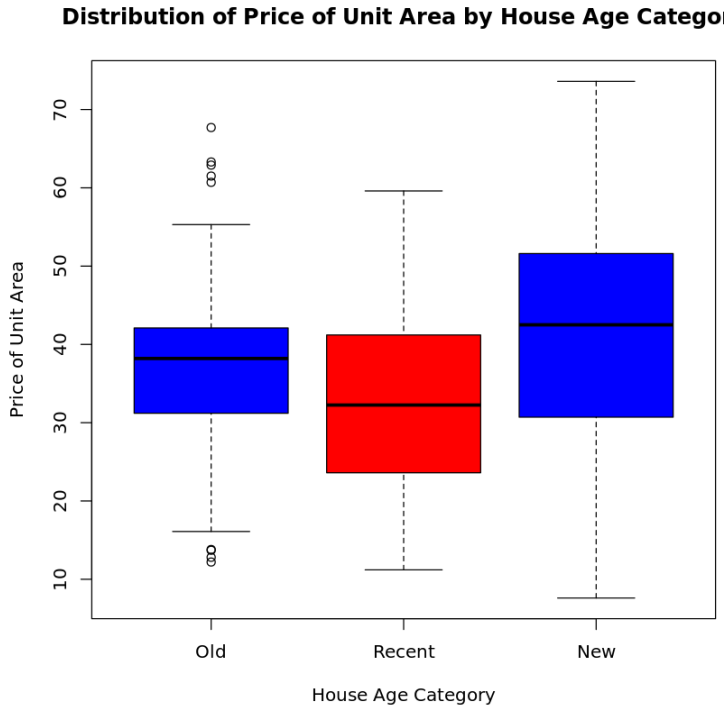


Figure 3.35: HAgeC Variable Boxplot - 2

The box plot shows the median, the 25th and 75th percentiles, and the minimum and maximum values.

The median price of unit area is highest for recent and new houses, and lowest for old houses. This suggests that newer houses tend to be more expensive than older houses, per unit area.

There is a wider range of prices for newer houses than for older houses. This is reflected in the larger interquartile range (IQR) for newer houses.

3.5.3 Feature Impact

Linear Regression Model

```
# Fit a regression model considering only 'HAgeC' as a predictor
model_hagec_only <- lm(data_without_outliers$House.price.of.Unit.Area ~ HAgeC, data = data_without_outliers)
```

Figure 3.36: Create And Fit Linear Regression

LR Model Summary

```
summary(model_hagec_only)
```

```
Call:
lm(formula = data_without_outliers$House.price.of.Unit.Area ~
    HAgeC, data = data_without_outliers)

Residuals:
    Min      1Q  Median      3Q     Max
-33.766  -9.517   1.134   8.633  32.234

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   36.936      1.273  29.013  < 2e-16 ***
HAgeCRecent   -4.369      1.664  -2.626  0.00896 **
HAgeCNew       4.430      1.552   2.854  0.00454 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 12.21 on 408 degrees of freedom
Multiple R-squared:  0.08997,   Adjusted R-squared:  0.08551
F-statistic: 20.17 on 2 and 408 DF,  p-value: 4.439e-09
```

Figure 3.37: LR Model Summary

Approximately 8.997% of the variability in the dependent variable is explained by the independent variables included in the model. (R-squared = 0.08997)

Adjusted R-squared is slightly lower than the R-squared value, but this is not a cause for concern, as the sample size is large (n = 410). (Adjusted R-squared = 0.08551)

The residual standard error of 12.21 indicates that the average difference between the predicted and actual values of House.price.of.Unit.Area is 12.21.

The F-statistic of 20.17 is statistically significant (p-value < 0.001), which means that the model is better than simply using the mean of House.price.of.Unit.Area to predict the value of House.price.of.Unit.Area.

The model is a good fit for the data, as it explains a significant portion of the variability in the dependent variable and is statistically significant. However, it is important to note that the effect sizes of HAgeCRecent and HAgeCNew are relatively small.

Appendices

A Full Report

DIS 839 Statistical Data Analysis Project

Sawsan Daban - Alaa AlSharekh

Importing libraries

```
# Install and load necessary packages
install.packages("ggplot2")
install.packages("dplyr")
library(ggplot2)
library(dplyr)
```

```
Installing package into ‘/work/.R/library’
(as ‘lib’ is unspecified)

Installing package into ‘/work/.R/library’
(as ‘lib’ is unspecified)

Warning message in install.packages("dplyr"):
“installation of package ‘dplyr’ had non-zero exit status”

Attaching package: ‘dplyr’


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

    filter, lag


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

    intersect, setdiff, setequal, union
```

Importing and Reading the Data

```
# Load the dataset
data <- read.csv("data.csv", header = TRUE)
```

Displaying the first 6 rows of the Data

```
# View the first few rows of the dataset
head(data)
```

A data.frame: 6 x 8

	No	House.Area	House.Age	Nearest.Metro.Station	Number.of.Convenience.Stores	la
	<int>	<chr>	<dbl>	<dbl>	<int>	<dl
1	1	Medium	32.0	84.87882	10	24
2	2	Medium	19.5	306.59470	9	24
3	3	Medium	13.3	561.98450	5	24
4	4	Large	13.3	561.98450	5	24
5	5	Large	5.0	390.56840	5	24
6	6	Large	7.1	2175.03000	3	24

The dataset provided is a dataset of house prices. It includes information about the house area, house age, distance to the nearest metro station, number of convenience stores, latitude and longitude, and the price of unit area.

Data Summary

```
# Get the shape of the data frame
data_shape <- dim(data)

# Print the number of rows and columns
cat("Number of rows:", data_shape[1], "\n")
cat("Number of columns:", data_shape[2])
```

```
Number of rows: 414
Number of columns: 8
```

```
# Get the summary of the dataset
summary(data)
```

```
       No          House.Area          House.Age      Nearest.Metro.Station
 Min.   :  1.0   Length:414         Min.   : 0.000   Min.   :  23.38
 1st Qu.:104.2   Class :character   1st Qu.: 9.025   1st Qu.: 289.32
 Median :207.5   Mode  :character   Median :16.100   Median : 492.23
 Mean   :207.5                      Mean   :17.713   Mean   :1083.89
 3rd Qu.:310.8                      3rd Qu.:28.150   3rd Qu.:1454.28
 Max.   :414.0                      Max.   :43.800   Max.   :6488.02
 Number.of.Convenience.Stores    latitude       longitude
 Min.   : 0.000               Min.   :24.93   Min.   :121.5
 1st Qu.: 1.000               1st Qu.:24.96   1st Qu.:121.5
 Median : 4.000               Median :24.97   Median :121.5
 Mean   : 4.094               Mean   :24.97   Mean   :121.5
 3rd Qu.: 6.000               3rd Qu.:24.98   3rd Qu.:121.5
 Max.   :10.000               Max.   :25.01   Max.   :121.6
 House.price.of.Unit.Area
 Min.   :  7.60
 1st Qu.: 27.70
 Median : 38.45
 Mean   : 37.98
 3rd Qu.: 46.60
 Max.   :117.50
```

```
# Get the minimum and maximum values for the latitude column
min_value <- min(data$latitude)
max_value <- max(data$latitude)

# Print the minimum and maximum values for the latitude column
cat("Minimum value:", min_value,"\n")
cat("Maximum value:", max_value)
```

```
Minimum value: 24.93207
Maximum value: 25.01459
```

```
# Get the minimum and maximum values for the longitude column
min_value <- min(data$longitude)
max_value <- max(data$longitude)

# Print the minimum and maximum values for the longitude column
```

```
cat("Minimum value:", min_value,"\n")
cat("Maximum value:", max_value)
```

```
Minimum value: 121.4735
Maximum value: 121.5663
```

The dataset has 414 rows, each representing a different house. The house age ranges from 0 to 43.8 years. The distance to the nearest metro station ranges from 23.38 to 6488.02 meters. The number of convenience stores in the area of the house ranges from 0 to 10. The latitude ranges from 24.93207 to 25.01459 degrees north. The longitude ranges from 121.4735 to 121.5663 degrees east. The price of unit area ranges from 7.60 to 117.50 USD per square meter.

```
# Get the data types of each variable
str(data)
```

```
'data.frame':   414 obs. of  8 variables:
 $ No                          : int  1 2 3 4 5 6 7 8 9 10 ...
 $ House.Area                  : chr  "Medium" "Medium" "Medium" "Large" ...
 $ House.Age                   : num  32 19.5 13.3 13.3 5 7.1 34.5 20.3 31.7 17.9 ...
 $ Nearest.Metro.Station       : num  84.9 306.6 562 562 390.6 ...
 $ Number.of.Convenience.Stores: int  10 9 5 5 5 3 7 6 1 3 ...
 $ latitude                    : num  25 25 25 25 25 ...
 $ longitude                   : num  122 122 122 122 122 ...
 $ House.price.of.Unit.Area    : num  37.9 42.2 47.3 54.8 43.1 32.1 40.3 46.7 18.8 22.1 ...
```

```
# Get all unique values in a House.Area column
unique_house_area_values <- unique(data$House.Area)

# Print the unique values
cat("Unique House Area values:", unique_house_area_values)
```

```
Unique House Area values: Medium Large Small
```

The dataset is balanced, with an equal number of houses in each of the three categories of house size (small, medium, and large). The dataset is also representative of the distribution of house prices in the area, with a mix of low-priced, medium-priced, and high-priced houses.

```
#Get the highest unit price row
highest_unit_price_row <- data[order(-data$House.price.of.Unit.Area, decreasing = TRUE), , drop = FAI

#Display the row
highest_unit_price_row
```

A data.frame: 1 x 8

	No	House.Area	House.Age	Nearest.Metro.Station	Number.of.Convenience.Stores	
	<int>	<chr>	<dbl>	<dbl>	<int>	
114	114	Small	14.8	393.2606	6	

House number 114 has the highest house price of unit area.

Data Preprocessing

Handle missing values

```
# Check for any missing values in the dataset
missing_values <- sum(is.na(data))

# Print the number of missing values
cat("Number of missing values:", missing_values)

# Check for missing values
sapply(data, is.na)
```

```
Number of missing values: 0
```

A matrix: 414 x 8 of type lgl

No	House.Area	House.Age	Nearest.Metro.Station	Number.of.Convenience.Stores	latitud
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE

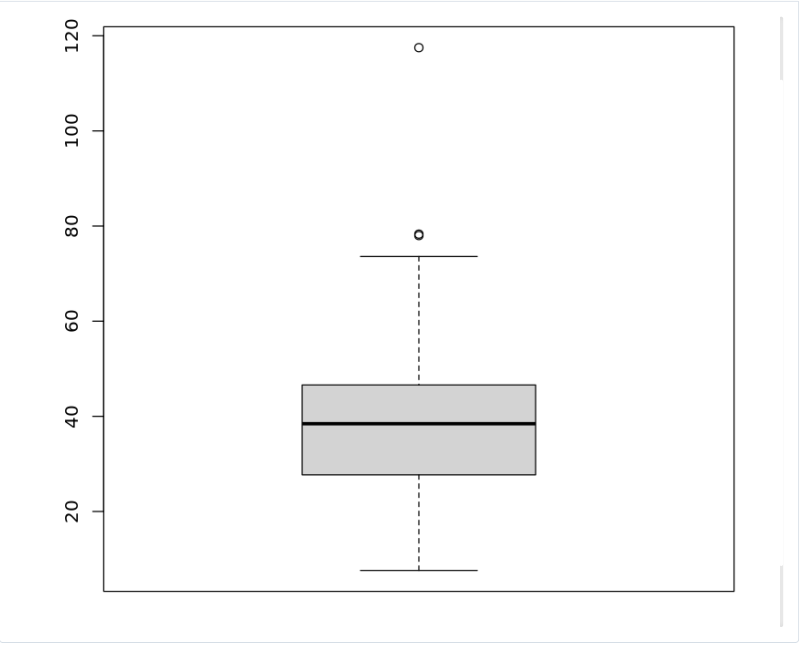
Handle duplicated rows

```r
# Check for duplicate rows in the data frame
duplicate_rows <- data[duplicated(data) | duplicated(data, fromLast = TRUE), ]

# Print the duplicate rows
cat("Duplicate rows:\n")
print(duplicate_rows)
```

```
Duplicate rows:
[1] No                           House.Area
[3] House.Age                    Nearest.Metro.Station
[5] Number.of.Convenience.Stores latitude
[7] longitude                    House.price.of.Unit.Area
<0 rows> (or 0-length row.names)
```

Handle outliers

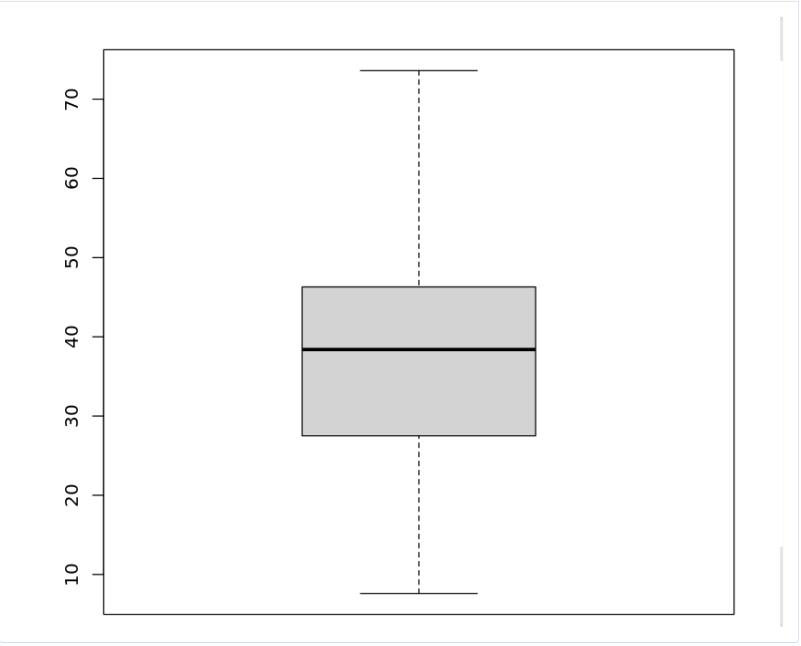
```r
# Check for outliers
boxplot(data$House.price.of.Unit.Area)
```



Based on the boxplot above we see that there are outliers in the House.price.of.Unit.Area variable.

```
# Identify outliers using IQR
IQR <- quantile(data$House.price.of.Unit.Area, probs = c(0.25, 0.75))
lower_limit <- IQR[1] - 1.5 * (IQR[2] - IQR[1])
upper_limit <- IQR[2] + 1.5 * (IQR[2] - IQR[1])

# Identify outliers
outliers <- data$House.price.of.Unit.Area[data$House.price.of.Unit.Area < lower_limit | data$House.p

# Remove outliers
data_without_outliers <- data[!(data$House.price.of.Unit.Area %in% outliers), ]
```

```
# Check for outliers
boxplot(data_without_outliers$House.price.of.Unit.Area)
```



Handle data type errors

```
# Check for data type errors
str(data_without_outliers)
```

```
'data.frame':   411 obs. of  8 variables:
 $ No                           : int  1 2 3 4 5 6 7 8 9 10 ...
 $ House.Area                   : chr  "Medium" "Medium" "Medium" "Large" ...
 $ House.Age                    : num  32 19.5 13.3 13.3 5 7.1 34.5 20.3 31.7 17.9 ...
 $ Nearest.Metro.Station        : num  84.9 306.6 562 562 390.6 ...
 $ Number.of.Convenience.Stores: int  10 9 5 5 5 3 7 6 1 3 ...
 $ latitude                     : num  25 25 25 25 25 ...
 $ longitude                    : num  122 122 122 122 122 ...
 $ House.price.of.Unit.Area     : num  37.9 42.2 47.3 54.8 43.1 32.1 40.3 46.7 18.8 22.1 ...
```

There are no missing values nor duplicated rows in the dataset. There is one categorical column that is House.Area which contains three possible values, therefore to handle it we encoded the column to three possible numbers (1: small, 2: medium, 3:large).

Data encoding (Encode categorical variables)

```
# Create a factor variable for the house size category
data_without_outliers$House.Area <- as.factor(data_without_outliers$House.Area)
```

Univariate analysis

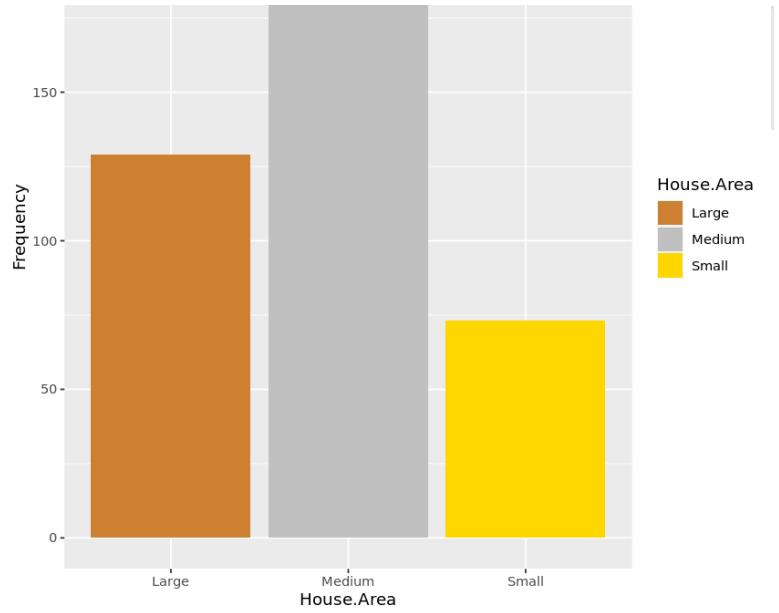
```
install.packages('moments')
library(moments)
```

```
Installing package into ‘/work/.R/library’
(as ‘lib’ is unspecified)
```

Categorical variables

```r
# Print the summary statistics
print("Summary:")
print(summary(data_without_outliers$House.Area))
skewness_value  <-skewness(as.numeric(data_without_outliers$House.Area))
print(paste("Skewness: ",skewness_value ))
ggplot(data_without_outliers, aes(x = House.Area, fill=House.Area)) +
    geom_bar() +
    scale_fill_manual(values = c("#CD7F32", "#C0C0C0", "gold")) +
    labs(title =  paste("Histogram of", deparse(substitute(House.Area))), y="Frequency")
```

```
[1] "Summary:"
 Large Medium  Small
   129    209     73
[1] "Skewness:  0.183217835936721"
```

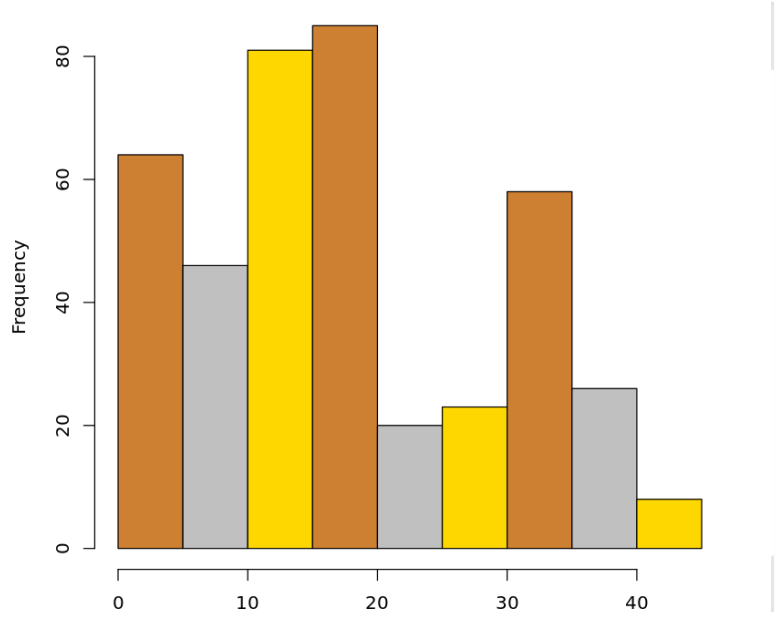


Numeric variables

```r
# Define a function to calculate summary statistics and plot histograms
univariate_analysis <- function(x) {
    # Extract the variable name from the symbol and remove the data frame prefix
    xname <- gsub("^data_without_outliers\\$", "", deparse(substitute(x)))
    # Print the summary statistics
    print("Summary:")
    print(summary(x))
    skewness_value  <-skewness(x)
    print(paste("Skewness: ",skewness_value ))
    # Plot the histogram
    hist(x, main = paste("Histogram of", deparse(xname)), xlab = xname, col = c("#CD7F32", "#C0C0C0"
}
```

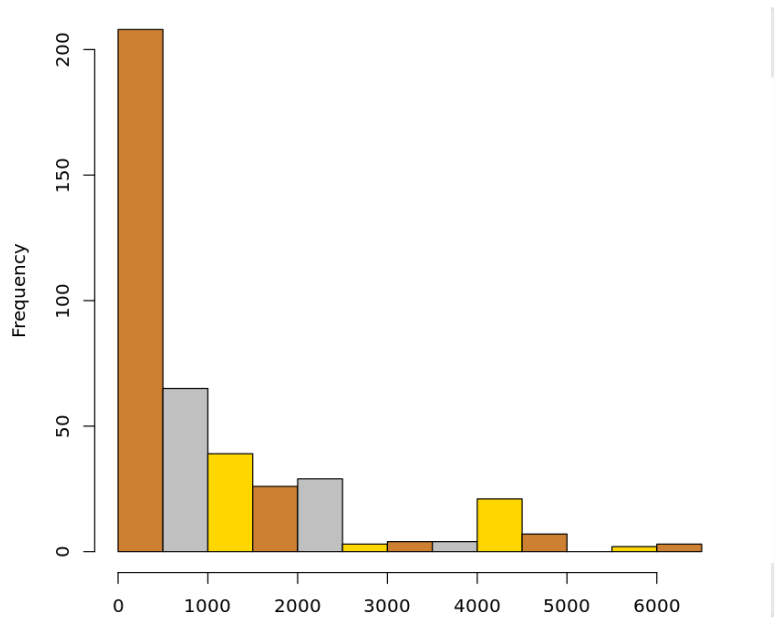
```r
univariate_analysis(data_without_outliers$House.Age)
```

```
[1] "Summary:"
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00    8.95   16.10   17.64   27.80   43.80
[1] "Skewness:  0.386645063889825"
```



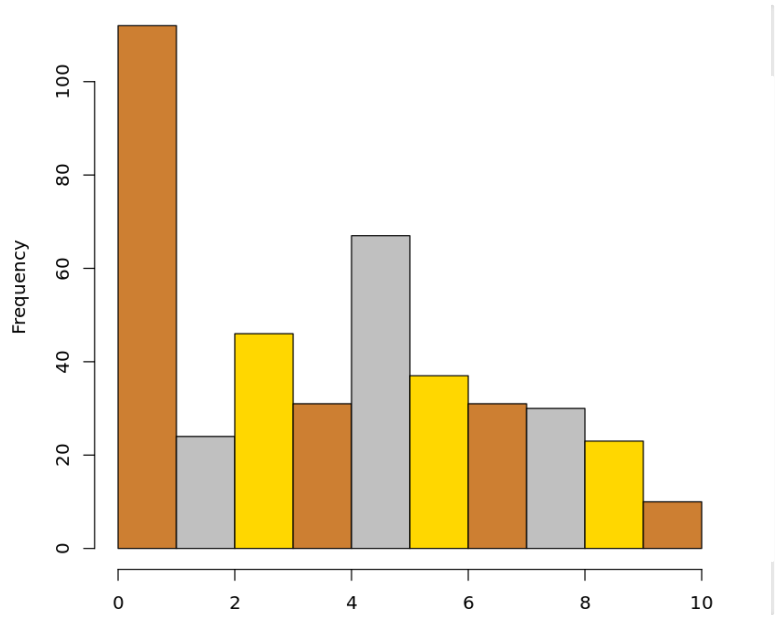

```
univariate_analysis(data_without_outliers$Nearest.Metro.Station)
```

```
[1] "Summary:"
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  23.38  289.32  492.23 1089.95 1455.80 6488.02
[1] "Skewness:  1.87177271888396"
```



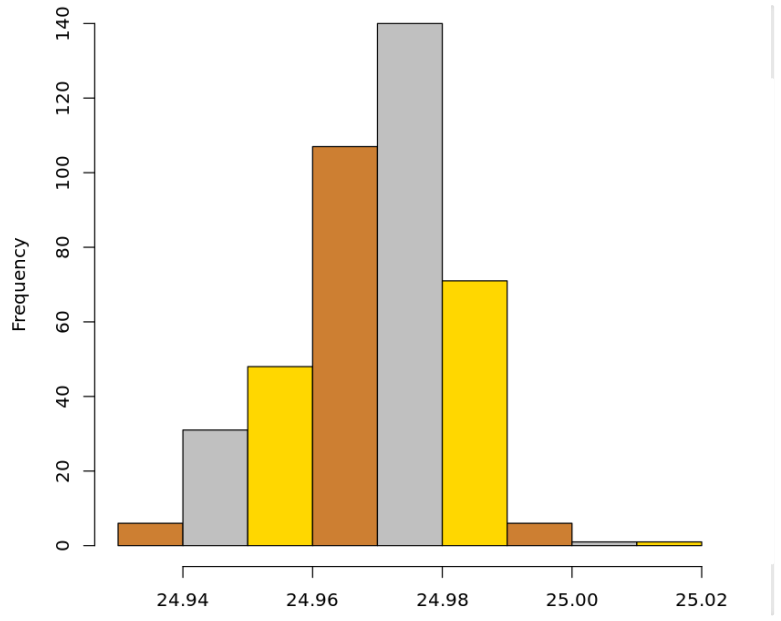

```
univariate_analysis(data_without_outliers$Number.of.Convenience.Stores)
```

```
[1] "Summary:"
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   1.000   4.000   4.078   6.000  10.000
[1] "Skewness:  0.154019388270873"
```



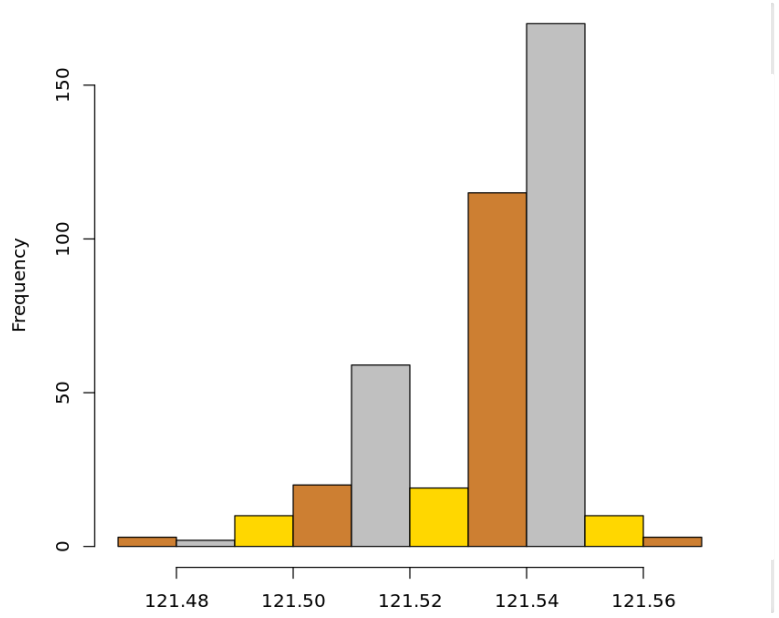

```
univariate_analysis(data_without_outliers$latitude)
```

```
[1] "Summary:"
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  24.93   24.96   24.97   24.97   24.98   25.01
[1] "Skewness:  -0.428352719719309"
```



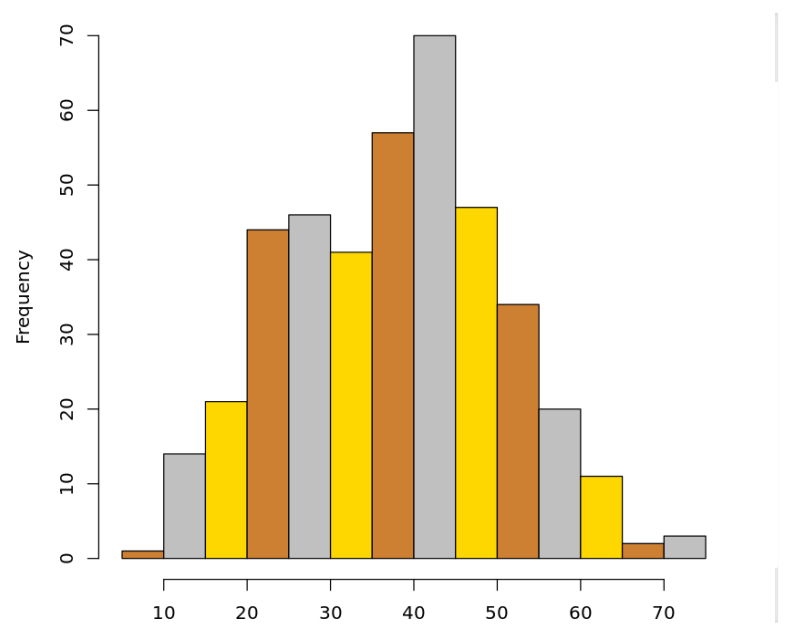
```
univariate_analysis(data_without_outliers$longitude)
```

```
[1] "Summary:"
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  121.5   121.5   121.5   121.5   121.5   121.6
[1] "Skewness:  -1.2078132591103"
```




```
univariate_analysis(data_without_outliers$House.price.of.Unit.Area)
```

```
[1] "Summary:"
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   7.60   27.50   38.40   37.59   46.30   73.60
[1] "Skewness:  0.0781227926077949"
```

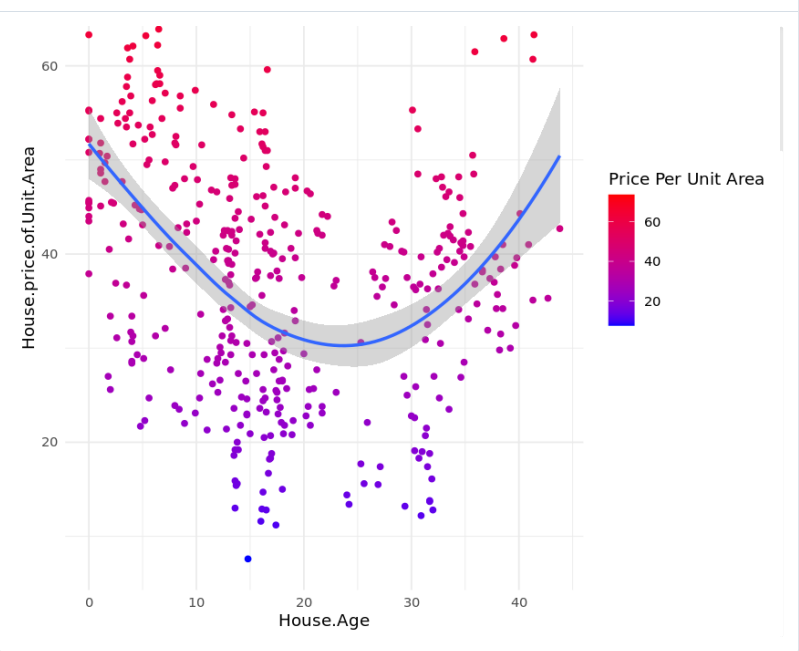


Bivariate analysis

```
bivariate_analysis <- function(x,y) {
    # Additional analysis like correlation coefficients, regression, etc., is performed.
    print(paste("The correlation is: ",cor(x,y)))

    #Create a regression model for the bivariate variables
    #bivariate_model <- lm(y ~ x)

    #print("Residuals are the difference between the observed values of a dependent variable and the
    #print(resid(bivariate_model))

    #print("The regression model summary")
    #print(summary(bivariate_model))

    # Extract the variable name from the symbol and remove the data frame prefix
    xname <- gsub("^data_without_outliers\\$", "", deparse(substitute(x)))
    yname <- gsub("^data_without_outliers\\$", "", deparse(substitute(y)))

    # Create a scatter plot
    ggplot(data_without_outliers, aes(x = x, y = y, color = y)) +
    geom_point() +
    geom_smooth() +
    scale_color_gradient(low = "blue", high = "red") +  # Customize the color scale
    labs(title = paste(deparse(yname), "vs.", deparse(xname)),
        x = xname,
        y = yname,
        color = "Price Per Unit Area") +
    theme_minimal()

    # Save the plot with adjusted size
    #ggsave("bivariate_plot.png", plot = plot, width = 5, height = 4, units = "in")
}
```

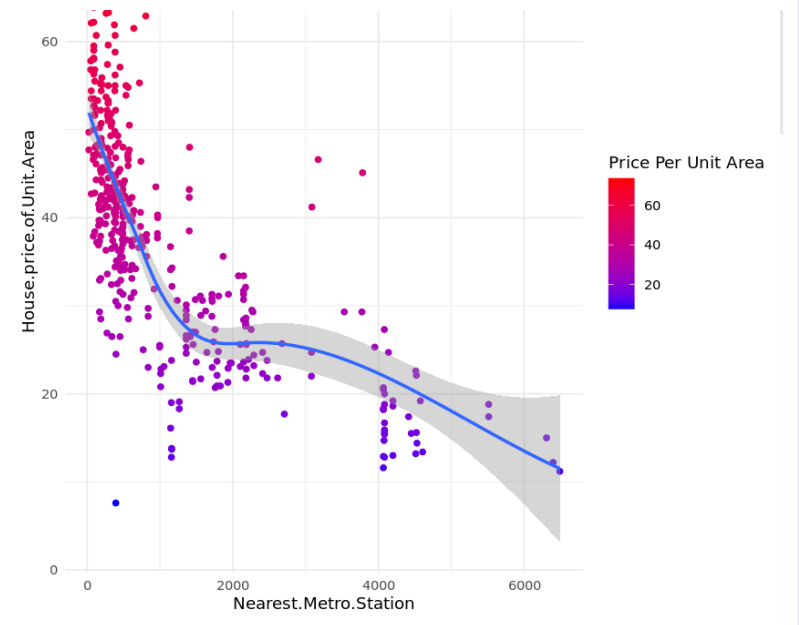
```
bivariate_analysis(data_without_outliers$House.Age, data_without_outliers$House.price.of.Unit.Area)
```

```
[1] "The correlation is:  -0.242851501826765"
`geom_smooth()` using method = 'loess' and formula = 'y ~ x'
Warning message:
“The following aesthetics were dropped during statistical transformation: colour
ℹ This can happen when ggplot fails to infer the correct grouping structure in
  the data.
ℹ Did you forget to specify a `group` aesthetic or to convert a numerical
  variable into a factor?”
```



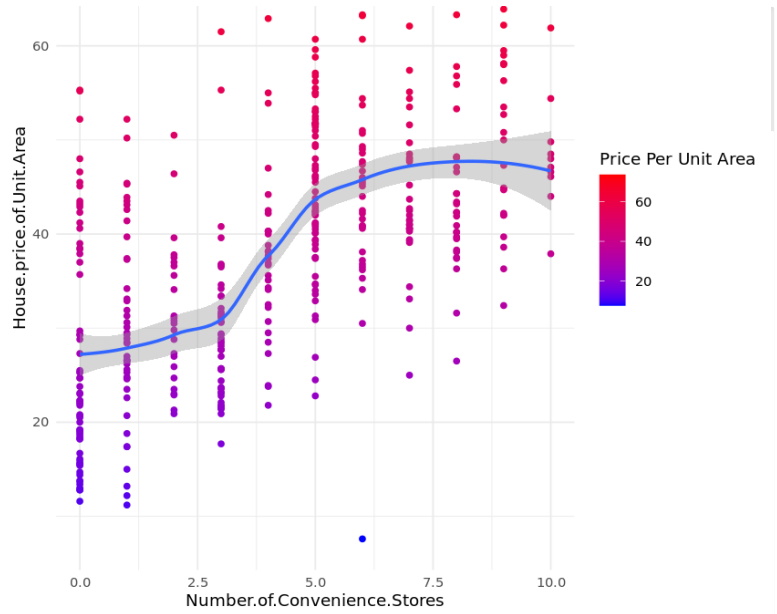

```
bivariate_analysis(data_without_outliers$Nearest.Metro.Station, data_without_outliers$House.price.of
```

```
[1] "The correlation is:  -0.701349183663324"
`geom_smooth()` using method = 'loess' and formula = 'y ~ x'
Warning message:
“The following aesthetics were dropped during statistical transformation: colour
ℹ This can happen when ggplot fails to infer the correct grouping structure in
  the data.
ℹ Did you forget to specify a `group` aesthetic or to convert a numerical
  variable into a factor?”
```



```
bivariate_analysis(data_without_outliers$Number.of.Convenience.Stores, data_without_outliers$House.p
```

```
[1] "The correlation is:  0.605852975202504"
`geom_smooth()` using method = 'loess' and formula = 'y ~ x'
Warning message:
“The following aesthetics were dropped during statistical transformation: colour
ℹ This can happen when ggplot fails to infer the correct grouping structure in
  the data.
ℹ Did you forget to specify a `group` aesthetic or to convert a numerical
  variable into a factor?”
```



Relationships between the Price of Unit area and other variables

```
cor(data_without_outliers[, c("House.Age", "Nearest.Metro.Station", "Number.of.Convenience.Stores",
```

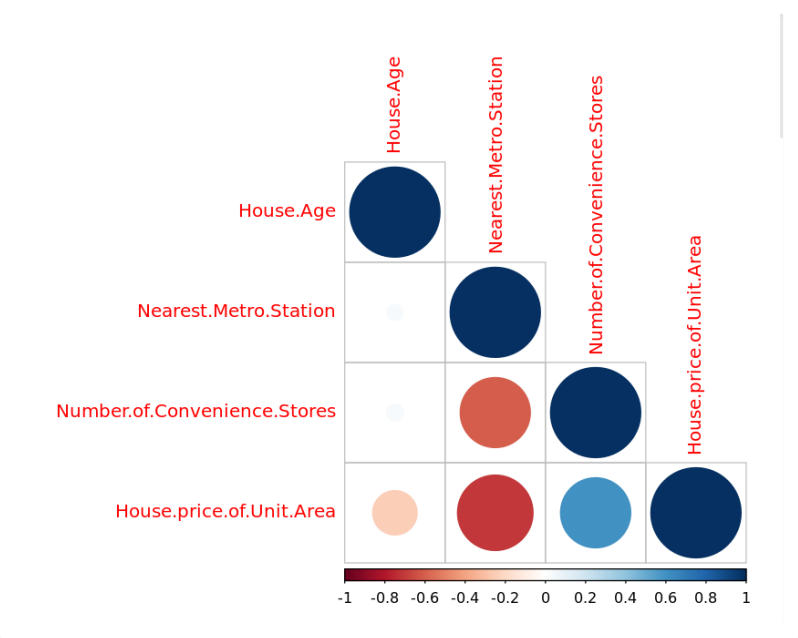
A matrix: 4 x 4 of type dbl

	House.Age	Nearest.Metro.Station	Number.of.Convenience.Stores
House.Age	1.00000000	0.03016725	0.03538514
Nearest.Metro.Station	0.03016725	1.00000000	-0.60471041
Number.of.Convenience.Stores	0.03538514	-0.60471041	1.00000000
House.price.of.Unit.Area	-0.24285150	-0.70134918	0.60585298

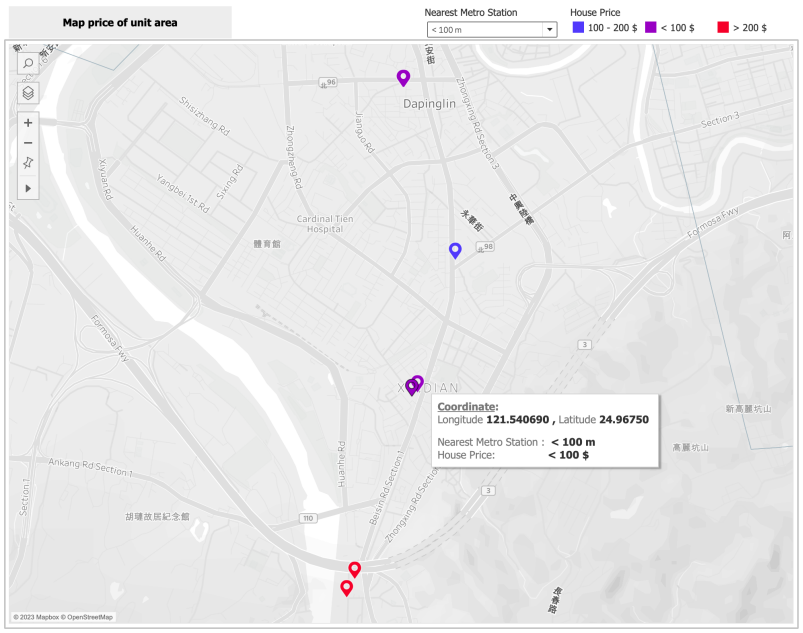
```
install.packages('corrplot')
library(corrplot)
```

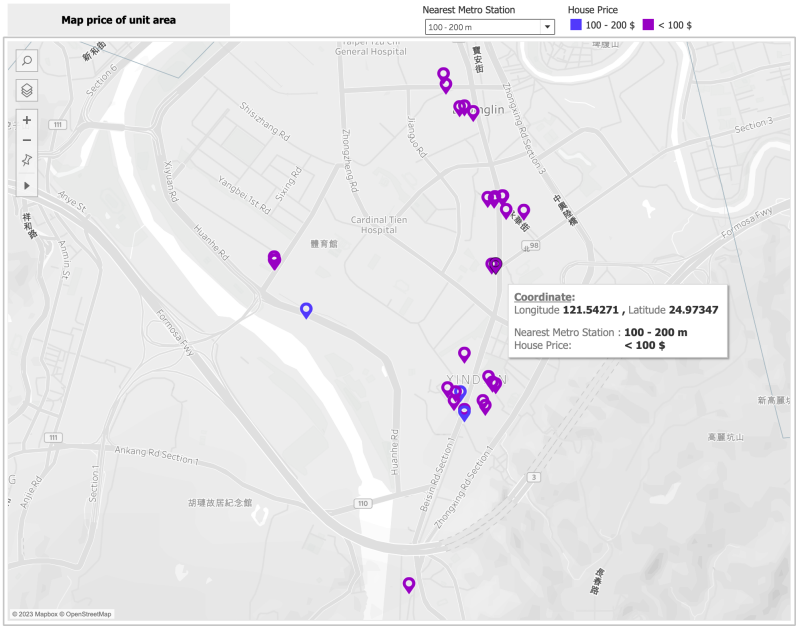
```
Installing package into ‘/work/.R/library’
(as ‘lib’ is unspecified)

corrplot 0.92 loaded
```

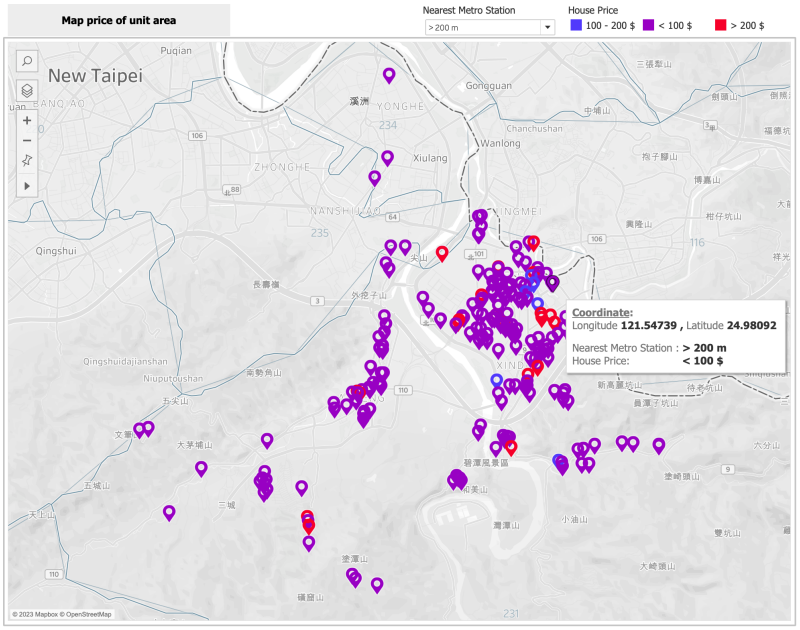
```
correlations <- cor(data_without_outliers[,c("House.Age", "Nearest.Metro.Station", "Number.of.Conven
corrplot(correlations, method="circle", type="lower",  sig.level = 0.01, insig = "blank")
```



Map of the Houses per Price of Unit area







Multiple Linear Regression

The dependent variable is House.price.of.Unit.Area.

The independent variables are House.Area, House.Age, Nearest.Metro.Station, Number.of.Convenience.Stores, latitude and longitude.

Create and Fit the Multiple Linear Regression Model

```r
# Building the regression model
model <- lm(House.price.of.Unit.Area ~ House.Area + House.Age + Nearest.Metro.Station +
            Number.of.Convenience.Stores + latitude + longitude, data = data_without_outliers)
```

Analyze the Model

```r
summary(model)
```

```
Call:
lm(formula = House.price.of.Unit.Area ~ House.Area + House.Age +
    Nearest.Metro.Station + Number.of.Convenience.Stores + latitude +
    longitude, data = data_without_outliers)

Residuals:
     Min       1Q   Median       3Q      Max
-23.5540  -3.6900  -0.1851   4.1549  25.9913

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                 -4.075e+03  4.647e+03  -0.877    0.381
House.AreaMedium            -9.737e+00  9.758e-01  -9.978  < 2e-16 ***
House.AreaSmall             -1.663e+01  1.460e+00 -11.393  < 2e-16 ***
House.Age                   -3.455e-03  3.844e-02  -0.090    0.928
Nearest.Metro.Station       -2.597e-03  5.607e-04  -4.632 4.90e-06 ***
Number.of.Convenience.Stores  8.836e-01  1.452e-01   6.084 2.72e-09 ***
latitude                     1.686e+02  3.458e+01   4.875 1.57e-06 ***
longitude                   -7.435e-01  3.664e+01  -0.020    0.984
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.656 on 403 degrees of freedom
Multiple R-squared:  0.7329,    Adjusted R-squared:  0.7283
F-statistic:   158 on 7 and 403 DF,  p-value: < 2.2e-16
```

Model Evaluation

```r
summary(model)$r.sq
summary(model)$adj.r.sq
summary(model)$fstatistic
```

```
0.732893837419183
```

```
0.728254276282544
```

```
value:      157.966198921571 numdf:      7 dendf:      403
```

```r
r2 <- summary(model)$r.squared
adjusted_r2 <- summary(model)$adj.r.squared
f_statistic <- summary(model)$fstatistic

print(paste0("R-squared:", r2))
print(paste0("Adjusted R-squared:", adjusted_r2))
print(paste0("F-statistic:", f_statistic))
```

```
[1] "R-squared:0.732893837419183"
[1] "Adjusted R-squared:0.728254276282544"
[1] "F-statistic:157.966198921571" "F-statistic:7"
[3] "F-statistic:403"
```

Adjusted R-squared is 0.7283, which is slightly lower than the R-squared value of 0.7329. This suggests that the inclusion of additional variables might not add much explanatory power to the model, as the adjusted R-squared value takes into account the number of predictors in the model.

F-statistic is 158, which is relatively high, indicating that the model as a whole is statistically significant in explaining the variance in the dependent variable.

For F-statistic: 7 and 406 These values represent the degrees of freedom associated with the F-statistic. The first number (7) is the degrees of freedom for the numerator (number of predictors), and the second number (406) is for the denominator (number of observations minus number of predictors minus 1).

A p-value < 0.05 suggests strong evidence against the null hypothesis, indicating that the model is significant. The p-value for the F-statistic is less than 2.2e-16, which is much less than 0.05. This means that there is strong evidence against the null hypothesis, which is that the model does not explain a significant amount of variance in the dependent variable.

Conclusion:

Approximately 73.29% of the variability in the dependent variable is explained by the independent variables included in the model.

Interpretation of regression coefficients

```
print(coef(model))
```

```
                (Intercept)             House.AreaMedium
              -4.074518e+03               -9.736541e+00
            House.AreaSmall                    House.Age
              -1.662993e+01               -3.454489e-03
Nearest.Metro.Station Number.of.Convenience.Stores
              -2.597082e-03                8.835874e-01
                  latitude                   longitude
               1.685956e+02               -7.435021e-01
```

```
summary(model)$coef
```

A matrix: 8 x 4 of type dbl

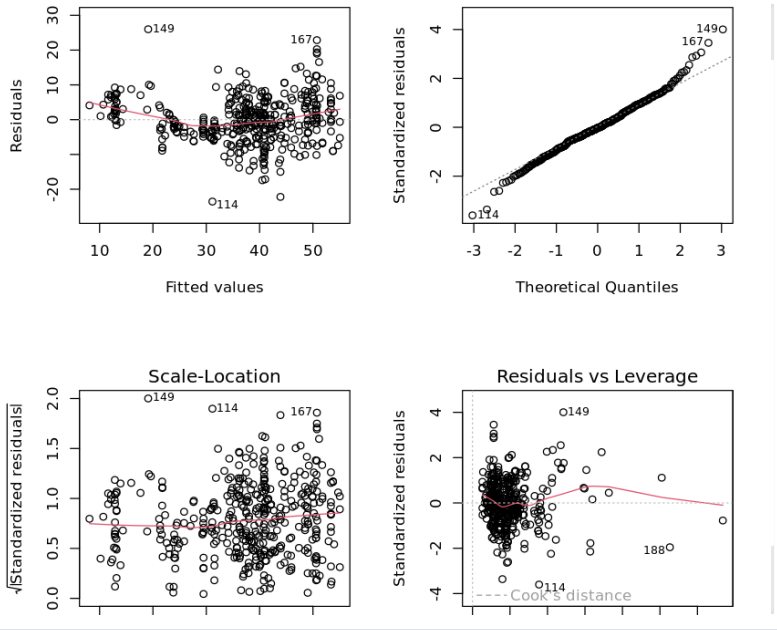
	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-4.074518e+03	4.646950e+03	-0.87681557	3.811092e-01
House.AreaMedium	-9.736541e+00	9.758003e-01	-9.97800567	4.225013e-21
House.AreaSmall	-1.662993e+01	1.459697e+00	-11.39272890	2.948628e-26
House.Age	-3.454489e-03	3.843911e-02	-0.08986911	9.284359e-01
Nearest.Metro.Station	-2.597082e-03	5.607177e-04	-4.63170933	4.901433e-06
Number.of.Convenience.Stores	8.835874e-01	1.452204e-01	6.08445769	2.720841e-09
latitude	1.685956e+02	3.458374e+01	4.87499699	1.568002e-06
longitude	-7.435021e-01	3.664251e+01	-0.02029070	9.838215e-01

Residual Analysis

```
print(resid(model))
```

```
        382         383         384         385         386         387
-7.41839676  9.68691236  1.54221016 -0.27272060  0.63812822 10.64048092
        388         389         390         391         392         393
-5.14834715 -3.82411194 16.55398817 -3.72262839 -6.35304977 -5.47817390
        394         395         396         397         398         399
 0.96300824  8.77398472  4.48153127 -2.00849777 -7.11618272 -1.66138830
        400         401         402         403         404         405
 1.05615160 -3.51556003 -13.07392314 -7.72746377 -5.21924586  0.35157076
        406         407         408         409         410         411
-0.64241900 -10.15533439 -13.84004666 -2.94010482  2.49875218 -3.37520245
        412         413         414
-1.27395707  3.98289615 10.52790659
```

```
par(mfrow = c(2, 2))  # Arrange plots in a 2x2 grid
plot(model)
```



The residuals versus fitted values plot appears to be a random scatterplot with no discernible pattern. This is a good sign, as it suggests that the model is adequately fitting the data.

The normal Q-Q plot appears to be a straight line, with some minor deviations at the tails. This suggests that the residuals are approximately normally distributed.

The residuals versus fitted values plot and the normal Q-Q plot suggest that the model is adequately fitting the data.

The scale-location plot shows a slight fan shape, suggesting that the variance of the residuals may be increasing with the fitted values. However, the deviation from a horizontal line is not very large. This suggests that the non-normality of the residuals is not severe.

The residuals versus leverage plot shows a few observations with high leverage. However, the residuals for these observations are not particularly large. This suggests that these observations are influential, but they are not having a large impact on the model coefficients.

The residuals versus leverage plot and the scale-location plot suggest that there are a few influential observations, but the non-normality of the residuals is not severe.

Model validation

```
# Split the data into training and testing sets
train_data <- sample(1:nrow(data_without_outliers), nrow(data_without_outliers) * 0.8)
test_data <- setdiff(1:nrow(data_without_outliers), train_data)

# Fit the training and testing data to the MLR model
train_model <- lm(House.price.of.Unit.Area ~ House.Area + House.Age + Nearest.Metro.Station + Number
predicted_values <- predict(train_model, data = data_without_outliers[test_data, ])
```

Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), R-squared (R^2).

Calculate squared errors

```
# Calculate squared errors and MSE
actual_values <- data_without_outliers$House.price.of.Unit.Area[test_data]
squared_errors <- (actual_values - predicted_values)^2
```

```
Warning message in actual_values - predicted_values:
“longer object length is not a multiple of shorter object length”
```

Calculate the Mean Squared Error (MSE)

```
# Calculate MSE
mse <- mean(squared_errors)
```

Calculate the Root Mean Squared Error (RMSE)

```
# Calculate RMSE
rmse <- sqrt(mean(squared_errors))
```

Calculate the R-squared (Coefficient of Determination)

```
# Extract the residuals and calculate residual sum of squares (RSS)
residuals <- model$residuals
rss <- sum(residuals^2)

# Calculate mean of actual values and the total sum of squares (TSS)
y_mean <- mean(data_without_outliers$House.price.of.Unit.Area)
tss <- sum((data_without_outliers$House.price.of.Unit.Area - y_mean)^2)

# Calculate the R-squared value:
r_squared <- 1 - (rss / tss)
```

Calculate Metrics Output

```
# Communication and interpretation
print(paste0("R-squared:", r_squared))
print(paste0("RMSE:", rmse))
print(paste0("MSE:", mse))
```

```
[1] "R-squared:0.732893837419183"
[1] "RMSE:16.8425339193873"
[1] "MSE:283.670948825712"
```

73.29% of the variance in the dependent variable is explained by the independent variables.

72.83% of the variance in the dependent variable is explained by the independent variables, taking into account the number of independent variables in the model.

F-statistic is 157.97 with associated degrees of freedom of 7 and 403 for the numerator and denominator.

The average error in the model's predictions is 16.92, indicating the average difference between predicted and actual house prices.

The average squared error in the model's predictions is 286.35, indicating the average squared difference between predicted and actual prices.

Statistical Tests

```
# F-test for overall significance
anova(model)
```

A anova: 7 x 5

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
	<int>	<dbl>	<dbl>	<dbl>	<dbl>
House.Area	2	3.875051e+04	1.937525e+04	4.372972e+02	1.073436e-16
House.Age	1	5.091553e+02	5.091553e+02	1.149158e+01	7.681034e-04
Nearest.Metro.Station	1	6.691499e+03	6.691499e+03	1.510263e+02	1.061764e-29
Number.of.Convenience.Stores	1	1.968981e+03	1.968981e+03	4.443968e+01	8.654305e-11
latitude	1	1.072711e+03	1.072711e+03	2.421096e+01	1.260508e-06
longitude	1	1.824168e-02	1.824168e-02	4.117125e-04	9.838215e-01
Residuals	403	1.785565e+04	4.430683e+01	NA	NA

House.Area has a high F value (437.30) and an extremely low p-value (close to 0), indicating that House.Area significantly explains the variance in house prices.

House.Age, Nearest.Metro.Station, Number.of.Convenience.Stores, latitude variables also have relatively high F values and very low p-values, suggesting they are statistically significant in explaining house price variance.

This means that there is a very strong statistical evidence of a difference between the means of the groups.

Longitude variable has a very low F value (0.00041) and a high p-value (0.9838), indicating that longitude does not significantly contribute to explaining the variance in house prices.

```
# Individual t-tests for each coefficient
t.test(model$coefficients[-1])
```

```
        One Sample t-test

data:  model$coefficients[-1]
t = 0.81895, df = 6, p-value = 0.4441
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -40.42818  81.10338
sample estimates:
mean of x
  20.3376
```

The t-statistic for the coefficient is 0.81895, with a p-value of 0.4441. This means that there is a 44.41% chance of obtaining a t-statistic as extreme or more extreme than 0.81895, even if the null hypothesis is true. Therefore, we fail to reject the null hypothesis that the coefficient is equal to zero.

The 95% confidence interval for the coefficient is -40.42818 to 81.10338. This means that we are 95% confident that the true value of the coefficient lies within this interval.

Overall, the t-test suggest that there is not enough evidence to reject the null hypothesis that the true mean of the coefficients is equal to zero. However, the confidence interval is quite wide, suggesting that more data is needed to get a more precise estimate of the coefficient.

Create a new variable called `HAgeC`

Create the new variable '`HAgeC`' based on conditions

```
# Create the new variable 'HAgeC' based on conditions
data_without_outliers$HAgeC <- ifelse(data_without_outliers$House.Age < 15.0, "New",
                   ifelse(data_without_outliers$House.Age >= 15.0 & data_without_outliers$House.Age

# Convert 'HAgeC' to a factor with specified levels
data_without_outliers$HAgeC <- factor(data_without_outliers$HAgeC, levels = c("Old", "Recent", "New")

# Check the unique values in 'HAgeC' to ensure it has multiple levels
unique(data_without_outliers$HAgeC)
```

```
Old · Recent · New
▶ Levels:
```

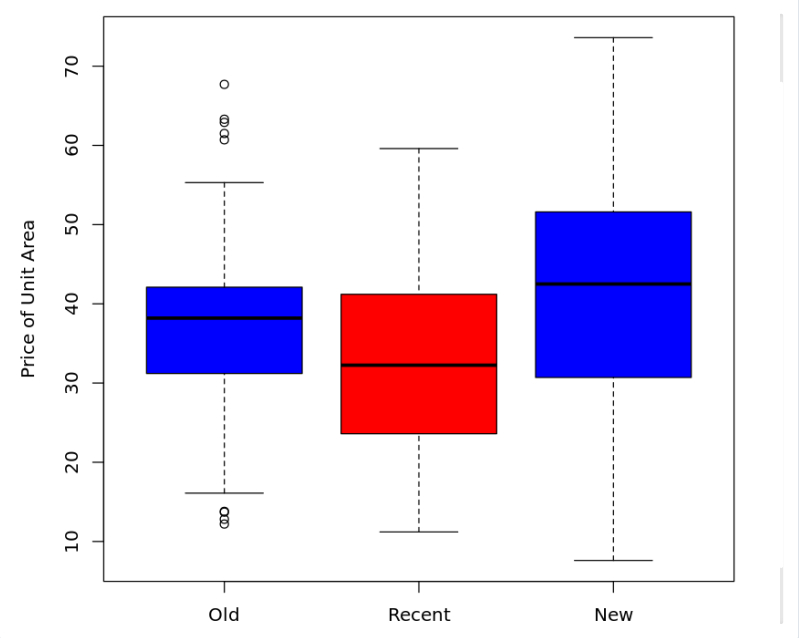
`HAgeC` variable summary

```
# Summary of the new variable
summary(data_without_outliers$HAgeC)
```

```
Old:      92 Recent:      130 New:      189
```

Visualization of "Price of Unit area" by "`HAgeC`" Levels

```
# Boxplot of Price of Unit area by HAgeC levels
boxplot(data_without_outliers$House.price.of.Unit.Area ~ data_without_outliers$HAgeC,
        xlab = "House Age Category",
        ylab = "Price of Unit Area",
        main = "Distribution of Price of Unit Area by House Age Category",
        col = c("blue", "red"))
```



Impact of "HAgeC" on "Price of Unit Area"

Create a regression model for House.price.of.Unit.Area variable and HAgeC variable

```
# Fit a regression model considering only 'HAgeC' as a predictor
model_hagec_only <- lm(data_without_outliers$House.price.of.Unit.Area ~ HAgeC, data = data_without_ou
```

Get the summary of the regression model

```
summary(model_hagec_only)
```

```
Call:
lm(formula = data_without_outliers$House.price.of.Unit.Area ~
    HAgeC, data = data_without_outliers)

Residuals:
    Min      1Q  Median      3Q     Max
-33.766  -9.517   1.134   8.633  32.234

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   36.936      1.273  29.013  < 2e-16 ***
HAgeCRecent   -4.369      1.664  -2.626  0.00896 **
HAgeCNew       4.430      1.552   2.854  0.00454 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 12.21 on 408 degrees of freedom
Multiple R-squared:  0.08997,   Adjusted R-squared:  0.08551
F-statistic: 20.17 on 2 and 408 DF,  p-value: 4.439e-09
```

Conclution

Approximately 8.997% of the variability in the dependent variable is explained by the independent variables included in the model. (R-squared = 0.08997)

Adjusted R-squared is slightly lower than the R-squared value, but this is not a cause for concern, as the sample size is large (n = 410). (Adjusted R-squared = 0.08551)

The residual standard error of 12.21 indicates that the average difference between the predicted and actual values of House.price.of.Unit.Area is 12.21.

The F-statistic of 20.17 is statistically significant (p-value < 0.001), which means that the model is better than simply using the mean of House.price.of.Unit.Area to predict the value of House.price.of.Unit.Area.

The model is a good fit for the data, as it explains a significant portion of the variability in the dependent variable and is statistically significant. However, it is important to note that the effect sizes of HAgeCRecent and HAgeCNew are relatively small.