NYC RealEstate

Fredrick Jones, Jian Quan Chen, Tilon Bobb

2024-04-24

Contents

Loading Required libraries

```
#Clear all
rm(list = ls())

options(scipen = 999)
```

Abstract

This study uses real estate transaction data to investigate the factors influencing property values in New York City. The dataset includes a variety of parameters that were gathered from public records and real estate listings, including property location, kind, size, and sale price. To understand the distribution and interrelationships of the dataset, exploratory data analysis (EDA) is the first systematic stage in the study technique. Data preparation is a step that comes next in order to encode categorical variables, handle missing values, and create new features. Stepwise regression, generalized linear models (GLM), robust regression, and conventional linear regression are all included in the design of regression models. The goal of developing predictive models that offer robustness against outliers while generalizing effectively to new data is achieved through the use of goodness-of-fit measures and diagnostic tests for residual analysis as the basis for model selection.

Keywords

NYC Real Estate Data Analysis Regression Modeling Housing Market Trends Neighborhood Analysis

Introduction

The New York real estate market is one of the most dynamic and influential sectors of urban development. With every real estate transaction representing a tangible real estate exchange, the NYC real estate market is a barometer of the city's socioeconomic landscape. In this report, we take an in-depth look at the NYC market through insights from a diverse dataset of New York real estate sales records.

Background and Motivation

The database under study contains a wealth of data spanning two decades, providing a detailed chronicle of real estate transactions in the five boroughs of New York - Manhattan, the Bronx, Brooklyn, Queens and Staten Island. Our analysis is based on the foundation created by this dataset, which provides insight into the multifaceted dynamics of NYC real estate sales.

The motivation is to identify the factors that influence NYC real estate prices, so our research is based on a variety of analytical techniques. and methods. From data analysis to regression modeling, we seek to uncover the complex interplay of variables that affect real estate. By examining real estate sales trends, identifying key predictors of sales prices, and assessing the impact of factors such as property size, location, and tax bracket, we aim to shed light on the mechanics behind the NYC real estate world.

As we delve into the depths of the NYC real estate market, our analysis aims to provide stakeholders with actionable insights on investors and from decision makers to real estate developers and potential buyers. By uncovering the drivers of real estate prices and delineating market trends, we aim to provide decision makers with the information they need to navigate the complexities of the real estate ecosystem.

Literature Review

Michael Gaynor's Project Michael Gaynor conducted a project to explore the NYC real estate market using SQL queries and Tableau visualization techniques. His investigation aimed to answer four main questions:

1. Which of the five boroughs is the most expensive?
2. Which of the five boroughs have the most sales?
3. What type of properties sell the most in each of the 5 boroughs?
4. What property type influences sales?

Gaynor's approach involved understanding the task, prepping the data, analyzing the data using SQL queries and Tableau visualization, and presenting the findings through an interactive dashboard. Through his analysis, Gaynor discovered insights such as the most expensive borough, the borough with the most property sales, and the types of properties that sell the most in each borough.

Comparison and Evaluation

Gaynor's research utilized the same NYC real estate dataset to address similar questions as our investigation. However, there are significant differences between Gaynor's approach and our own project:

- **Methodology**: Gaynor primarily used SQL queries and Tableau visualization tools for data analysis, while our investigation utilized R programming language. Our approach involved a combination of data preprocessing, exploratory data analysis (EDA), statistical modeling, and visualization techniques implemented in R.
- **Presentation of Findings**: Gaynor's project focused on presenting the results through an interactive dashboard created in Tableau. In contrast, our investigation may present the findings through various formats such as tables, charts, and narratives within the R Markdown document.
- **Data Preprocessing**: While Gaynor mentioned data cleaning and preprocessing, the details of these steps were not extensively discussed. In our investigation, we employed specific techniques such as handling missing values, outlier detection and removal, and data transformation using R packages like dplyr and tidyr.
- **Statistical Modeling**: Our investigation may involve the application of statistical models such as linear regression, generalized linear models, or machine learning algorithms to explore relationships between variables and predict real estate prices. Gaynor's project did not explicitly mention the use of statistical modeling.

Advantages and Drawbacks

The advantages of Gaynor's approach include:

- Comprehensive analysis of the NYC real estate market using SQL and Tableau.
- Clear presentation of findings through interactive visualization.

However, there may be some drawbacks to Gaynor's approach, such as:

- Reliance on SQL and Tableau tools may limit accessibility for researchers unfamiliar with these technologies.
- Lack of detailed explanation of data cleaning and preprocessing steps.

Regression Modelling Methodology

Data Preparation

The first part of our analysis consisted of importing the dataset that included data on property sales in New York City. After we uploaded the dataset, we conducted an exploratory data analysis to understand its organization, factors, and any problems that required attention.

During the exploratory data analysis, we faced one of the first hurdles with the discovery of missing values in multiple columns. To tackle this problem, we methodically pinpointed the variables with incomplete data and assessed the percentage of missing values in each instance. After careful deliberation, we made the choice to eliminate data points containing incomplete information, as they comprised less than 5% of the entire data set. This method enabled us to keep a large part of the data while reducing the effect of missing values on future analyses.

After addressing missing data, we focused on the distribution of numerical variables in the dataset. We noticed that many variables showed noticeable skewness, suggesting possible departures from normal distribution. To tackle this problem, we utilized Tukey's method for identifying and eliminating outliers. Our goal was to enhance the robustness of future analyses by improving the distributional properties of numeric variables through the identification and exclusion of outliers from the dataset.

Additionally, we examined the connections between variables using correlation analysis. This included computing correlation coefficients for pairs of numerical variables in order to evaluate the magnitude and orientation of their relationships. During this examination, we found multiple variables that showed strong positive relationships with each other, along with variables that had weaker or negative relationships. These results offered valuable information on potential factors that could predict real estate prices and guided the choice of variables for inclusion in regression analysis.

In addition, we analysed changes over time by graphing the time-based pattern of property values in New York City. This examination showed a rising pattern in mean sale prices over time, with variations in certain time frames. Through the visualization of time-based trends, we achieved a better comprehension of the fluctuations within the real estate market of New York City, pinpointing potential influences on property price fluctuations.

Regression Modelling

After completing thorough data preparation and exploratory analysis, we focused on the main objective of our research: using regression modelling to forecast real estate prices in New York City. The method we used involved carefully building linear regression models, using different predictor variables to understand the complex factors influencing real estate prices. Leading our regression modelling was the incorporation of important predictor variables that were considered to have a significant impact on real estate prices. These variables included a wide range of factors, each providing valuable perspectives on the intricate fabric of the New York City real estate market. Included in these predictors were the quantity of housing units in a property, offering an insight into its size and ability to house residents. The categorization of taxes for properties at various points highlighted their financial status and legal ramifications, revealing insights into the larger economic and legal environments in which these properties function. Furthermore, the year

in which the building was constructed was identified as a crucial factor in predicting real estate values, giving a historical perspective on how they change over time. We aimed to capture temporal trends and identify any seasonal or cyclical patterns that could affect pricing dynamics by including the sale date of properties in our models. Moreover, factors like total area and land area were essential in evaluating the physical size and spatial characteristics of properties, enhancing our comprehension of their inherent value. By conducting thorough regression analysis, we discovered significant statistical connections between these predictor variables and property prices, revealing the complex interaction of factors that influence pricing decisions in the real estate market of New York City. Nevertheless, even though our models were strong, the adjusted R-squared values suggested that there might be additional variability that was not accounted for, indicating the presence of hidden factors that were not included in our analysis.

Explorartory Data Analysis

Loading Data

```
nyc_data <- read.csv("C:/Users/Jian/Desktop/DATA 621 -Business Analytics and Data Mining/Final Project/r

head(nyc_data)
```

```
##   BOROUGH  NEIGHBORHOOD        BUILDING.CLASS.CATEGORY TAX.CLASS.AT.PRESENT
## 1       1 ALPHABET CITY        01 ONE FAMILY DWELLINGS                    1
## 2       1 ALPHABET CITY        02 TWO FAMILY DWELLINGS                    1
## 3       1 ALPHABET CITY 07 RENTALS - WALKUP APARTMENTS                   2B
## 4       1 ALPHABET CITY 07 RENTALS - WALKUP APARTMENTS                   2B
## 5       1 ALPHABET CITY 07 RENTALS - WALKUP APARTMENTS                    2
## 6       1 ALPHABET CITY 07 RENTALS - WALKUP APARTMENTS                   2A
##   BLOCK LOT EASE.MENT BUILDING.CLASS.AT.PRESENT                    ADDRESS
## 1   374  46                                  A4       347 EAST 4TH STREET
## 2   377   1                                  S2              110 AVENUE C
## 3   373  16                                  C1       326 EAST 4TH STREET
## 4   373  17                                  C1       328 EAST 4TH STREET
## 5   376  54                                  C4 719 EAST SIXTH STREET, 1B
## 6   377  52                                  C2       271 EAST 7TH STREET
##   APARTMENT.NUMBER ZIP.CODE RESIDENTIAL.UNITS COMMERCIAL.UNITS TOTAL.UNITS
## 1                    10009                 1                0           1
## 2                    10009                 2                1           3
## 3                    10009                10                0          10
## 4                    10009                10                0          10
## 5                    10009                20                0          20
## 6                    10009                 5                0           5
##   LAND.SQUARE.FEET GROSS.SQUARE.FEET YEAR.BUILT TAX.CLASS.AT.TIME.OF.SALE
## 1             2116              4400       1900                         1
## 2             1502              2790       1901                         1
## 3             2204              8625       1899                         2
## 4             2204              8625       1900                         2
## 5             2302              9750       1900                         2
## 6             2168              3728       1900                         2
##   BUILDING.CLASS.AT.TIME.OF.SALE SALE.PRICE           SALE.DATE
## 1                             A4     399000 2022-09-29 00:00:00
## 2                             S2    2999999 2022-09-15 00:00:00
## 3                             C1   16800000 2022-08-04 00:00:00
## 4                             C1   16800000 2022-08-04 00:00:00
## 5                             C4     158822 2022-09-27 00:00:00
## 6                             C2          0 2022-08-05 00:00:00
```

```r
glimpse(nyc_data)
```

Glipmse of the dataset

```
## Rows: 1,603,826
## Columns: 21
## $ BOROUGH                          <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ NEIGHBORHOOD                     <chr> "ALPHABET CITY", "ALPHABET CITY", "ALPH~
## $ BUILDING.CLASS.CATEGORY          <chr> "01 ONE FAMILY DWELLINGS", "02 TWO FAMI~
## $ TAX.CLASS.AT.PRESENT             <chr> "1", "1", "2B", "2B", "2", "2A", "2A", ~
## $ BLOCK                            <int> 374, 377, 373, 373, 376, 377, 377, 379,~
## $ LOT                              <int> 46, 1, 16, 17, 54, 52, 52, 25, 45, 47, ~
## $ EASE.MENT                        <chr> "", "", "", "", "", "", "", "", "", "",~
## $ BUILDING.CLASS.AT.PRESENT        <chr> "A4", "S2", "C1", "C1", "C4", "C2", "C2~
## $ ADDRESS                          <chr> "347 EAST 4TH STREET", "110 AVENUE C", ~
## $ APARTMENT.NUMBER                 <chr> "", "", "", "", "", "", "", "", "", "",~
## $ ZIP.CODE                         <dbl> 10009, 10009, 10009, 10009, 10009, 1000~
## $ RESIDENTIAL.UNITS                <dbl> 1, 2, 10, 10, 20, 5, 5, 7, 10, 10, 29, ~
## $ COMMERCIAL.UNITS                 <dbl> 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, ~
## $ TOTAL.UNITS                      <dbl> 1, 3, 10, 10, 20, 5, 5, 8, 10, 10, 29, ~
## $ LAND.SQUARE.FEET                 <dbl> 2116, 1502, 2204, 2204, 2302, 2168, 216~
## $ GROSS.SQUARE.FEET                <dbl> 4400, 2790, 8625, 8625, 9750, 3728, 372~
## $ YEAR.BUILT                       <dbl> 1900, 1901, 1899, 1900, 1900, 1900, 190~
## $ TAX.CLASS.AT.TIME.OF.SALE        <int> 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, ~
## $ BUILDING.CLASS.AT.TIME.OF.SALE <chr> "A4", "S2", "C1", "C1", "C4", "C2", "C2~
## $ SALE.PRICE                       <dbl> 399000, 2999999, 16800000, 16800000, 15~
## $ SALE.DATE                        <chr> "2022-09-29 00:00:00", "2022-09-15 00:0~
```

Assessing missing values

```r
# Check for missing values
missing_values <- colSums(is.na(nyc_data))

# View columns with missing values
missing_columns <- names(missing_values[missing_values > 0])
print(missing_columns)
```

```
## [1] "ZIP.CODE"                  "RESIDENTIAL.UNITS"
## [3] "COMMERCIAL.UNITS"          "TOTAL.UNITS"
## [5] "LAND.SQUARE.FEET"          "GROSS.SQUARE.FEET"
## [7] "YEAR.BUILT"                "TAX.CLASS.AT.TIME.OF.SALE"
## [9] "SALE.PRICE"
```

Drop missing values since there is less than 5% of dataset missing values hence safe to drop all missing values

```r
clean_nyc <- na.omit(nyc_data)
str(clean_nyc)
```

```
## 'data.frame':    1470283 obs. of  21 variables:
##  $ BOROUGH                       : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ NEIGHBORHOOD                  : chr  "ALPHABET CITY" "ALPHABET CITY" "ALPHABET CITY" "ALPHABET CI
##  $ BUILDING.CLASS.CATEGORY       : chr  "01 ONE FAMILY DWELLINGS" "02 TWO FAMILY DWELLINGS" "07 RENTA
##  $ TAX.CLASS.AT.PRESENT          : chr  "1" "1" "2B" "2B" ...
##  $ BLOCK                         : int  374 377 373 373 376 377 377 379 389 389 ...
##  $ LOT                           : int  46 1 16 17 54 52 52 25 45 47 ...
##  $ EASE.MENT                     : chr  "" "" "" "" ...
```

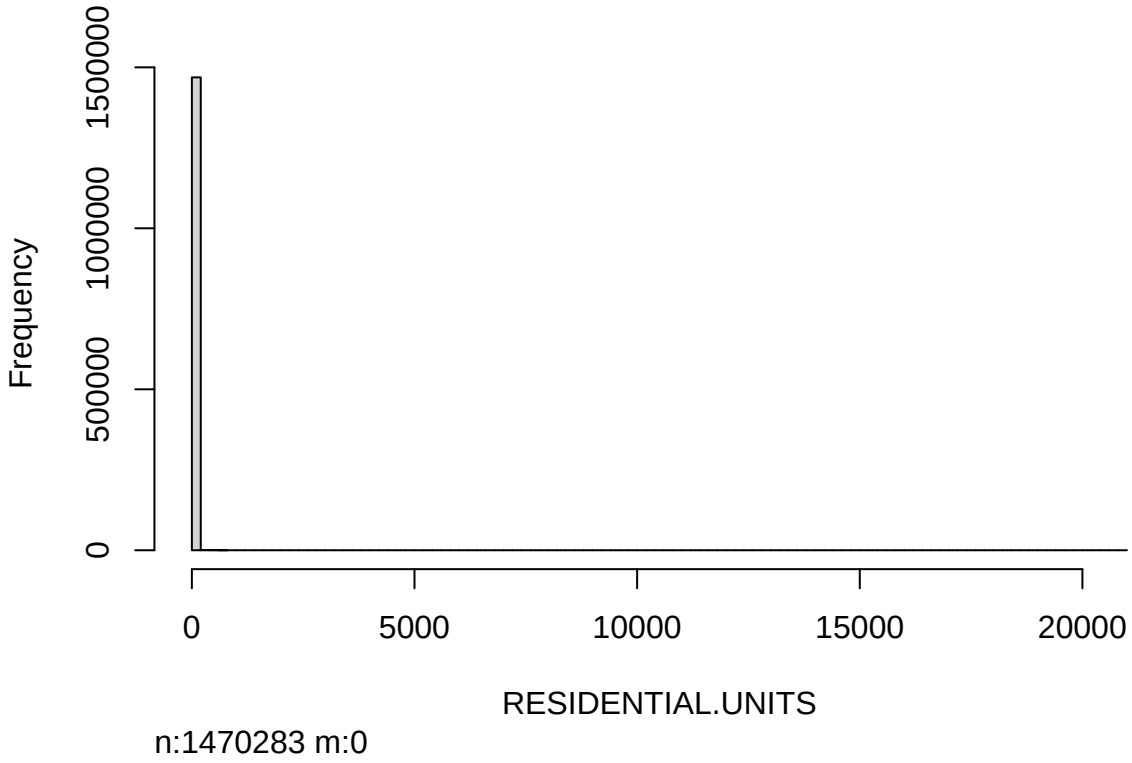
```
##  $ BUILDING.CLASS.AT.PRESENT     : chr  "A4" "S2" "C1" "C1" ...
##  $ ADDRESS                       : chr  "347 EAST 4TH STREET" "110 AVENUE C" "326 EAST 4TH STREET" "3
##  $ APARTMENT.NUMBER              : chr  "" "" "" "" ...
##  $ ZIP.CODE                      : num  10009 10009 10009 10009 10009 ...
##  $ RESIDENTIAL.UNITS             : num  1 2 10 10 20 5 5 7 10 10 ...
##  $ COMMERCIAL.UNITS              : num  0 1 0 0 0 0 0 1 0 0 ...
##  $ TOTAL.UNITS                   : num  1 3 10 10 20 5 5 8 10 10 ...
##  $ LAND.SQUARE.FEET              : num  2116 1502 2204 2204 2302 ...
##  $ GROSS.SQUARE.FEET             : num  4400 2790 8625 8625 9750 ...
##  $ YEAR.BUILT                    : num  1900 1901 1899 1900 1900 ...
##  $ TAX.CLASS.AT.TIME.OF.SALE     : int  1 1 2 2 2 2 2 2 2 2 ...
##  $ BUILDING.CLASS.AT.TIME.OF.SALE: chr  "A4" "S2" "C1" "C1" ...
##  $ SALE.PRICE                    : num  399000 2999999 16800000 16800000 158822 ...
##  $ SALE.DATE                     : chr  "2022-09-29 00:00:00" "2022-09-15 00:00:00" "2022-08-04 00:0
##  - attr(*, "na.action")= 'omit' Named int [1:133543] 29 30 31 32 33 34 35 36 37 38 ...
##   ..- attr(*, "names")= chr [1:133543] "29" "30" "31" "32" ...
```

All numeric variables are heavily skewed to the right,hence a clear indication of outliers

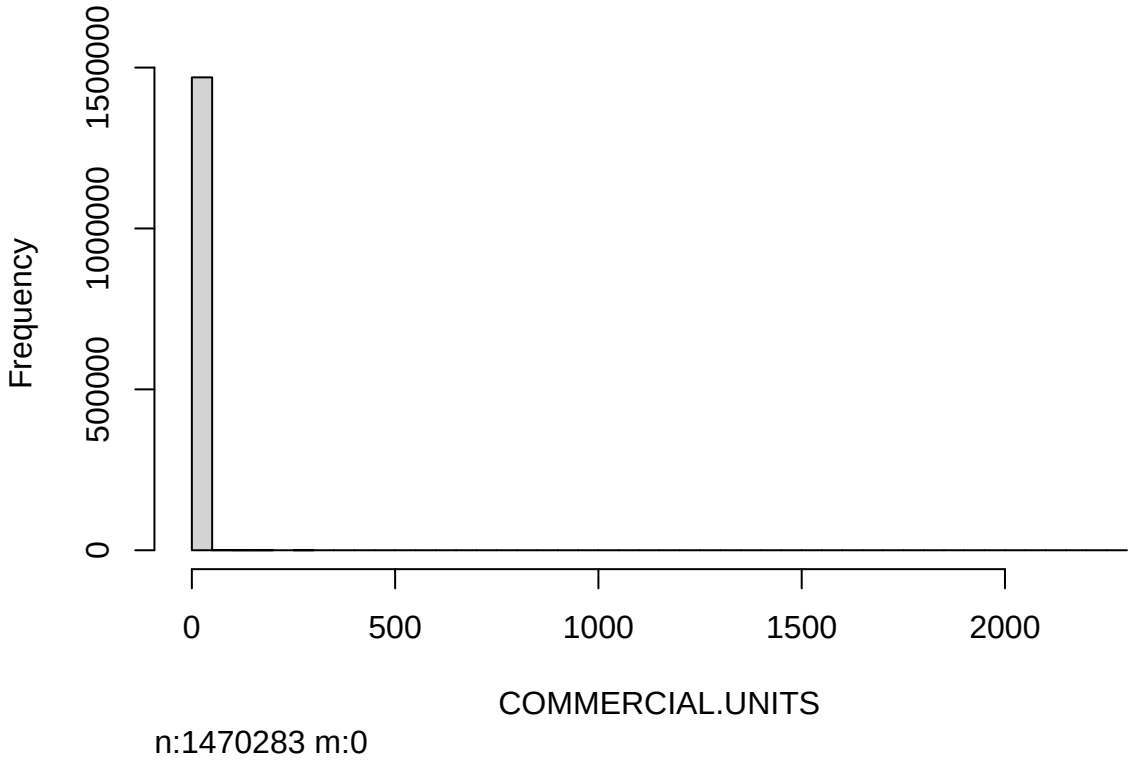
All distributions exhibit a highly skewed pattern, with a single bar extending vertically at the rightmost end of the x-axis, suggesting the presence of potential outliers with extremely high unit counts.

```
# Numeric variables
numeric_vars <- c("RESIDENTIAL.UNITS", "COMMERCIAL.UNITS", "TOTAL.UNITS",
                  "LAND.SQUARE.FEET", "GROSS.SQUARE.FEET", "SALE.PRICE")


num_data <-clean_nyc[, numeric_vars]
for (i in 1:length(names(num_data))){
  print(i)
  hist( num_data[i], main='hist', breaks=20, prob=TRUE)
}
```

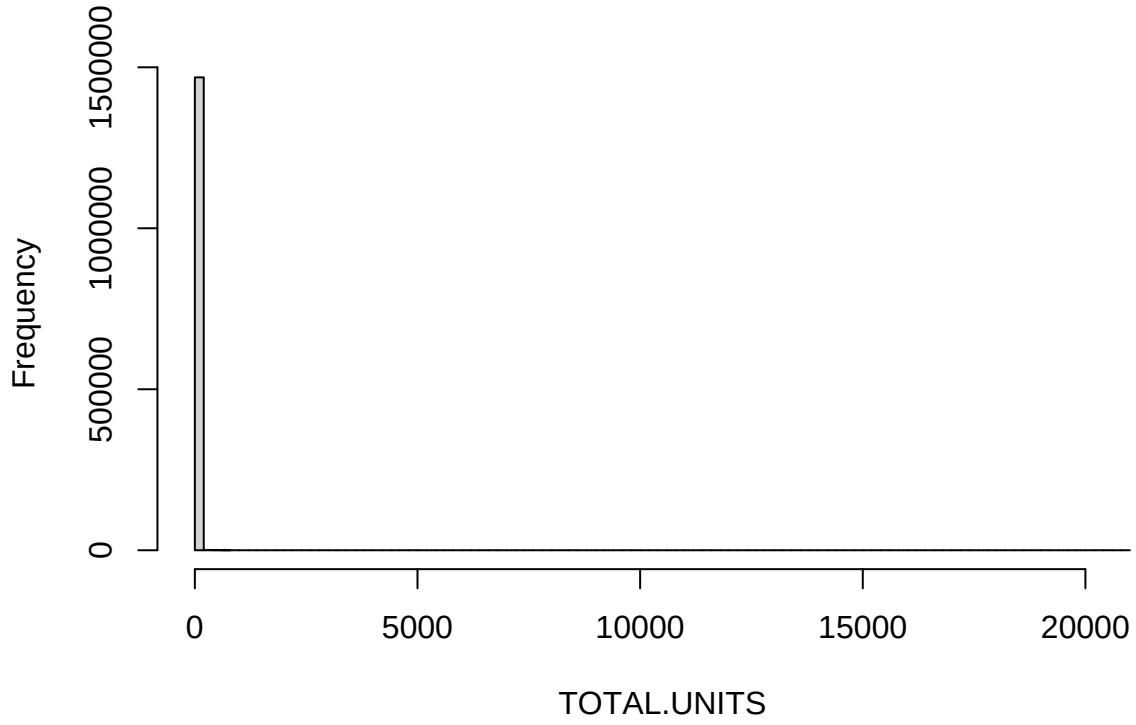
```
## [1] 1
```


Frequency
0
500000
1000000
1500000
0
5000
10000
15000
20000
RESIDENTIAL.UNITS
n:1470283 m:0

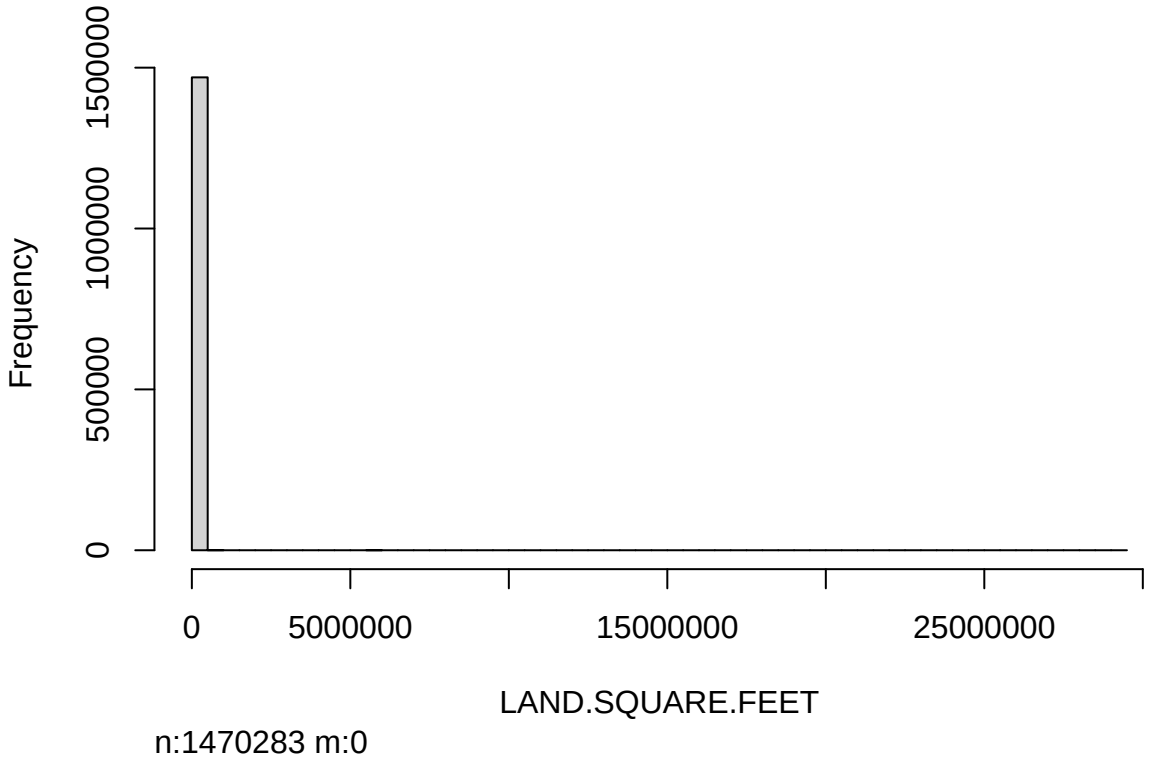
```
## [1] 2
```


Frequency
0
500000
1000000
1500000
0
500
1000
1500
2000
COMMERCIAL.UNITS
n:1470283 m:0

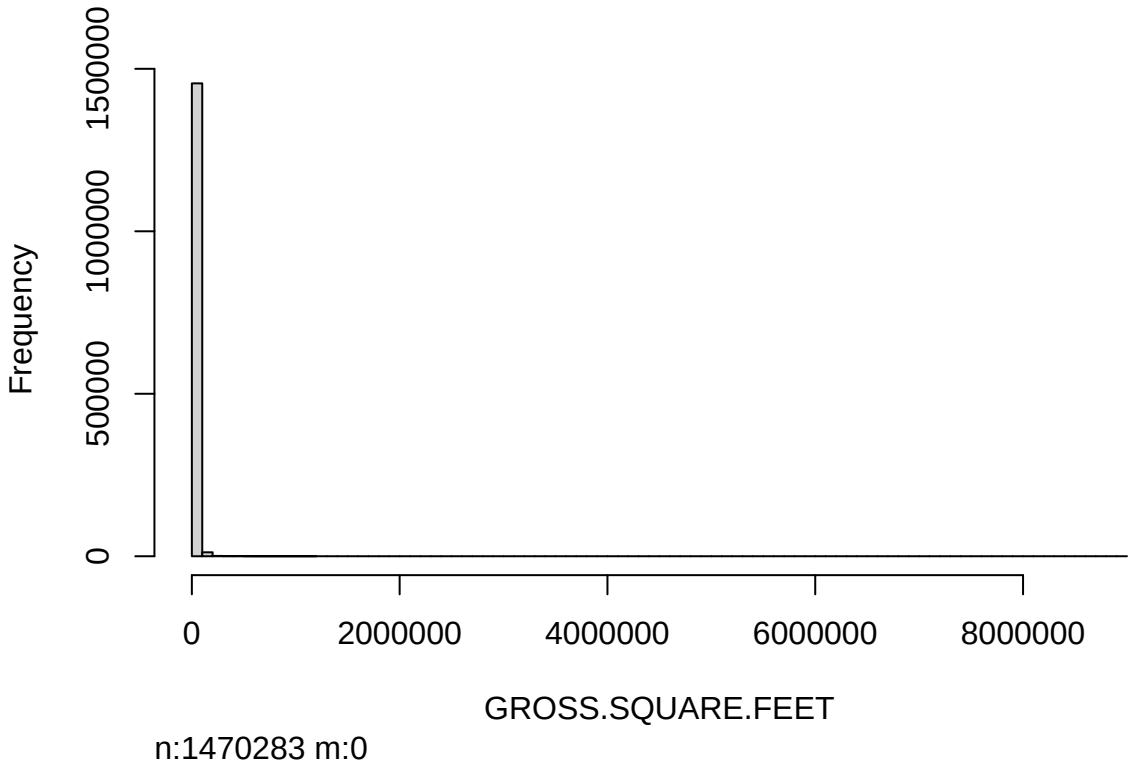
```
## [1] 3
```


Frequency
0
500000
1000000
1500000
0
5000
10000
15000
20000
TOTAL.UNITS
n:1470283 m:0

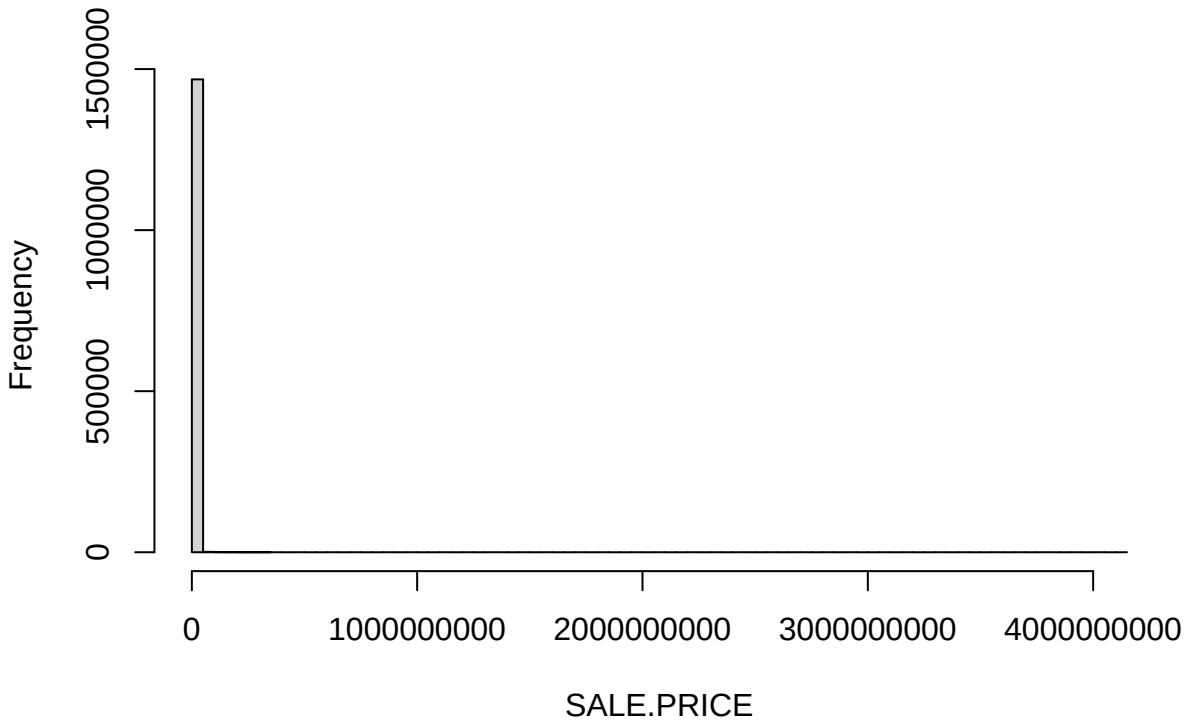
```
## [1] 4
```


Frequency
0
500000
1000000
1500000
0
5000000
15000000
25000000
LAND.SQUARE.FEET
n:1470283 m:0

```
## [1] 5
```


Frequency
0
500000
1000000
1500000
0
2000000
4000000
6000000
8000000
GROSS.SQUARE.FEET
n:1470283 m:0

```
## [1] 6
```




```r
# Function to remove outliers based on Tukey's method
remove_outliers <- function(data, variable) {
  q1 <- quantile(data[[variable]], 0.25)
  q3 <- quantile(data[[variable]], 0.75)
  iqr <- q3 - q1
  lower_bound <- q1 - 1.5 * iqr
  upper_bound <- q3 + 1.5 * iqr
  filtered_data <- data[data[[variable]] >= lower_bound & data[[variable]] <= upper_bound, ]
  return(filtered_data)
}

# Apply the function to each numeric variable in clean_nyc
for (var in numeric_vars) {
  clean_nyc <- remove_outliers(clean_nyc, var)
}
```

There was a clear improvement of distribution after removal of outliers

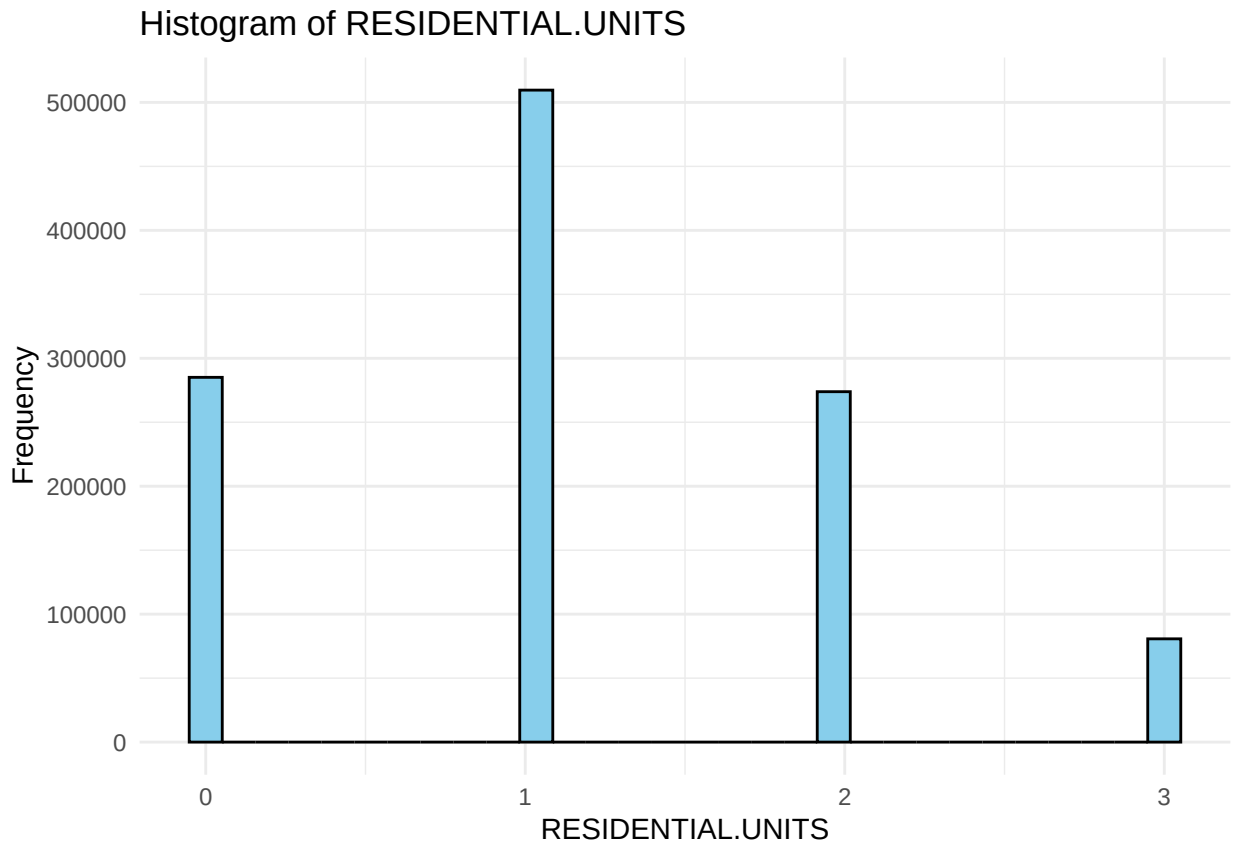
Distribution after applying an outlier removal technique, such as the Interquartile Range (IQR) method. The blue bars show a somewhat more balanced distribution, with the most extreme outliers removed, resulting in a narrower range of residential unit counts.

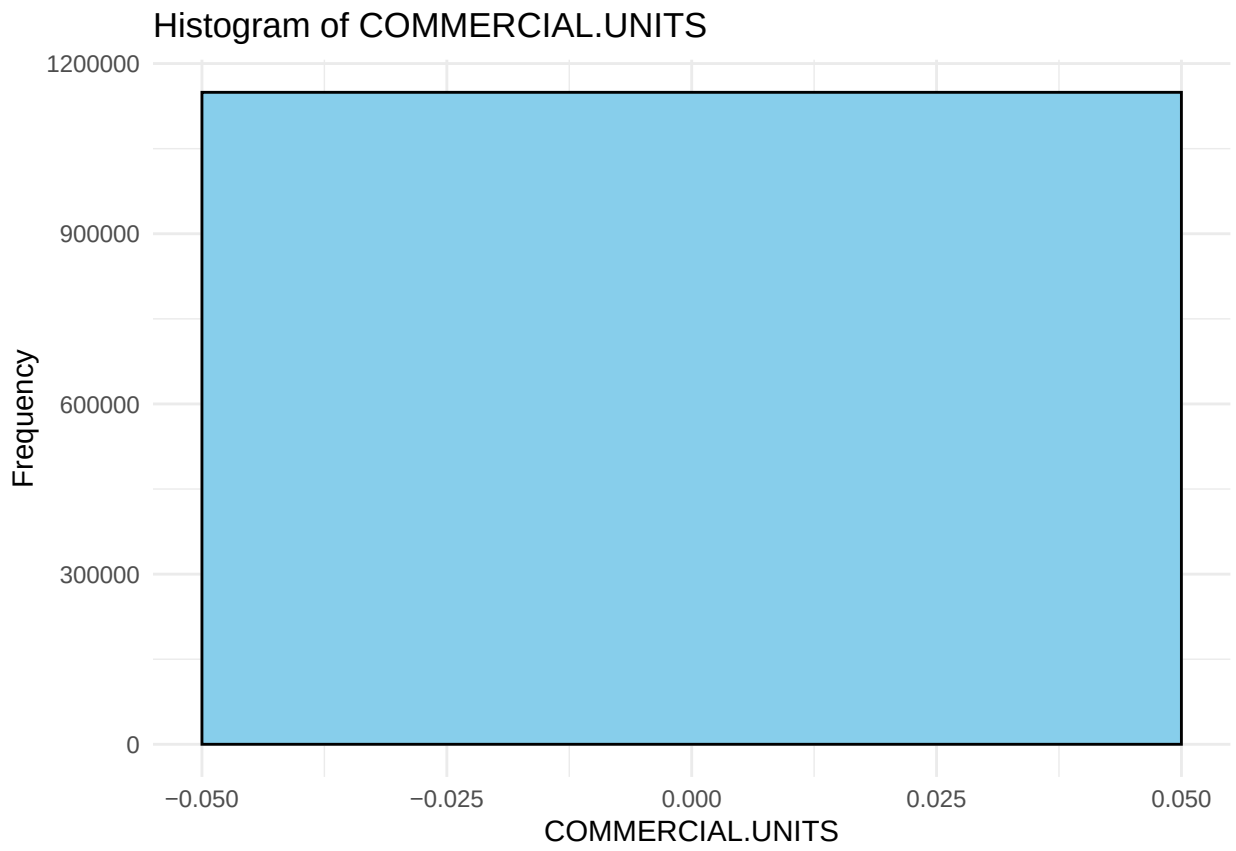
```r
num_data <-clean_nyc[, numeric_vars]
# Create histograms for each numeric variable
hist_plots <- lapply(numeric_vars, function(var) {
  ggplot(data = num_data, aes_string(x = var)) +
    geom_histogram(fill = "skyblue", color = "black", bins = 30) +
```

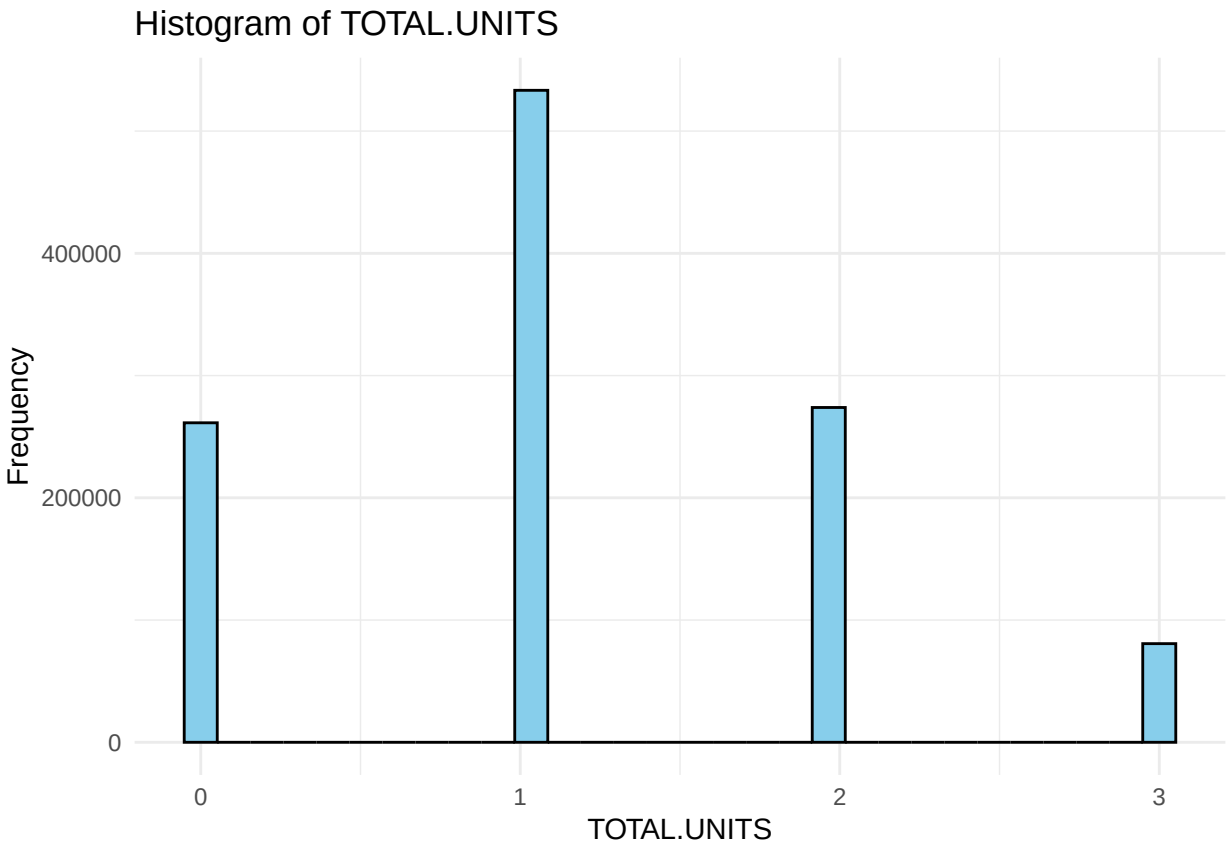
```
    labs(title = paste("Histogram of", var),
         x = var,
         y = "Frequency") +
    theme_minimal()
})
```

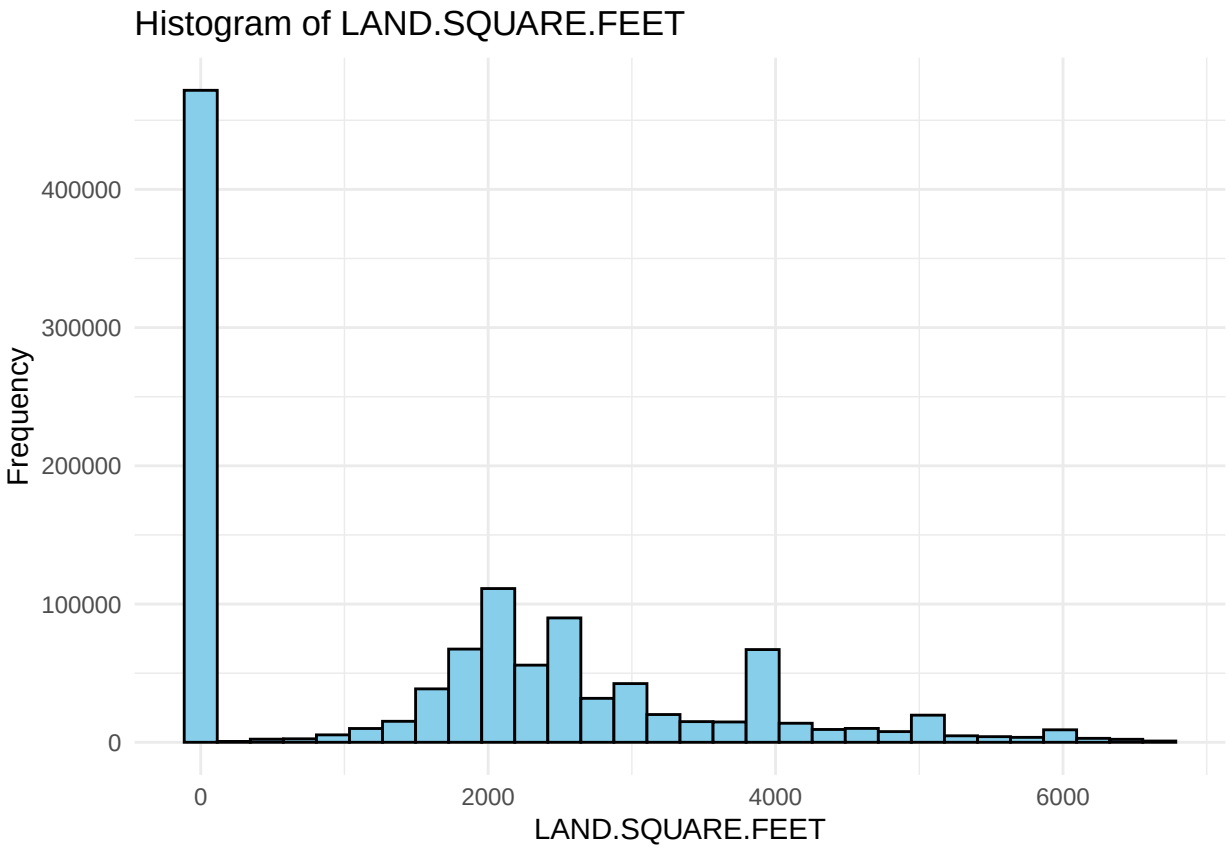
```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with `aes()`.
## i See also `vignette("ggplot2-in-packages")` for more information.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

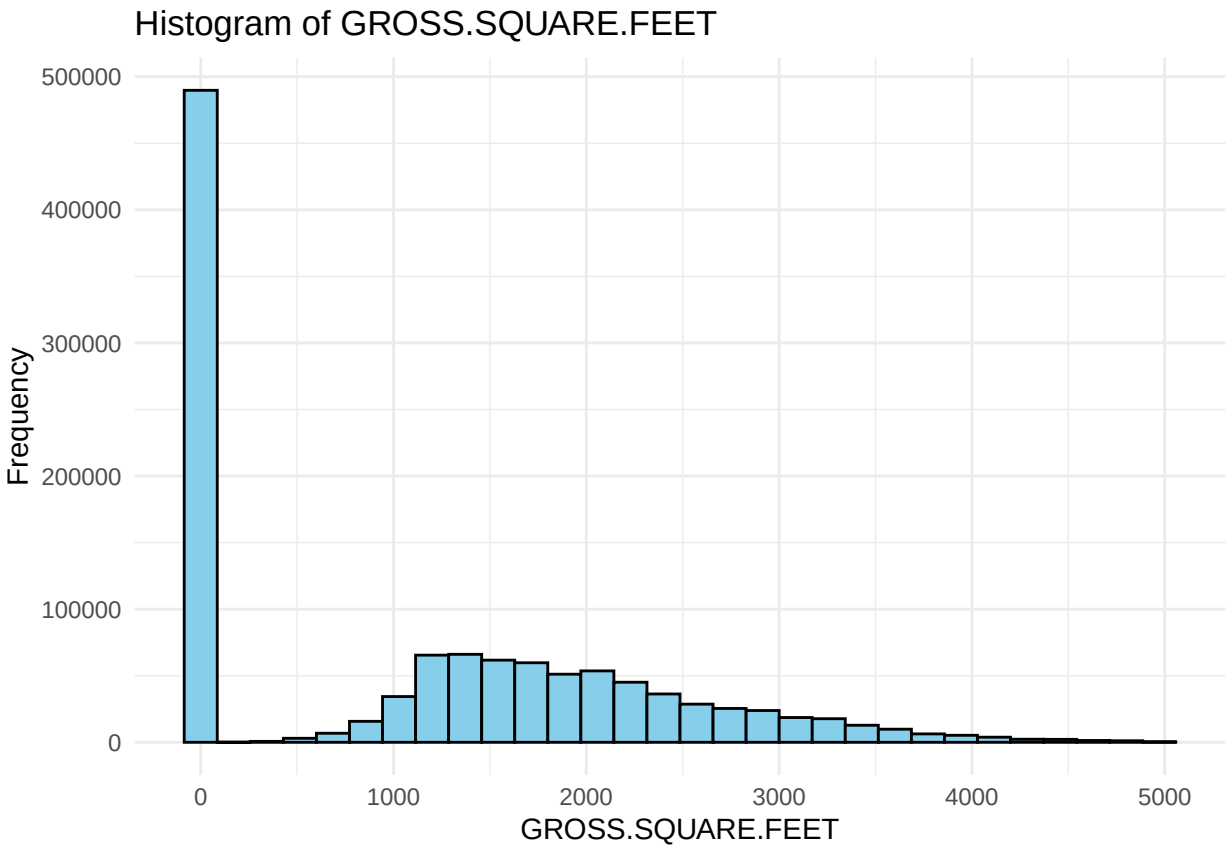
```
# Output the histograms
for (plot in hist_plots) {
  print(plot)
}
```

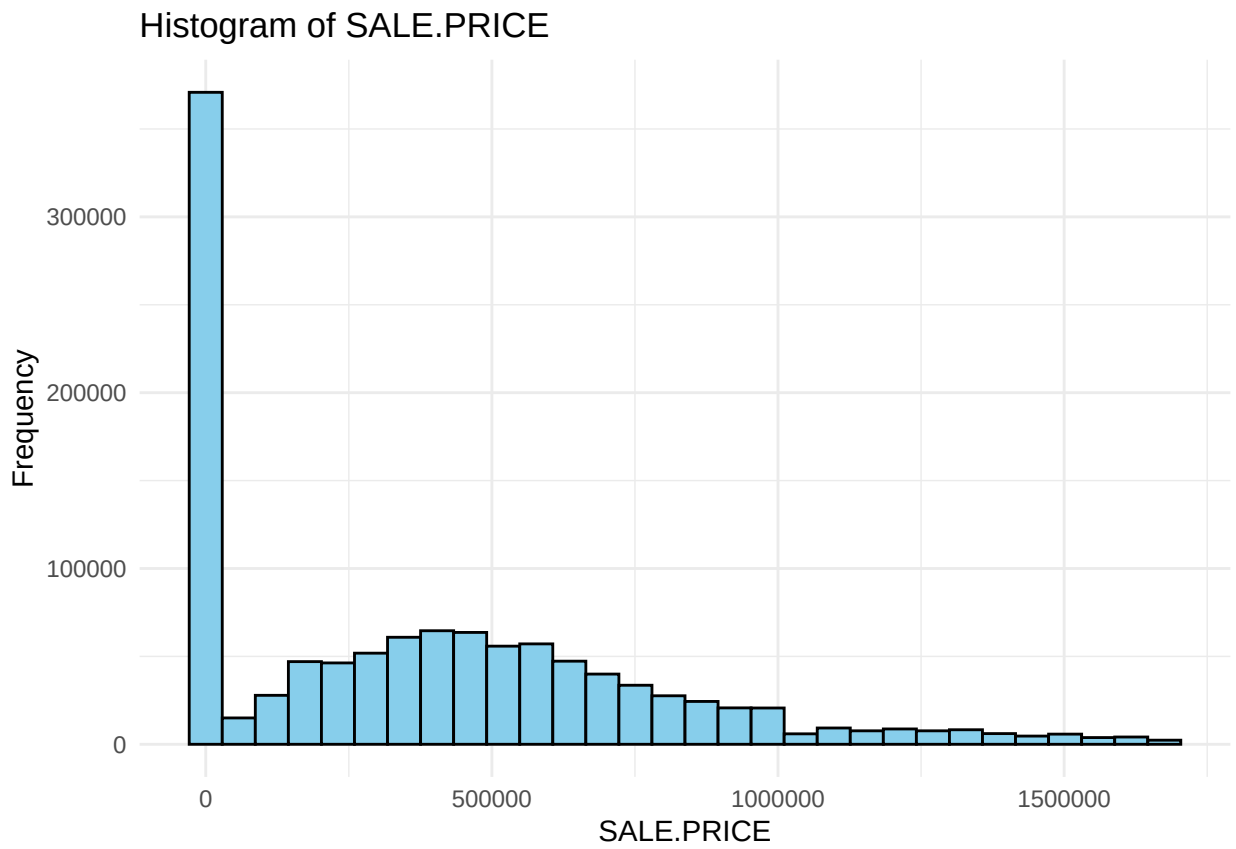



Histogram of COMMERCIAL.UNITS
Frequency
1200000
900000
600000
300000
0
−0.050
−0.025
0.000
0.025
0.050
COMMERCIAL.UNITS


Histogram of TOTAL.UNITS
Frequency
400000
200000
0
0
1
2
3
TOTAL.UNITS


Histogram of LAND.SQUARE.FEET
Frequency
400000
300000
200000
100000
0
0
2000
4000
6000
LAND.SQUARE.FEET


Histogram of GROSS.SQUARE.FEET
Frequency
500000
400000
300000
200000
100000
0
0
1000
2000
3000
4000
5000
GROSS.SQUARE.FEET

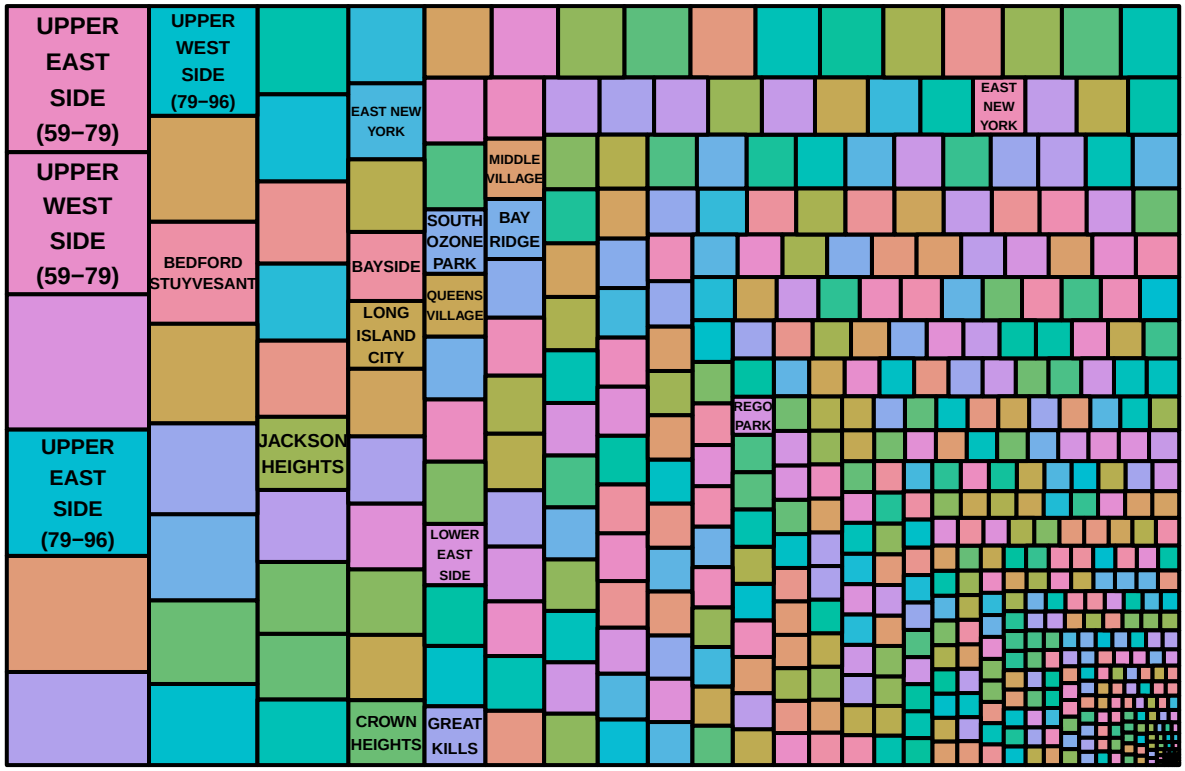



```
# Drop COMMERCIAL.UNITS variable
clean_nyc <- clean_nyc[, !names(clean_nyc) %in% "COMMERCIAL.UNITS"]
```

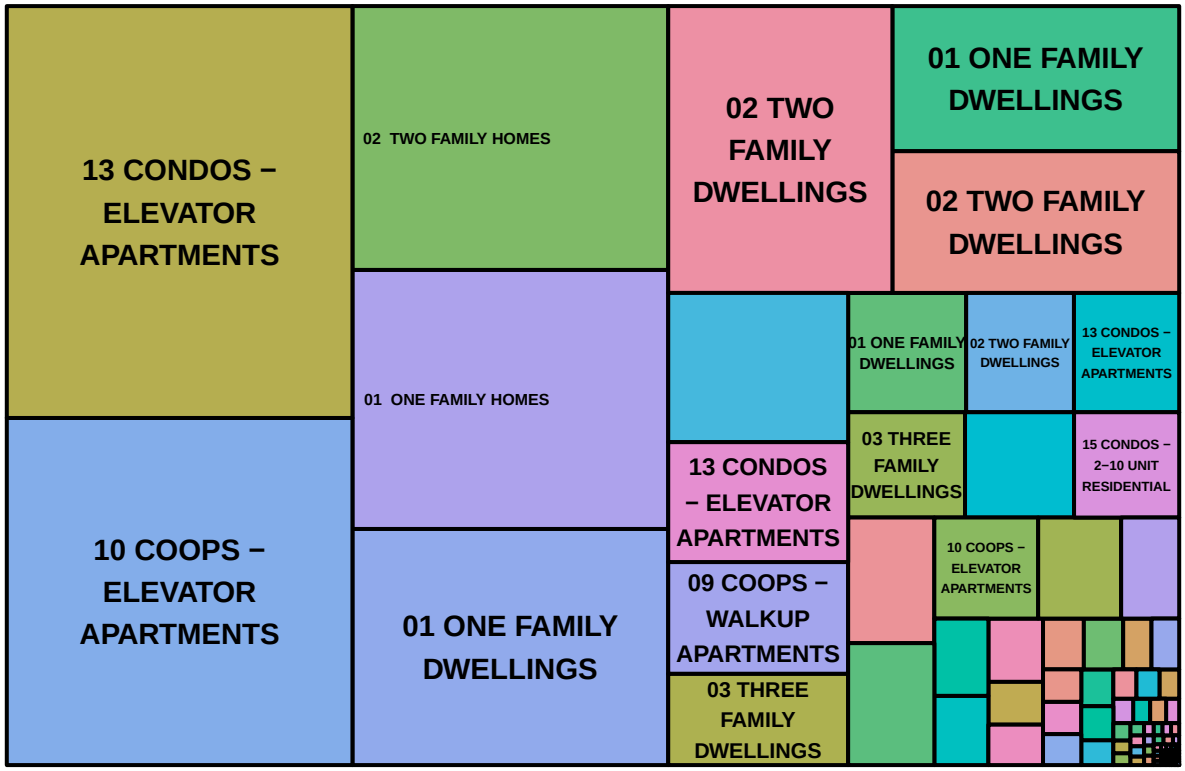
Categorical variables distributions

```
library(treemap)

# Categorical variables
categorical_vars <- c("NEIGHBORHOOD", "BUILDING.CLASS.CATEGORY",
                      "TAX.CLASS.AT.PRESENT", "BUILDING.CLASS.AT.PRESENT",
                      "TAX.CLASS.AT.TIME.OF.SALE", "BUILDING.CLASS.AT.TIME.OF.SALE")

# Create treemaps for each categorical variable
treemap_plots <- lapply(categorical_vars, function(var) {
  treemap(clean_nyc, index = var, vSize = "SALE.PRICE", title = paste("Treemap of", var))
})
```

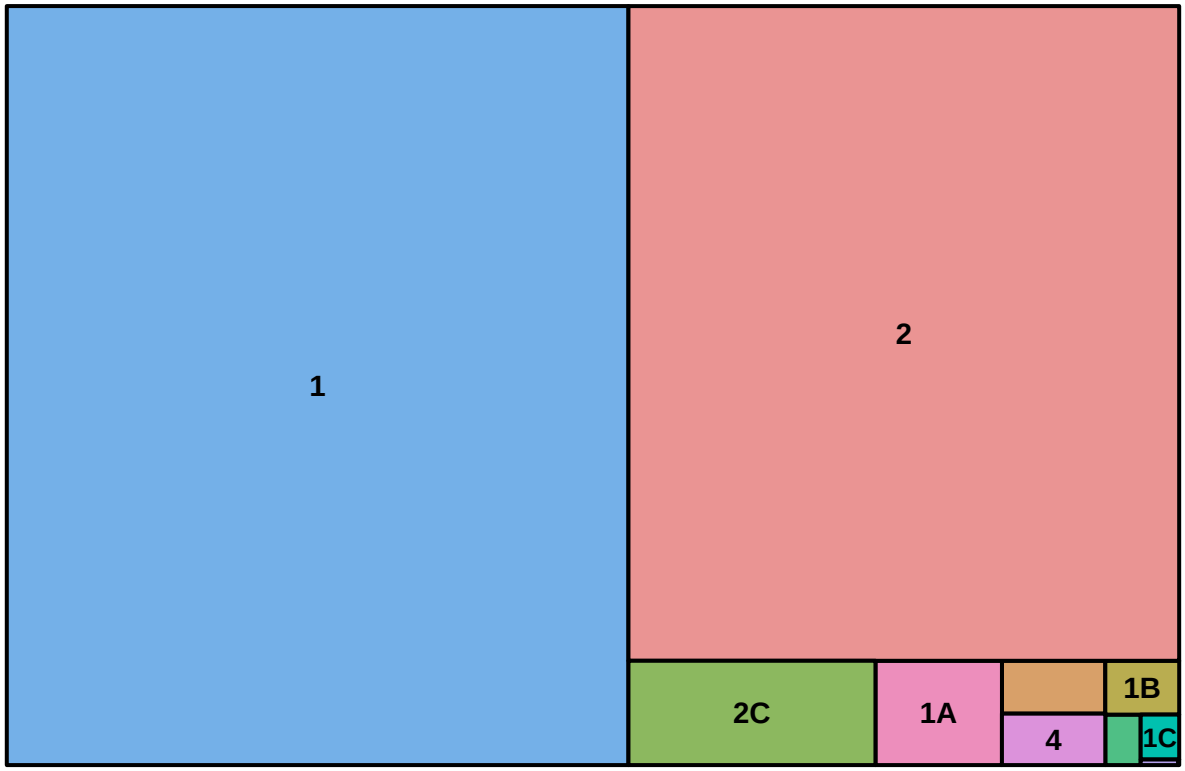
Treemap of NEIGHBORHOOD



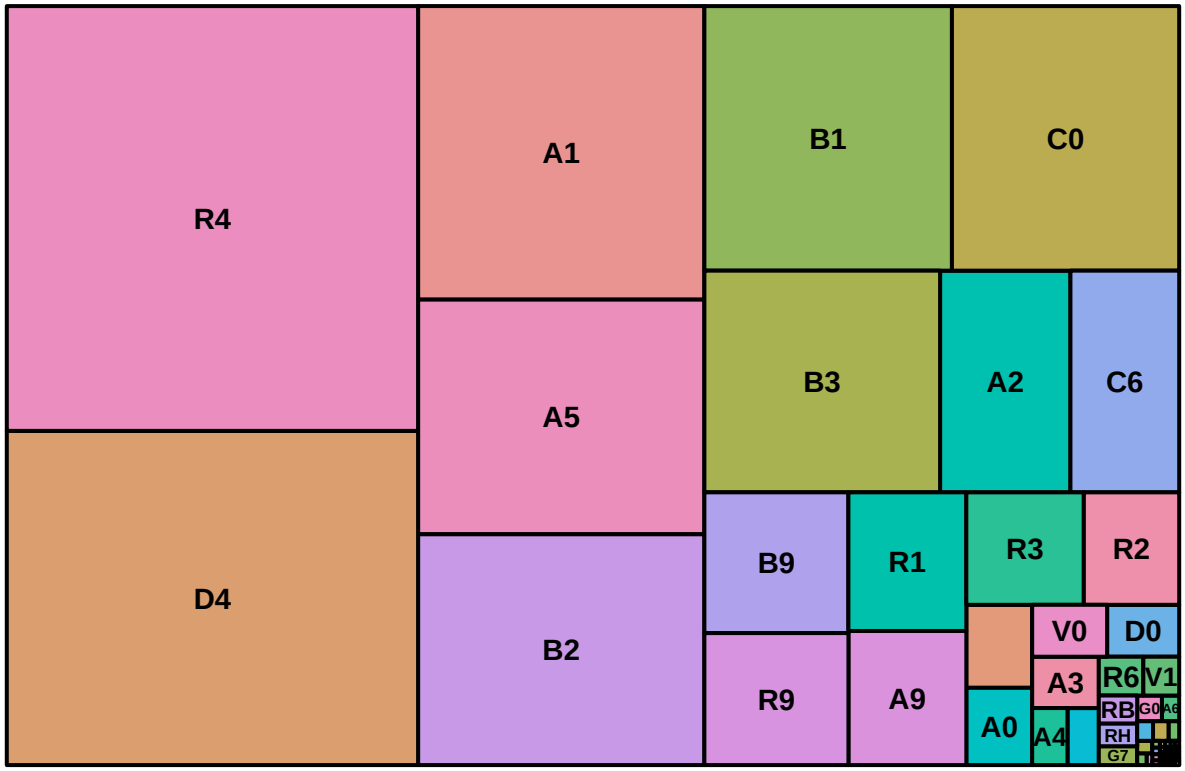
Treemap of BUILDING.CLASS.CATEGORY



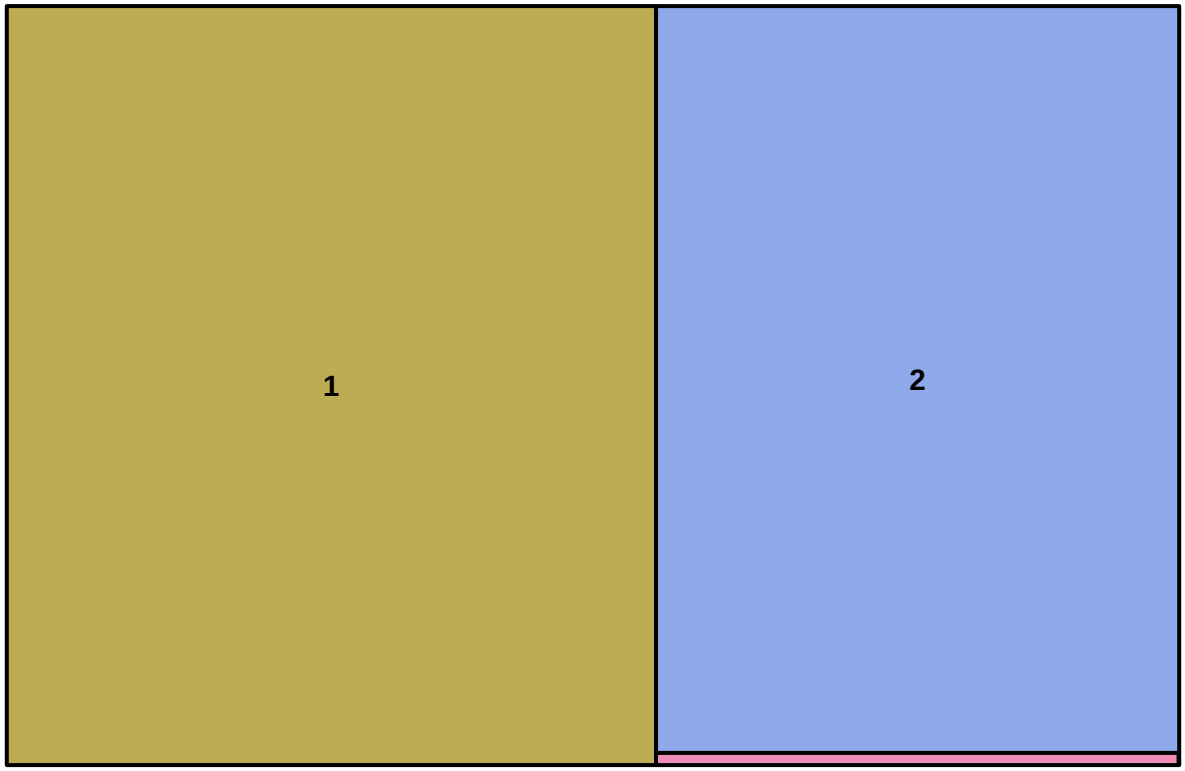
Treemap of TAX.CLASS.AT.PRESENT

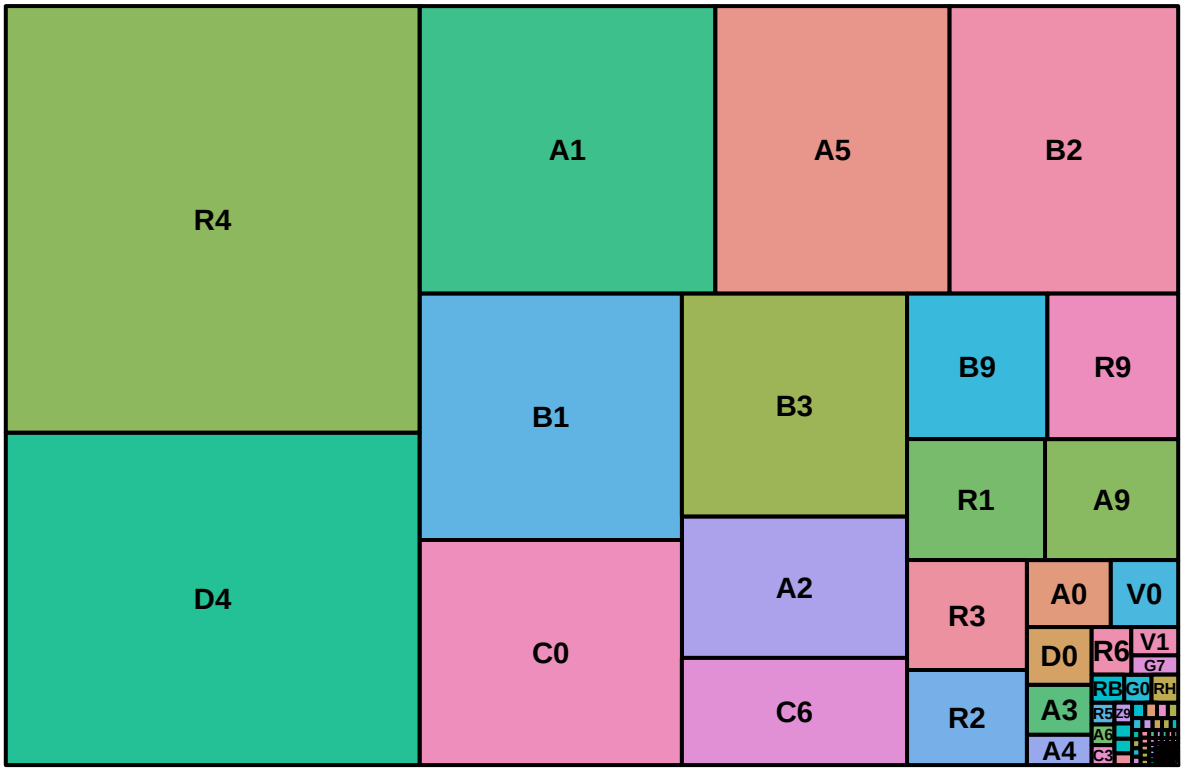


Treemap of BUILDING.CLASS.AT.PRESENT



Treemap of TAX.CLASS.AT.TIME.OF.SALE






```
# Output the treemaps
for (plot in treemap_plots) {
  plot
}
```

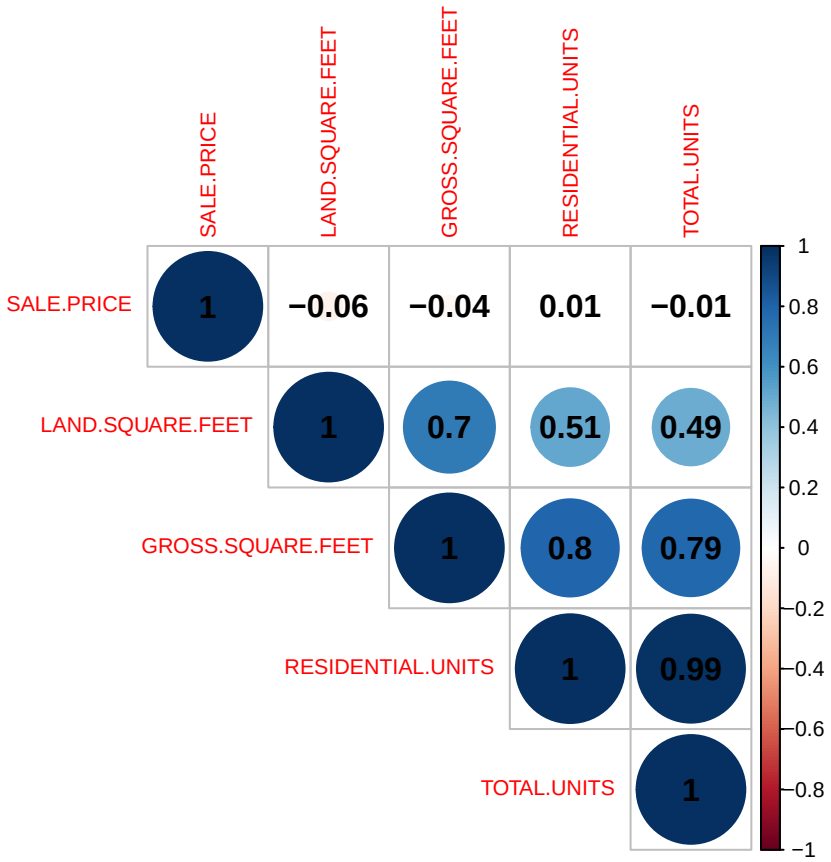
The darker blue circles indicate a stronger positive correlation, while the lighter blue and red circles represent weaker or negative correlations. For example, there is a strong positive correlation between gross square feet and land square feet, as well as between residential units and total units. However, sale price has a weak or slightly negative correlation with most of the other variables, suggesting that higher sale prices may not necessarily be associated with larger property sizes or more units.

```
print(sum(any(is.na(clean_nyc))))
```

```
## [1] 0
```

```
num_data <- as.data.frame(num_data)

# Drop COMMERCIAL.UNITS variable
num_data <- num_data[, !names(num_data) %in% "COMMERCIAL.UNITS"]


# Remove observations with missing, NaN, and infinite values
clean_data <- num_data[complete.cases(num_data) & !is.infinite(rowSums(num_data)), ]

# Calculate correlation matrix
correlation_matrix <- cor(clean_data)

# Plot correlation matrix
```

```
corrplot(correlation_matrix, method = "circle", type = "upper", order = "hclust",
         addCoef.col = "black", tl.cex = 0.7, cl.cex = 0.7)
```



Transform Categorical Variables to Factors

```
cat_vars <- c("BUILDING.CLASS.CATEGORY", "TAX.CLASS.AT.PRESENT", "BUILDING.CLASS.AT.PRESENT", "TAX.CLASS

# Convert categorical variables to factors
for (var in cat_vars) {
  clean_nyc[[var]] <- factor(clean_nyc[[var]])
}

# Verify the transformation
str(clean_nyc[cat_vars])
```

```
## 'data.frame':    1149281 obs. of  4 variables:
##  $ BUILDING.CLASS.CATEGORY  : Factor w/ 124 levels "                                                  ",..
##  $ TAX.CLASS.AT.PRESENT     : Factor w/ 11 levels " ","  ","1","1A",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ BUILDING.CLASS.AT.PRESENT: Factor w/ 129 levels " ","  ","A0",..: 7 12 17 17 16 12 7 12 16 13 ...
##  $ TAX.CLASS.AT.TIME.OF.SALE: Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 1 1 1 1 ...
```

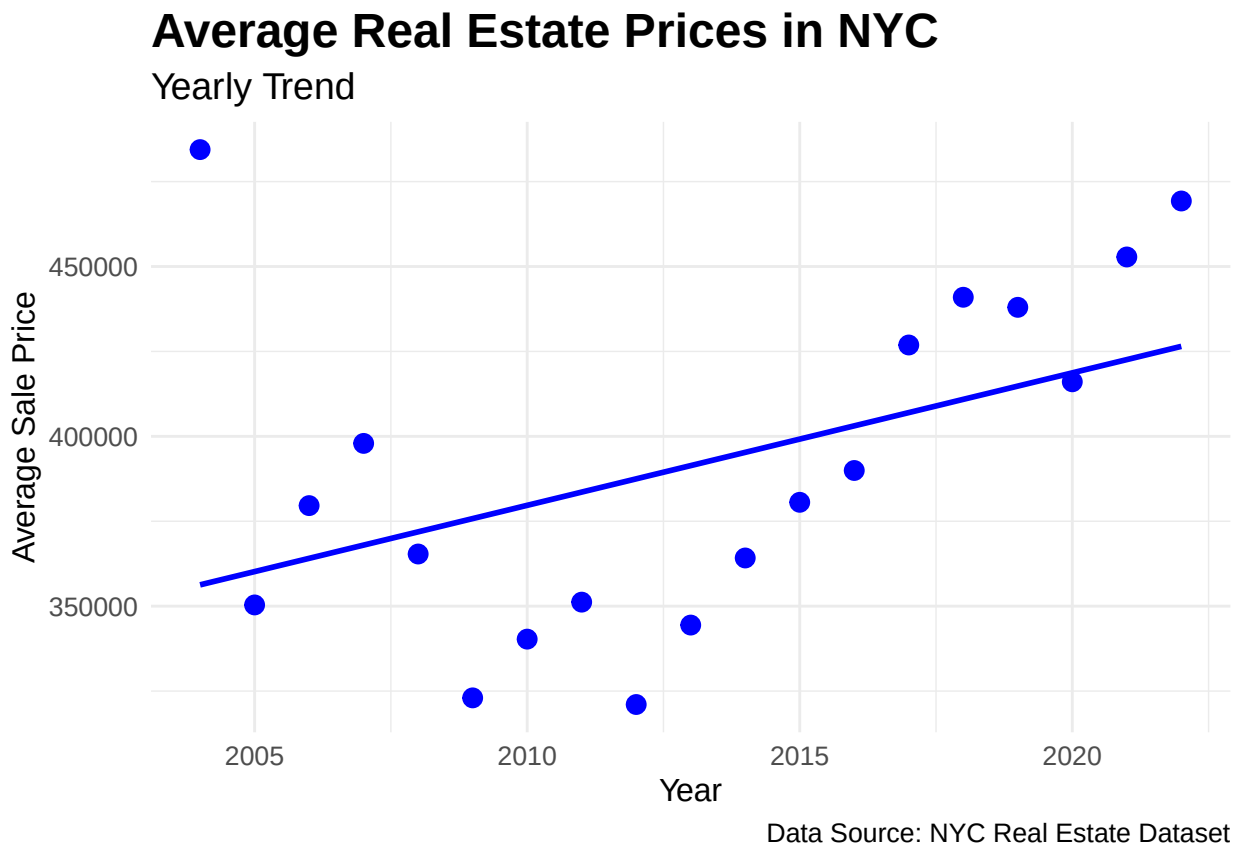
2. ANALYSIS

Over the years, there is a clear upward trend, indicating that the average real estate prices in NYC have been steadily increasing. The line exhibits a consistent upward slope, with prices rising from around $350,000 in 2005 to over $450,000 by 2020. Although there are some fluctuations in specific years, the overall trajectory demonstrates a significant increase in real estate prices in NYC over the 15-year period depicted in the graph.

```r
# Convert SALE_DATE to Date format
clean_nyc$SALE_DATE <- as.Date(clean_nyc$SALE.DATE)


# Group data by year and calculate average sale price per year
yearly_prices <- clean_nyc %>%
  mutate(year = lubridate::year(SALE_DATE)) %>%
  group_by(year) %>%
  summarise(avg_price = mean(SALE.PRICE))

# Create a line plot of average sale price over time (yearly)
ggplot(yearly_prices, aes(x = year, y = avg_price)) +
  geom_smooth(method = "lm", se = FALSE, color = "blue", linetype = "solid", size = 1) +  #smoother lin
  geom_point(color = "blue", size = 3) +
  labs(title = "Average Real Estate Prices in NYC",
       subtitle = "Yearly Trend",
       x = "Year",
       y = "Average Sale Price",
       caption = "Data Source: NYC Real Estate Dataset") +
  theme_minimal() +
  theme(plot.title = element_text(face = "bold", size = 18),
        plot.subtitle = element_text(size = 14),
        plot.caption = element_text(size = 10),
        axis.title.x = element_text(size = 12),
        axis.title.y = element_text(size = 12),
        axis.text.x = element_text(size = 10),
        axis.text.y = element_text(size = 10))
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
str(clean_nyc)
```

```
## 'data.frame':    1149281 obs. of  21 variables:
##  $ BOROUGH                       : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ NEIGHBORHOOD                  : chr  "ALPHABET CITY" "CHELSEA" "CHELSEA" "CLINTON" ...
##  $ BUILDING.CLASS.CATEGORY       : Factor w/ 124 levels "
##  $ TAX.CLASS.AT.PRESENT          : Factor w/ 11 levels " "," "  ","1","1A",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ BLOCK                         : int  374 722 772 1056 467 573 630 640 587 592 ...
##  $ LOT                           : int  46 72 42 24 52 59 2 70 64 11 ...
##  $ EASE.MENT                     : chr  "" "" "" "" ...
##  $ BUILDING.CLASS.AT.PRESENT     : Factor w/ 129 levels " "," "  ","A0",..: 7 12 17 17 16 12 7 12 16 13
##  $ ADDRESS                       : chr  "347 EAST 4TH STREET" "460 WEST 25TH STREET" "205 WEST 22ND S
##  $ APARTMENT.NUMBER              : chr  "" "" "" "" ...
##  $ ZIP.CODE                      : num  10009 10001 10011 10036 10003 ...
##  $ RESIDENTIAL.UNITS             : num  1 1 3 3 2 1 1 1 2 2 ...
##  $ TOTAL.UNITS                   : num  1 1 3 3 2 1 1 1 2 2 ...
##  $ LAND.SQUARE.FEET              : num  2116 1626 822 2007 1700 ...
##  $ GROSS.SQUARE.FEET             : num  4400 3721 2776 4304 3740 ...
##  $ YEAR.BUILT                    : num  1900 1910 1901 1901 1899 ...
##  $ TAX.CLASS.AT.TIME.OF.SALE     : Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 1 1 1 1 ...
##  $ BUILDING.CLASS.AT.TIME.OF.SALE: chr  "A4" "A9" "C0" "C0" ...
##  $ SALE.PRICE                    : num  399000 0 0 0 0 0 0 0 0 0 ...
##  $ SALE.DATE                     : chr  "2022-09-29 00:00:00" "2022-03-10 00:00:00" "2022-02-10 00:0
##  $ SALE_DATE                     : Date, format: "2022-09-29" "2022-03-10" ...
```

Key Factors Influencing Real Estate Prices

The linear regression model suggests that various factors significantly influence real estate prices in New York City. Notably, residential units, tax class, year built, sale date, tax class at time of sale, gross square feet, and land square feet all demonstrate statistically significant relationships with sale prices. However, the model's adjusted R-squared value of 0.06164 indicates that only about 6.164% of the variability in sale prices is explained by these factors. Additionally, the residuals' distribution reveals a considerable spread, indicating potential heteroscedasticity or unaccounted-for factors in the model.

```
# regression analysis
lm_model <- lm(SALE.PRICE ~ RESIDENTIAL.UNITS + TAX.CLASS.AT.PRESENT + YEAR.BUILT + SALE_DATE + TAX.CLAS
summary(lm_model)
```

```
##
## Call:
## lm(formula = SALE.PRICE ~ RESIDENTIAL.UNITS + TAX.CLASS.AT.PRESENT +
##     YEAR.BUILT + SALE_DATE + TAX.CLASS.AT.TIME.OF.SALE + GROSS.SQUARE.FEET +
##     LAND.SQUARE.FEET, data = clean_nyc)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -775674 -321371  -42793  227956 1621763
##
## Coefficients:
##                                Estimate   Std. Error t value
## (Intercept)                  233521.3963   11343.9261  20.586
## RESIDENTIAL.UNITS             39787.3198     706.6867  56.301
## TAX.CLASS.AT.PRESENT        -204939.9079   10595.8855 -19.341
## TAX.CLASS.AT.PRESENT1       -332391.9904   11120.0074 -29.891
## TAX.CLASS.AT.PRESENT1A      -267384.5810   11232.6810 -23.804
## TAX.CLASS.AT.PRESENT1B      -429788.1121   11315.1214 -37.984
## TAX.CLASS.AT.PRESENT1C      -103921.4024   13997.6874  -7.424
## TAX.CLASS.AT.PRESENT1D      -218870.2787   24130.4554  -9.070
## TAX.CLASS.AT.PRESENT2       -193655.0806    9870.1175 -19.620
## TAX.CLASS.AT.PRESENT2C      -102191.1926   10150.2536 -10.068
## TAX.CLASS.AT.PRESENT3       -407329.4338  136065.8992  -2.994
## TAX.CLASS.AT.PRESENT4       -314626.6682   11079.0815 -28.398
## YEAR.BUILT                       5.4097       0.7650   7.071
## SALE_DATE                       18.6537       0.1876  99.407
## TAX.CLASS.AT.TIME.OF.SALE2  102950.7101    6465.4286  15.923
## TAX.CLASS.AT.TIME.OF.SALE3  -63295.0519  130266.4276  -0.486
## TAX.CLASS.AT.TIME.OF.SALE4 -163815.1670    5939.8191 -27.579
## GROSS.SQUARE.FEET               13.8087       0.6852  20.154
## LAND.SQUARE.FEET                19.2470       0.3839  50.138
##                                        Pr(>|t|)
## (Intercept)                < 0.0000000000000002 ***
## RESIDENTIAL.UNITS          < 0.0000000000000002 ***
## TAX.CLASS.AT.PRESENT       < 0.0000000000000002 ***
## TAX.CLASS.AT.PRESENT1      < 0.0000000000000002 ***
## TAX.CLASS.AT.PRESENT1A     < 0.0000000000000002 ***
## TAX.CLASS.AT.PRESENT1B     < 0.0000000000000002 ***
## TAX.CLASS.AT.PRESENT1C        0.00000000000114 ***
## TAX.CLASS.AT.PRESENT1D     < 0.0000000000000002 ***
## TAX.CLASS.AT.PRESENT2      < 0.0000000000000002 ***
## TAX.CLASS.AT.PRESENT2C     < 0.0000000000000002 ***
```

```
## TAX.CLASS.AT.PRESENT3                   0.00276 **
## TAX.CLASS.AT.PRESENT4      < 0.0000000000000002 ***
## YEAR.BUILT                    0.000000000001536 ***
## SALE_DATE                  < 0.0000000000000002 ***
## TAX.CLASS.AT.TIME.OF.SALE2 < 0.0000000000000002 ***
## TAX.CLASS.AT.TIME.OF.SALE3              0.62705
## TAX.CLASS.AT.TIME.OF.SALE4 < 0.0000000000000002 ***
## GROSS.SQUARE.FEET          < 0.0000000000000002 ***
## LAND.SQUARE.FEET           < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 367900 on 1149262 degrees of freedom
## Multiple R-squared:  0.06165,    Adjusted R-squared:  0.06164
## F-statistic:  4195 on 18 and 1149262 DF,  p-value: < 0.00000000000000022
```

The stepwise regression process, with a starting AIC of 29457675, selected a final model with predictors including residential units, tax class, year built, sale date, tax class at the time of sale, gross square feet, and land square feet. This model, fitted using linear regression, reveals statistically significant relationships between these predictors and sale prices, as indicated by the low p-values and the coefficients' significance levels. However, the adjusted R-squared value remains low at 0.06164, suggesting that this model explains only a small portion of the variability in sale prices.

```
# Perform stepwise regression
stepwise_model <- step(lm_model)
```

```
## Start:  AIC=29457675
## SALE.PRICE ~ RESIDENTIAL.UNITS + TAX.CLASS.AT.PRESENT + YEAR.BUILT +
##     SALE_DATE + TAX.CLASS.AT.TIME.OF.SALE + GROSS.SQUARE.FEET +
##     LAND.SQUARE.FEET
##
##                              Df        Sum of Sq                RSS      AIC
## <none>                                           155591315571405088 29457675
## - YEAR.BUILT                  1    6769530540384 155598085101945472 29457723
## - GROSS.SQUARE.FEET           1   54990716199232 155646306287604320 29458080
## - TAX.CLASS.AT.TIME.OF.SALE   3  197247391038944 155788562962444032 29459126
## - LAND.SQUARE.FEET            1  340328837159072 155931644408564160 29460185
## - RESIDENTIAL.UNITS           1  429142788856672 156020458360261760 29460839
## - TAX.CLASS.AT.PRESENT       10  658928982669696 156250244554074784 29462512
## - SALE_DATE                   1 1337840771901312 156929156343306400 29467513
```

```
# Summary of the stepwise model
summary(stepwise_model)
```

```
##
## Call:
## lm(formula = SALE.PRICE ~ RESIDENTIAL.UNITS + TAX.CLASS.AT.PRESENT +
##     YEAR.BUILT + SALE_DATE + TAX.CLASS.AT.TIME.OF.SALE + GROSS.SQUARE.FEET +
##     LAND.SQUARE.FEET, data = clean_nyc)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -775674 -321371  -42793  227956 1621763
##
## Coefficients:
##                               Estimate   Std. Error t value
```

```
## (Intercept)                 233521.3963   11343.9261  20.586
## RESIDENTIAL.UNITS            39787.3198     706.6867  56.301
## TAX.CLASS.AT.PRESENT       -204939.9079   10595.8855 -19.341
## TAX.CLASS.AT.PRESENT1      -332391.9904   11120.0074 -29.891
## TAX.CLASS.AT.PRESENT1A     -267384.5810   11232.6810 -23.804
## TAX.CLASS.AT.PRESENT1B     -429788.1121   11315.1214 -37.984
## TAX.CLASS.AT.PRESENT1C     -103921.4024   13997.6874  -7.424
## TAX.CLASS.AT.PRESENT1D     -218870.2787   24130.4554  -9.070
## TAX.CLASS.AT.PRESENT2      -193655.0806    9870.1175 -19.620
## TAX.CLASS.AT.PRESENT2C     -102191.1926   10150.2536 -10.068
## TAX.CLASS.AT.PRESENT3      -407329.4338  136065.8992  -2.994
## TAX.CLASS.AT.PRESENT4      -314626.6682   11079.0815 -28.398
## YEAR.BUILT                      5.4097       0.7650   7.071
## SALE_DATE                      18.6537       0.1876  99.407
## TAX.CLASS.AT.TIME.OF.SALE2  102950.7101    6465.4286  15.923
## TAX.CLASS.AT.TIME.OF.SALE3  -63295.0519  130266.4276  -0.486
## TAX.CLASS.AT.TIME.OF.SALE4 -163815.1670    5939.8191 -27.579
## GROSS.SQUARE.FEET               13.8087       0.6852  20.154
## LAND.SQUARE.FEET                19.2470       0.3839  50.138
##                                        Pr(>|t|)
## (Intercept)                < 0.0000000000000002 ***
## RESIDENTIAL.UNITS          < 0.0000000000000002 ***
## TAX.CLASS.AT.PRESENT       < 0.0000000000000002 ***
## TAX.CLASS.AT.PRESENT1      < 0.0000000000000002 ***
## TAX.CLASS.AT.PRESENT1A     < 0.0000000000000002 ***
## TAX.CLASS.AT.PRESENT1B     < 0.0000000000000002 ***
## TAX.CLASS.AT.PRESENT1C        0.000000000000114 ***
## TAX.CLASS.AT.PRESENT1D     < 0.0000000000000002 ***
## TAX.CLASS.AT.PRESENT2      < 0.0000000000000002 ***
## TAX.CLASS.AT.PRESENT2C     < 0.0000000000000002 ***
## TAX.CLASS.AT.PRESENT3                   0.00276 **
## TAX.CLASS.AT.PRESENT4      < 0.0000000000000002 ***
## YEAR.BUILT                    0.000000000001536 ***
## SALE_DATE                  < 0.0000000000000002 ***
## TAX.CLASS.AT.TIME.OF.SALE2 < 0.0000000000000002 ***
## TAX.CLASS.AT.TIME.OF.SALE3              0.62705
## TAX.CLASS.AT.TIME.OF.SALE4 < 0.0000000000000002 ***
## GROSS.SQUARE.FEET          < 0.0000000000000002 ***
## LAND.SQUARE.FEET           < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 367900 on 1149262 degrees of freedom
## Multiple R-squared:  0.06165,    Adjusted R-squared:  0.06164
## F-statistic:  4195 on 18 and 1149262 DF,  p-value: < 0.00000000000000022
```

Considering that Normality was not satisfied

The next model fitted using generalized linear regression with a Gaussian family and an identity link function maintains predictors including residential units, tax class, year built, sale date, tax class at the time of sale, gross square feet, and land square feet. The coefficients and their significance remain consistent with the previous models. The null and residual deviances provide additional information on the goodness of fit, with the residual deviance being slightly lower than the null deviance, suggesting some level of model improvement.

```r
# Fit GLM with different error distribution and link function
glm_model <- glm(SALE.PRICE ~ RESIDENTIAL.UNITS + TAX.CLASS.AT.PRESENT + YEAR.BUILT + SALE_DATE + TAX.CL
                 data = clean_nyc,
                 family = gaussian(link = "identity"))
summary(glm_model)
```

```
##
## Call:
## glm(formula = SALE.PRICE ~ RESIDENTIAL.UNITS + TAX.CLASS.AT.PRESENT +
##     YEAR.BUILT + SALE_DATE + TAX.CLASS.AT.TIME.OF.SALE + GROSS.SQUARE.FEET +
##     LAND.SQUARE.FEET, family = gaussian(link = "identity"), data = clean_nyc)
##
## Coefficients:
##                                  Estimate   Std. Error t value
## (Intercept)                   233521.3963   11343.9261  20.586
## RESIDENTIAL.UNITS              39787.3198     706.6867  56.301
## TAX.CLASS.AT.PRESENT         -204939.9079   10595.8855 -19.341
## TAX.CLASS.AT.PRESENT1        -332391.9904   11120.0074 -29.891
## TAX.CLASS.AT.PRESENT1A       -267384.5810   11232.6810 -23.804
## TAX.CLASS.AT.PRESENT1B       -429788.1121   11315.1214 -37.984
## TAX.CLASS.AT.PRESENT1C       -103921.4024   13997.6874  -7.424
## TAX.CLASS.AT.PRESENT1D       -218870.2787   24130.4554  -9.070
## TAX.CLASS.AT.PRESENT2        -193655.0806    9870.1175 -19.620
## TAX.CLASS.AT.PRESENT2C       -102191.1926   10150.2536 -10.068
## TAX.CLASS.AT.PRESENT3        -407329.4338  136065.8992  -2.994
## TAX.CLASS.AT.PRESENT4        -314626.6682   11079.0815 -28.398
## YEAR.BUILT                         5.4097       0.7650   7.071
## SALE_DATE                         18.6537       0.1876  99.407
## TAX.CLASS.AT.TIME.OF.SALE2    102950.7101    6465.4286  15.923
## TAX.CLASS.AT.TIME.OF.SALE3    -63295.0519  130266.4276  -0.486
## TAX.CLASS.AT.TIME.OF.SALE4   -163815.1670    5939.8191 -27.579
## GROSS.SQUARE.FEET                 13.8087       0.6852  20.154
## LAND.SQUARE.FEET                  19.2470       0.3839  50.138
##                                        Pr(>|t|)
## (Intercept)                  < 0.0000000000000002 ***
## RESIDENTIAL.UNITS            < 0.0000000000000002 ***
## TAX.CLASS.AT.PRESENT         < 0.0000000000000002 ***
## TAX.CLASS.AT.PRESENT1        < 0.0000000000000002 ***
## TAX.CLASS.AT.PRESENT1A       < 0.0000000000000002 ***
## TAX.CLASS.AT.PRESENT1B       < 0.0000000000000002 ***
## TAX.CLASS.AT.PRESENT1C          0.000000000000114 ***
## TAX.CLASS.AT.PRESENT1D       < 0.0000000000000002 ***
## TAX.CLASS.AT.PRESENT2        < 0.0000000000000002 ***
## TAX.CLASS.AT.PRESENT2C       < 0.0000000000000002 ***
## TAX.CLASS.AT.PRESENT3                     0.00276 **
## TAX.CLASS.AT.PRESENT4        < 0.0000000000000002 ***
## YEAR.BUILT                      0.000000000001536 ***
## SALE_DATE                    < 0.0000000000000002 ***
## TAX.CLASS.AT.TIME.OF.SALE2   < 0.0000000000000002 ***
## TAX.CLASS.AT.TIME.OF.SALE3                0.62705
## TAX.CLASS.AT.TIME.OF.SALE4   < 0.0000000000000002 ***
## GROSS.SQUARE.FEET            < 0.0000000000000002 ***
## LAND.SQUARE.FEET             < 0.0000000000000002 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 135383677152)
##
##     Null deviance: 165814015036012544  on 1149280  degrees of freedom
## Residual deviance: 155591315571405312  on 1149262  degrees of freedom
## AIC: 32719196
##
## Number of Fisher Scoring iterations: 2
```

The final model, fitted using robust linear regression (rlm), estimates the intercept at $290,788.64. Each additional residential unit increases the sale price by $33,326.57. For the tax class at present, each category shows significant negative impacts on the sale price, with Tax Class 1B having the largest effect, reducing the price by $436,466.67. A one-unit increase in the year built is associated with a $9.05 increase in sale price. Similarly, each day increment in the sale date adds $14.35 to the sale price. Other variables, such as gross square feet and land square feet, also exhibit significant positive effects on the sale price. The residual standard error, indicating the model's accuracy, is $428,200.

```r
library(MASS)

# Fit robust linear regression model
lm_model_robust <- rlm(SALE.PRICE ~ RESIDENTIAL.UNITS + TAX.CLASS.AT.PRESENT + YEAR.BUILT + SALE_DATE +
                       data = clean_nyc)
summary(lm_model_robust)
```

```
##
## Call: rlm(formula = SALE.PRICE ~ RESIDENTIAL.UNITS + TAX.CLASS.AT.PRESENT +
##     YEAR.BUILT + SALE_DATE + TAX.CLASS.AT.TIME.OF.SALE + GROSS.SQUARE.FEET +
##     LAND.SQUARE.FEET, data = clean_nyc)
## Residuals:
##     Min      1Q  Median      3Q     Max
## -689412 -310434  -23950  245265 1635049
##
## Coefficients:
##                              Value        Std. Error   t value
## (Intercept)                  290788.6415   10750.8300      27.0480
## RESIDENTIAL.UNITS             33326.5719     669.7389      49.7605
## TAX.CLASS.AT.PRESENT       -231264.0838   10041.8993     -23.0299
## TAX.CLASS.AT.PRESENT1      -311659.8313   10538.6185     -29.5731
## TAX.CLASS.AT.PRESENT1A     -260618.5149   10645.4012     -24.4818
## TAX.CLASS.AT.PRESENT1B     -436466.6668   10723.5312     -40.7018
## TAX.CLASS.AT.PRESENT1C     -137901.4600   13265.8443     -10.3952
## TAX.CLASS.AT.PRESENT1D     -210361.9947   22868.8393      -9.1986
## TAX.CLASS.AT.PRESENT2      -211613.8859    9354.0768     -22.6226
## TAX.CLASS.AT.PRESENT2C     -112557.9271    9619.5664     -11.7009
## TAX.CLASS.AT.PRESENT3      -416669.1176  128951.9462      -3.2312
## TAX.CLASS.AT.PRESENT4      -316231.0765   10499.8323     -30.1177
## YEAR.BUILT                      9.0531       0.7250      12.4866
## SALE_DATE                      14.3494       0.1778      80.6878
## TAX.CLASS.AT.TIME.OF.SALE2   93751.7630    6127.3957      15.3004
## TAX.CLASS.AT.TIME.OF.SALE3  -49322.2028  123455.6892      -0.3995
## TAX.CLASS.AT.TIME.OF.SALE4 -168030.0667    5629.2667     -29.8494
## GROSS.SQUARE.FEET               5.3647       0.6493       8.2618
## LAND.SQUARE.FEET               18.0511       0.3638      49.6167
##
```

```
## Residual standard error: 428200 on 1149262 degrees of freedom
```