Unveiling Housing Dynamics in King County, WA

Gizem Gulsiye Guleli & Duae Marriam

2023-12-19

```
library(readr) #read daatset
library(dplyr) #data manipulation
library(ggplot2) # plots and visualizations
library(plotly) # interactive visualizations
library(reshape2)
library(ggpubr)  # For ggarrange function
library(tidyverse)
library(ggrepel)
library(ggalt)
library(ggplot2)
library(gridExtra)
library(sf) #to read shape files
library(sp) #convert to sf files
```

Summary

The goal of this project is to leverage advanced visualization techniques in R to analyze house prices in King County, Washington. The dataset, obtained from Kaggle, comprises 21 variables and 21,613 observations, spanning the period from 02 May 2014 to 27 May 2015

Objectives

Develop advanced visualizations to explore relationships between variables and understand patterns in house sales data. Try to identify and interpret factors contributing to the value of houses.

Data

Data Overview

```
house_data <- read.csv("kc_house_data.csv",header = TRUE, sep = ",")
```

Source: Kaggle *Link:* https://www.kaggle.com/datasets/shivachandel/kc-house-data/data *Variables:* 21 (id, date, price, bedrooms, bathrooms, sqft_living, sqft_lot, floors, waterfront, view, condition, grade, sqft_above, sqft_basement, yr_built, yr_renovated, zipcode, lat, long, sqft_living15, sqft_lot15) *Observations:* 21.613 *Period:* 02 May 2014 to 27 May 2015 *Geographic coverage:* King County, including Seattle

```r
str(house_data)
```

Structure of Dataset

```
## 'data.frame':    21613 obs. of  21 variables:
##  $ id           : num  7.13e+09 6.41e+09 5.63e+09 2.49e+09 1.95e+09 ...
##  $ date         : chr  "20141013T000000" "20141209T000000" "20150225T000000" "20141209T000000" ...
##  $ price        : num  221900 538000 180000 604000 510000 ...
##  $ bedrooms     : int  3 3 2 4 3 4 3 3 3 3 ...
##  $ bathrooms    : num  1 2.25 1 3 2 4.5 2.25 1.5 1 2.5 ...
##  $ sqft_living  : int  1180 2570 770 1960 1680 5420 1715 1060 1780 1890 ...
##  $ sqft_lot     : int  5650 7242 10000 5000 8080 101930 6819 9711 7470 6560 ...
##  $ floors       : num  1 2 1 1 1 1 2 1 1 2 ...
##  $ waterfront   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ view         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ condition    : int  3 3 3 5 3 3 3 3 3 3 ...
##  $ grade        : int  7 7 6 7 8 11 7 7 7 7 ...
##  $ sqft_above   : int  1180 2170 770 1050 1680 3890 1715 1060 1050 1890 ...
##  $ sqft_basement: int  0 400 0 910 0 1530 0 0 730 0 ...
##  $ yr_built     : int  1955 1951 1933 1965 1987 2001 1995 1963 1960 2003 ...
##  $ yr_renovated : int  0 1991 0 0 0 0 0 0 0 0 ...
##  $ zipcode      : int  98178 98125 98028 98136 98074 98053 98003 98198 98146 98038 ...
##  $ lat          : num  47.5 47.7 47.7 47.5 47.6 ...
##  $ long         : num  -122 -122 -122 -122 -122 ...
##  $ sqft_living15: int  1340 1690 2720 1360 1800 4760 2238 1650 1780 2390 ...
##  $ sqft_lot15   : int  5650 7639 8062 5000 7503 101930 6819 9711 8113 7570 ...
```

The dataset consists of 21,613 observations (rows) and 21 variables (columns). The variables include information such as id, date, price, bedrooms, bathrooms, sqft_living, sqft_lot, floors, waterfront, view, condition, grade, sqft_above, sqft_basement, yr_built, yr_renovated, zipcode, lat, long, sqft_living15, and sqft_lot15.

The id variable appears to be a unique identifier for each observation. All variables structured as numeric/integer except the date variable which is currently stored as a character type. It might be useful to convert it to a date type for time-related analyses. So we will first convert the date variable from character to a date type.

It's important to note that while all variables may be structured as numeric, certain variables, despite their numeric representation, hold categorical significance. These categorical variables are essentially numerically coded to represent different categories or levels within the dataset. This nuance is crucial to consider when interpreting and analyzing the data.

```r
house_data$date <- as.Date(house_data$date, format = "%Y%m%dT%H%M%S")

# Verify the changes
str(house_data$date)
```

```
##  Date[1:21613], format: "2014-10-13" "2014-12-09" "2015-02-25" "2014-12-09" "2015-02-18" ...
```

```r
summary(house_data)
```

Summary Statistics

```
##        id                  date                price            bedrooms
##  Min.   :1.000e+06   Min.   :2014-05-02   Min.   :  75000   Min.   : 0.000
##  1st Qu.:2.123e+09   1st Qu.:2014-07-22   1st Qu.: 321950   1st Qu.: 3.000
##  Median :3.905e+09   Median :2014-10-16   Median : 450000   Median : 3.000
##  Mean   :4.580e+09   Mean   :2014-10-29   Mean   : 540088   Mean   : 3.371
##  3rd Qu.:7.309e+09   3rd Qu.:2015-02-17   3rd Qu.: 645000   3rd Qu.: 4.000
##  Max.   :9.900e+09   Max.   :2015-05-27   Max.   :7700000   Max.   :33.000
##
##    bathrooms      sqft_living       sqft_lot           floors
##  Min.   :0.000   Min.   :  290   Min.   :    520   Min.   :1.000
##  1st Qu.:1.750   1st Qu.: 1427   1st Qu.:   5040   1st Qu.:1.000
##  Median :2.250   Median : 1910   Median :   7618   Median :1.500
##  Mean   :2.115   Mean   : 2080   Mean   :  15107   Mean   :1.494
##  3rd Qu.:2.500   3rd Qu.: 2550   3rd Qu.:  10688   3rd Qu.:2.000
##  Max.   :8.000   Max.   :13540   Max.   :1651359   Max.   :3.500
##
##    waterfront            view          condition         grade
##  Min.   :0.000000   Min.   :0.0000   Min.   :1.000   Min.   : 1.000
##  1st Qu.:0.000000   1st Qu.:0.0000   1st Qu.:3.000   1st Qu.: 7.000
##  Median :0.000000   Median :0.0000   Median :3.000   Median : 7.000
##  Mean   :0.007542   Mean   :0.2343   Mean   :3.409   Mean   : 7.657
##  3rd Qu.:0.000000   3rd Qu.:0.0000   3rd Qu.:4.000   3rd Qu.: 8.000
##  Max.   :1.000000   Max.   :4.0000   Max.   :5.000   Max.   :13.000
##
##    sqft_above   sqft_basement       yr_built     yr_renovated
##  Min.   : 290   Min.   :   0.0   Min.   :1900   Min.   :   0.0
##  1st Qu.:1190   1st Qu.:   0.0   1st Qu.:1951   1st Qu.:   0.0
##  Median :1560   Median :   0.0   Median :1975   Median :   0.0
##  Mean   :1788   Mean   : 291.5   Mean   :1971   Mean   :  84.4
##  3rd Qu.:2210   3rd Qu.: 560.0   3rd Qu.:1997   3rd Qu.:   0.0
##  Max.   :9410   Max.   :4820.0   Max.   :2015   Max.   :2015.0
##  NA's   :2
##     zipcode           lat             long        sqft_living15
##  Min.   :98001   Min.   :47.16   Min.   :-122.5   Min.   : 399
##  1st Qu.:98033   1st Qu.:47.47   1st Qu.:-122.3   1st Qu.:1490
##  Median :98065   Median :47.57   Median :-122.2   Median :1840
##  Mean   :98078   Mean   :47.56   Mean   :-122.2   Mean   :1987
##  3rd Qu.:98118   3rd Qu.:47.68   3rd Qu.:-122.1   3rd Qu.:2360
##  Max.   :98199   Max.   :47.78   Max.   :-121.3   Max.   :6210
##
##    sqft_lot15
##  Min.   :   651
##  1st Qu.:  5100
##  Median :  7620
##  Mean   : 12768
##  3rd Qu.: 10083
##  Max.   :871200
##
```

id - The `id` variable represents a unique identifier for each home sold.

date - The `date` variable, contains information about the date of the house sale and spans from May 2, 2014, to May 27, 2015.

price: - The `price` variable is the **dependent variable** and shows a wide range, with the minimum house price at $75,000 and the maximum at $7,700,000. - The median house price is $450,000, and the mean is $540,088.

bedrooms and bathrooms: - The variables related to the number of bedrooms and bathrooms (0.5 accounts for a room with a toilet but no shower) exhibit varying ranges and distributions. - The number of bedrooms ranges from 0 to 33, with a mean of approximately 3.37. - The number of bathrooms ranges from 0 to 8, with a mean of approximately 2.12.

sqft_living and sqft_lot: - These variables represent the size of houses. - `sqft_living` reflects the Square footage of the apartments interior living area, ranging from 290 to 13,540 square feet, with a mean of 2080. - `sqft_lot` represents the lot size, ranging from 520 to 1,651,359 square feet, with a mean of 15,107.

floors -The `floors` variable is represents the levels of the houses. The majority of houses have 1 or 1.5 floors. - Notably, there seems to be a common occurrence of houses with 1.5 floors, while the mean is approximately 1.494. - This suggests that many houses have a split-level design or additional space on an upper level, contributing to the fractional floor values.

waterfront:
- The `waterfront` variable is a dummy variable mostly 0 , represents the property has no waterfront view and 1 for with waterfront.

view and condition:
- `view` represents the overall view rating (0 to 4) with a mean of 0.23 -`condition` represents the overall condition rating (0 to 5) with a mean of 3.41 for `condition`.

grade: - `grade` represents the overall grade given to the housing unit and ranges from 1 to 13 where 1-3 falls short of building construction and design, 7 has an average level of construction and design, and 11-13 have a high quality level of construction and design.

sqft_above, and sqft_basement: - `sqft_above` and `sqft_basement` show the square footage above ground and is below ground level¶ (in the basement), respectively.

yr_built, yr_renovated: - Houses were built between 1900 and 2015 (`yr_built`), with the majority built in the mid to late 20th century. - `yr_renovated` indicates the last renovation year, with a mean of 84.4 and many zero values, suggesting no renovations.

Geographical Information (lat, long, zipcode): - `lat` and `long` provide latitude and longitude information of house locations, respectively. - `zipcode` represents the zip code of the house location.

sqft_living15 and sqft_lot15: - `sqft_living15` and `sqft_lot15` indicate the living room and lot size in 2015, reflecting potential renovations or changes. (? some sources mention it differently and main source couldnt find !!!)

These summary statistics provide an overview of the distribution and characteristics of each numeric variable in the dataset, with a specific focus on understanding the relationships with the **dependent variable, 'price.'**

Missing Values

- Most variables in the dataset have complete data; however, it's worth noting that `sqft_above` has two missing values (NA's).
- Given the small number of missing values (only two observations) in relation to the overall dataset size, we have decided to remove these specific observations. This decision is based on considering the number of observations and the minimal impact on the overall analysis.
- Removing these observations ensures that the dataset remains largely complete and is a reasonable approach in this context.

```r
# Remove observations with missing values in 'sqft_above'
house_data2 <- house_data[complete.cases(house_data$sqft_above), ]
```

Data Exploration

In organizing our variables by type, we enhance the precision of our analysis and visualization methods. This thoughtful categorization enables us to apply tailored techniques to each variable type, ensuring more insightful and nuanced exploration of the dataset.

```r
# All variables
all_vars <- house_data2[, c("price", "bedrooms", "bathrooms", "sqft_living",
                            "sqft_lot","floors", "waterfront", "view", "condition",
                            "grade", "sqft_above", "sqft_basement", "yr_built",
                            "yr_renovated", "zipcode", "lat", "long", "sqft_living15",
                            "sqft_lot15")]

# Continuous Numeric Variables
cont_vars <- c("price", "sqft_living","sqft_living15","sqft_lot",
               "sqft_lot15","sqft_above", "sqft_basement")

# Discrete Numeric Variables
disc_vars <- c("bedrooms", "floors", "bathrooms")

# Categorical Variables
cat_vars <- c("waterfront", "view", "condition", "grade")

# Date Variables
date_vars <- c("date", "yr_built", "yr_renovated")

# Geographical Variables
geo_vars <- c("lat", "long", "zipcode")
```

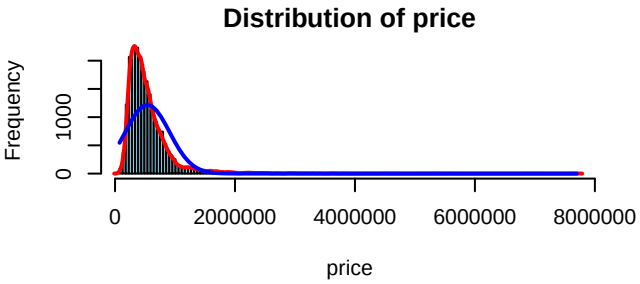
Exploratory Data Analysis (EDA)

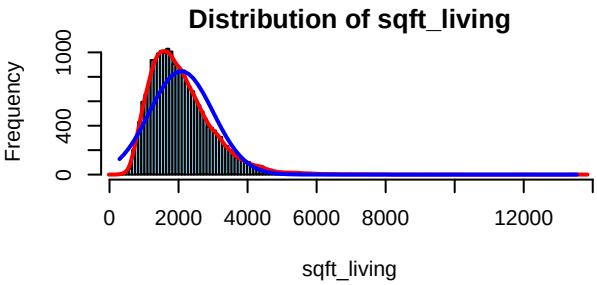
Utilize various R packages (e.g., ggplot2, plotly) for data exploration. Conduct correlation analysis, distribution analysis.

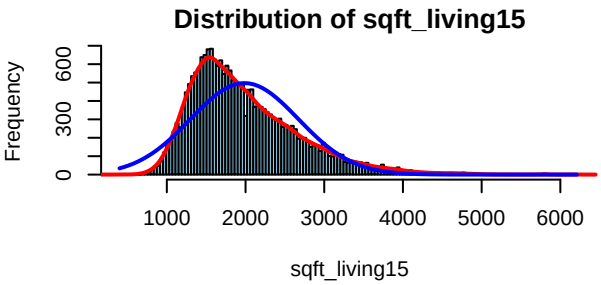
```r
#  to display numeric values without scientific notation and with more digits
options(scipen = 999, digits = 9)
```

```r
# Set up a layout grid
par(mfrow = c(3, 2), mar = c(4, 4, 2, 1))  # Adjust margins for better appearance

# Create histograms for  numeric variables
for (cont in cont_vars) {
  # Determine appropriate bin width based on the range and number of observations
  bin_width <- (max(house_data2[[cont]]) - min(house_data2[[cont]])) /
    sqrt(length(house_data2[[cont]]))
```

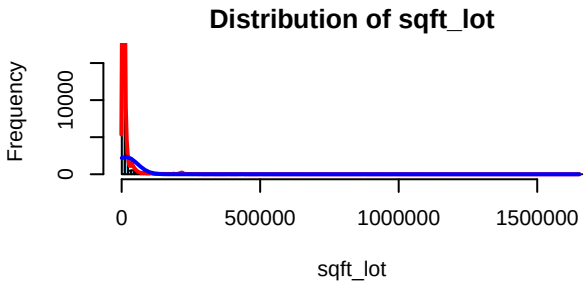
```r
# Create histogram with scaled x-axis
hist(house_data2[[cont]], main = paste("Distribution of", cont), xlab = cont,
     col = "skyblue", breaks = seq(min(house_data2[[cont]]),
                                   max(house_data2[[cont]]) + bin_width, bin_width))

# Add smoother distribution line
  density_curve <- density(house_data2[[cont]], bw = "nrd0")
  lines(density_curve$x, density_curve$y * bin_width * length(house_data2[[cont]]),
        col = "red", lwd = 2)

# Add normal distribution line
mu <- mean(house_data2[[cont]])
sigma <- sd(house_data2[[cont]])
x <- seq(min(house_data2[[cont]]), max(house_data2[[cont]]), length = 100)
y <- dnorm(x, mean = mu, sd = sigma) * bin_width * length(house_data2[[cont]])
lines(x, y, col = "blue", lwd = 2)

# Identify potential outliers using a boxplot
boxplot(house_data2[[cont]], main = paste("Boxplot of", cont), col = "lightblue",
        border = "black", horizontal = TRUE)
}
```

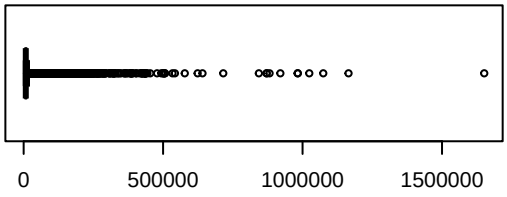


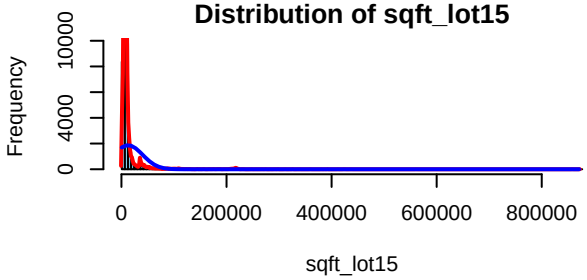


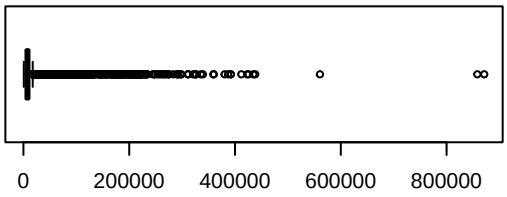


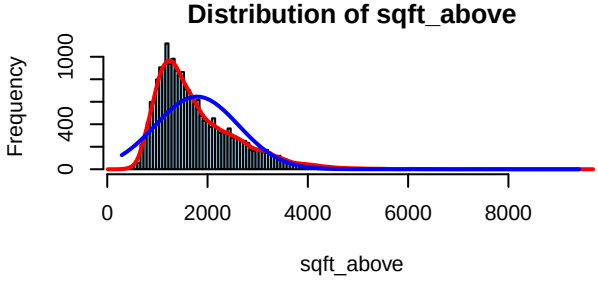
Distribution of Contunious Numeric Variables

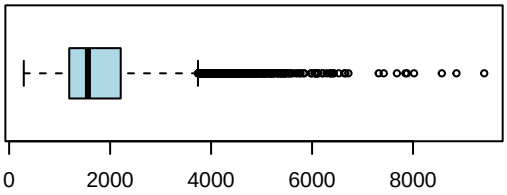




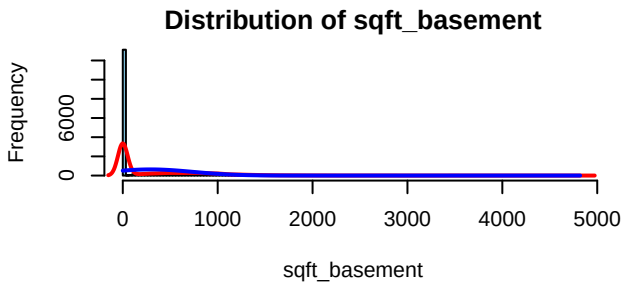


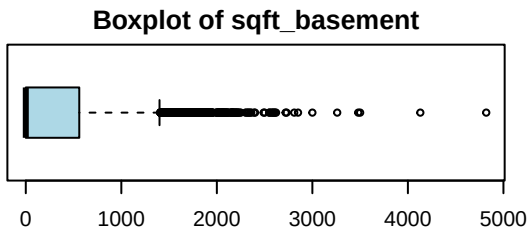






```r
# Reset the plotting layout
par(mfrow = c(1, 1))
```



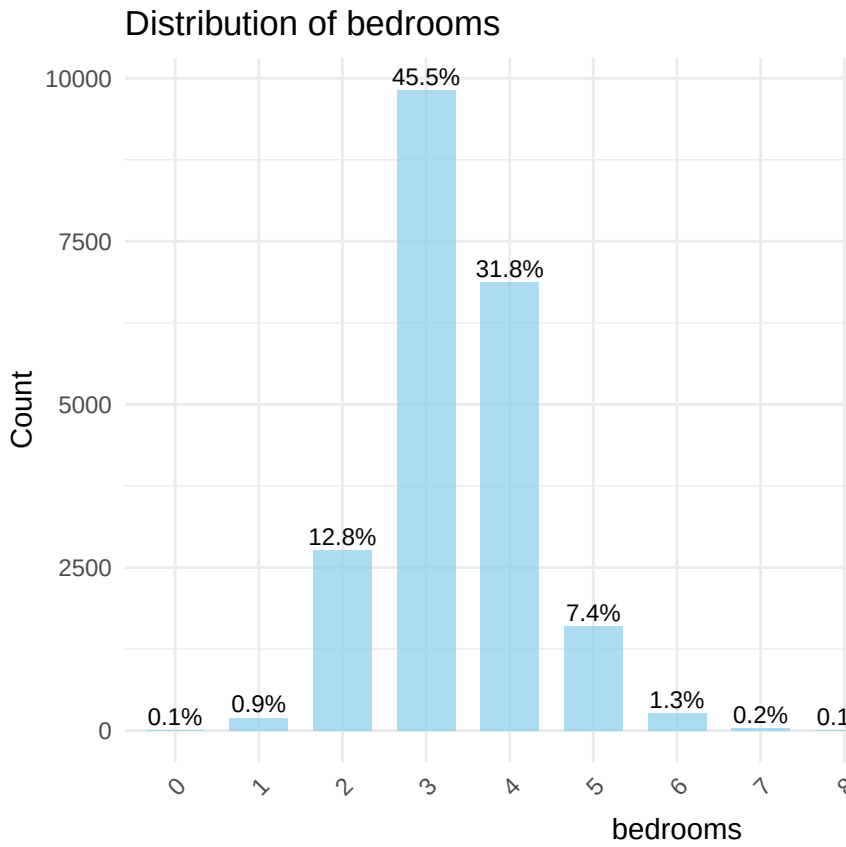


The visual inspection of the plots suggests that distribution of the variables are skewed right/ non-normal distributions with a considerable number of outliers. Given the context of the dataset, where very luxurious or unique properties may contribute to these extreme values, it is justifiable to observe such outliers.

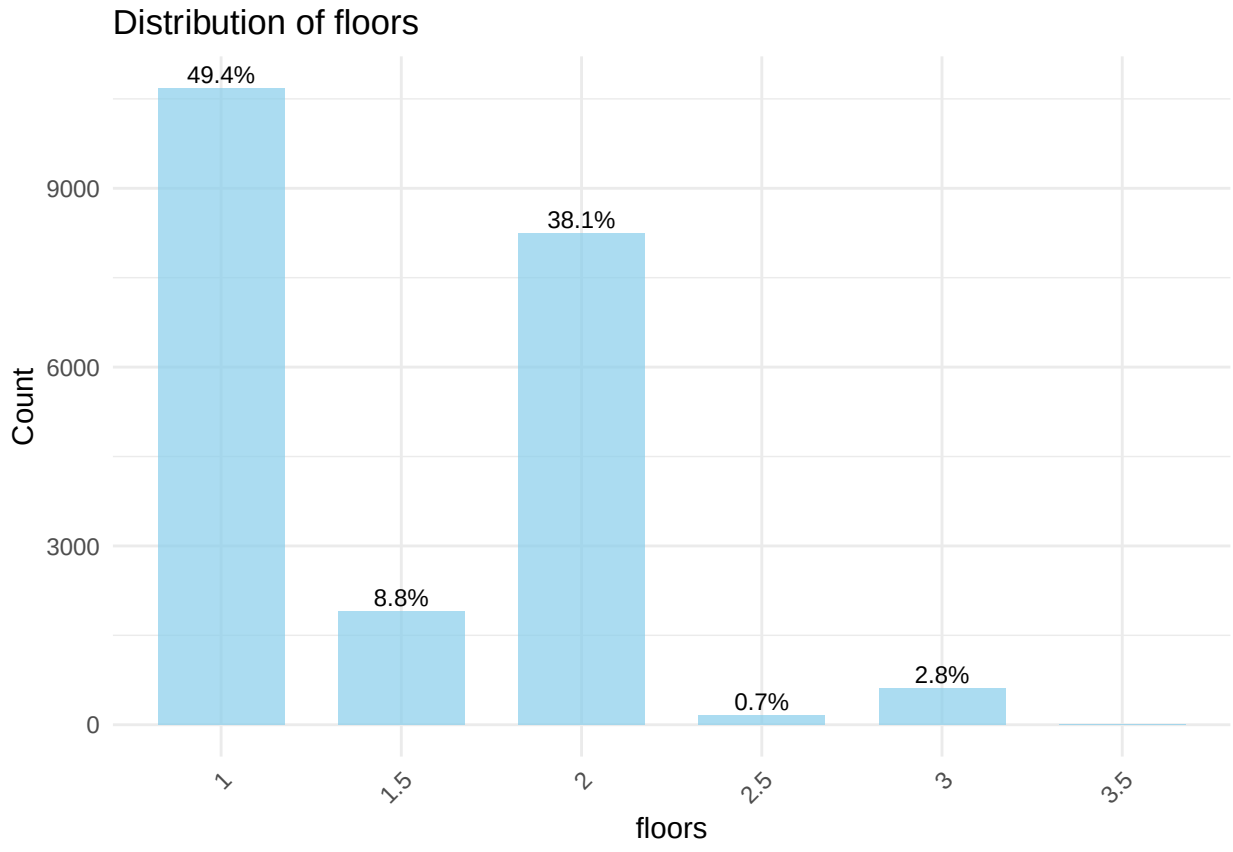
Instead of removing or transforming these outliers, a more suitable strategy might involve employing robust statistical methods to handle outliers problem for further analysis. Robust methods are designed to be less sensitive to extreme values, allowing for a more reliable analysis that acknowledges the presence of these high-end properties without disproportionately impacting the results.

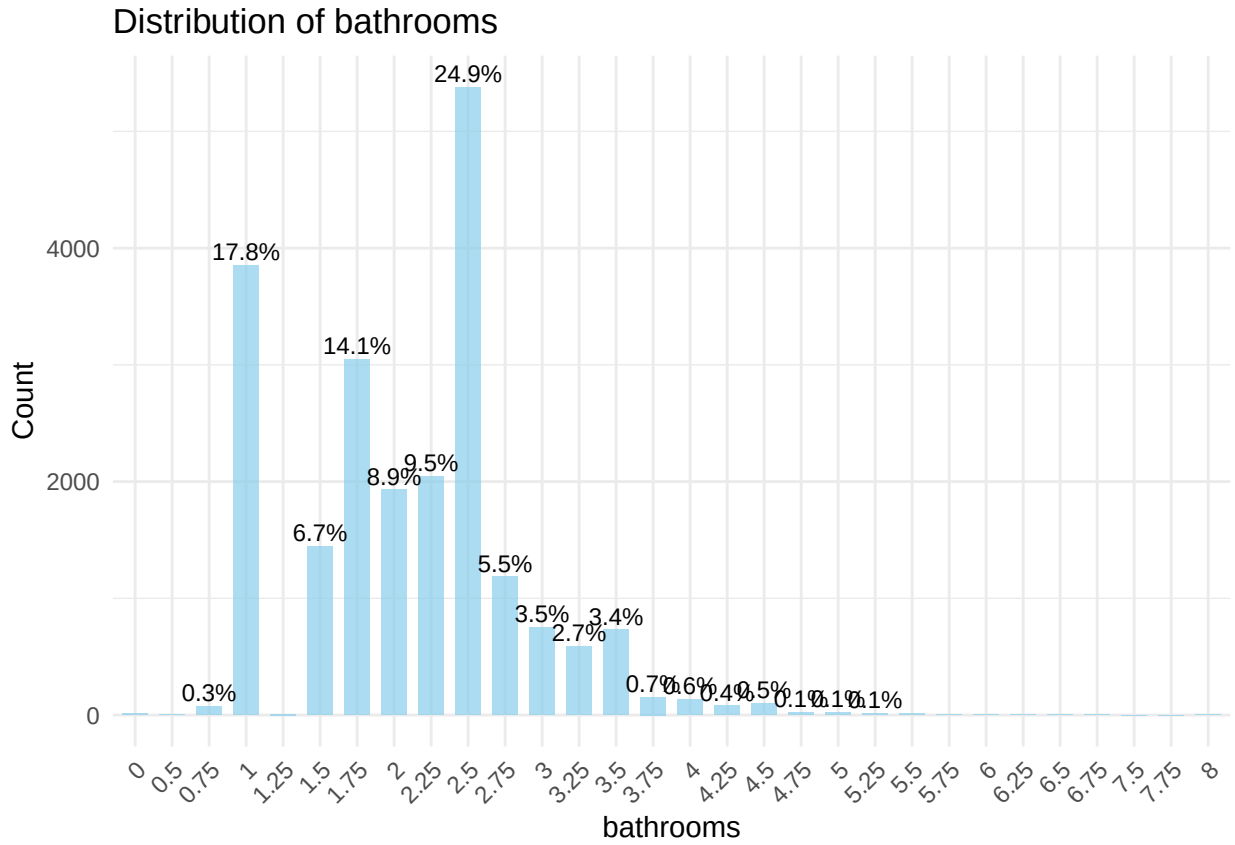
```r
# Visualize the Distribution
for (disc in disc_vars) {
  # Convert discrete numeric variables to factors
  house_data2[[disc]] <- as.factor(house_data2[[disc]])

  # Create bar plots for discrete numeric variables
  bar_plot <- ggplot(house_data2, aes(x = !!sym(disc), fill = !!sym(disc))) +
    geom_bar(position = "dodge", fill = "skyblue", alpha = 0.7, width = 0.7) +
    labs(title = paste("Distribution of", disc), x = disc, y = "Count") +
    theme_minimal() +
    geom_text(stat = "count",
```

```
        aes(label = ifelse(round(after_stat(count)/sum(after_stat(count)), 3) > 0,
         scales::percent(round(after_stat(count)/sum(after_stat(count)), 3)), ""),
         position = position_dodge(0.7), vjust = -0.3, size= 3) +  # Add percentage
    #labels only if count > 0
    theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotate x-axis labels

  # Display the plot
  print(bar_plot)
}
```



Distribution of Discrete Numeric Variables





Bedrooms :

The distribution of bedrooms in the dataset reveals a clear preference for houses with 3 bedrooms, constituting nearly half of the entries (45.5%). 4-bedroom homes follow closely with 31.8%, and 2 and 5-bedroom configurations are also prevalent, making up 12.8% and 7.4%. However, 0-bedroom and 1-bedroom houses have notably lower percentages with approximately 0.1% and 0.9%, respectively. The distribution is positively skewed, with a peak around 3 bedrooms.

Bathrooms Configuration:

The dataset showcases a diverse distribution of bathrooms. Houses with 2.5 bathrooms are most common, representing 24.9%. Additionally, 1 bathroom and 1.75 bathrooms are prevalent at 17.8% and 14.1%, respectively. The distribution exhibits multiple peaks, suggesting a variety of bathroom count configurations in the dataset.

In the United States, bathrooms are generally categorized as master bathroom, containing a varied shower and a tub that is adjoining to a master bedroom, a “full bathroom” (or “full bath”), containing four plumbing fixtures: bathtub/shower, or (separate shower), toilet, and sink; “half (1/2) bath” (or “powder room”) containing just a toilet and sink; and “3/4 bath” containing toilet, sink, and shower, although the terms vary from market to market. In some U.S. markets, a toilet, sink, and shower are considered a “full bath”. (wikipedia)

Floor Counts:

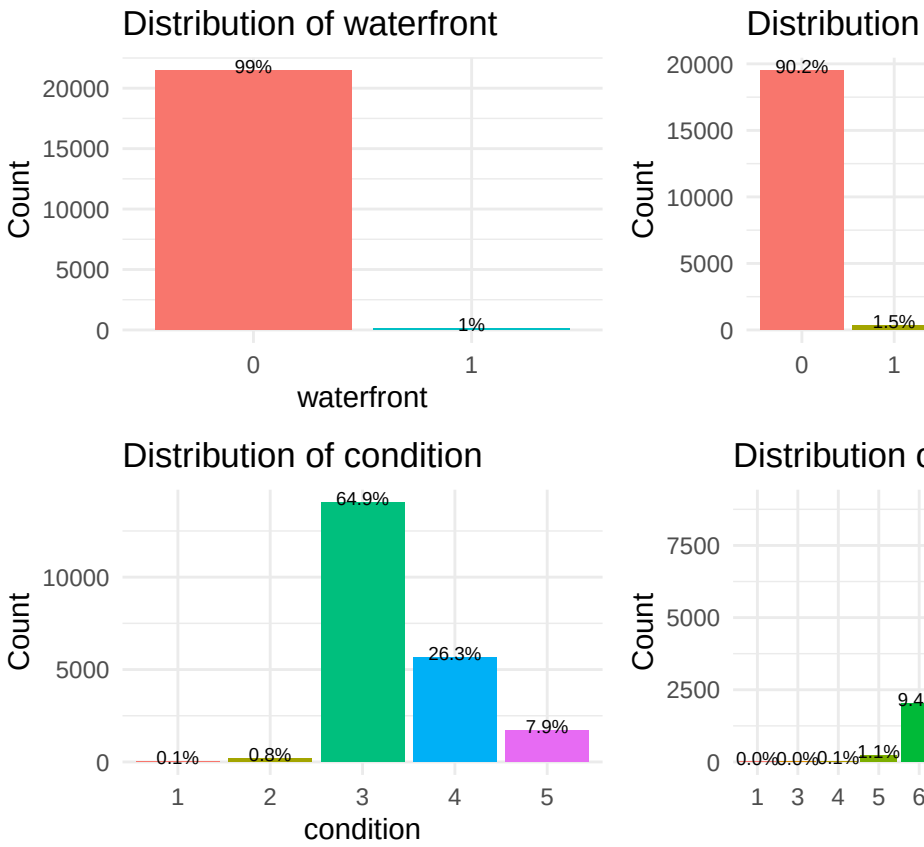
When considering the number of floors, Houses with 1 floor are predominant, making up 49.4% of the dataset. 2 floors houses follow closely at 38.1%, with 1.5 floors representing 8.8%. The distribution is skewed towards fewer floors, with a sharp decline for houses with more than 2 floors.

Note on 0 Values:

In the context of houses requiring bedrooms and bathrooms, the presence of 0 values in these categories

may indicate missing or incomplete data. It's uncommon for a house to have zero bedrooms or bathrooms. Investigating and addressing the reasons behind these zero values is crucial for ensuring the quality and accuracy of the dataset, as well as the reliability of any analyses conducted.

```
# Set up a layout grid
grid_layout <- matrix(c(1, 2, 3, 4), nrow = 2, byrow = TRUE)

# Create more advanced bar plots for categorical variables
plots <- list()
for (cat in cat_vars) {
  # Convert categorical variables to factors
  house_data2[[cat]] <- as.factor(house_data2[[cat]])

  # Create bar plots for categorical variables
  bar_plot <- ggplot(house_data2, aes(x = factor(!!sym(cat)), fill =
          factor(!!sym(cat)))) + geom_bar() +geom_text(stat = "count",
    aes(label = scales::percent(round(after_stat(count)/sum(after_stat(count)), 3))),
    vjust = 0.2, size= 2.5) +  # Add percentage labels
    labs(title = paste("Distribution of", cat), x = cat, y = "Count") +
    theme_minimal() +
    theme(legend.position = "none")

  # Add the plot to the list
  plots[[cat]] <- bar_plot
}

# Arrange the plots in a 2x2 grid
grid.arrange(grobs = plots, layout_matrix = grid_layout)
```



Distribution of Categorical Variables

```
# Reset the plotting layout
par(mfrow = c(1, 1))
```

Out of all observations, only 1 percent of houses are located on the waterfront.

Additionally, the majority of houses (90.2%) have a view score of 0. Among the remaining view scores, 4.5% have a score of 2, while scores of 1 and 4 each account for 1.5%. The remaining 2.4% of houses have a view score of 3. we observed that the 'view' variable predominantly contained 0 values, suggesting that many houses had not been viewed. In response, we decided to engineer a new feature named 'viewed' to capture this information more explicitly. The 'viewed' variable takes on a value of 1 if the house has been viewed and 0 otherwise.

```
# Create a new variable 'viewed' with value 1 if 'view' is not 0, and 0 otherwise
house_data2$viewed <- ifelse(house_data2$view != 0, 1, 0)

# Drop the original 'view' variable
house_data2 <- house_data2[, !names(house_data2) %in% c("view")]

# Categorical Variables
cat_vars <- c("waterfront", "viewed", "condition", "grade")
```

The majority of houses in the dataset are in good to average condition. Approximately 91.7% of houses fall within Condition 3, indicating that a significant portion of the properties is well-maintained.Condition 4 homes represent 26.3%, suggesting a sizable proportion of houses are in better-than-average condition. Meanwhile, Condition 5 homes, which likely denote excellent condition, constitute 7.9% of the dataset.

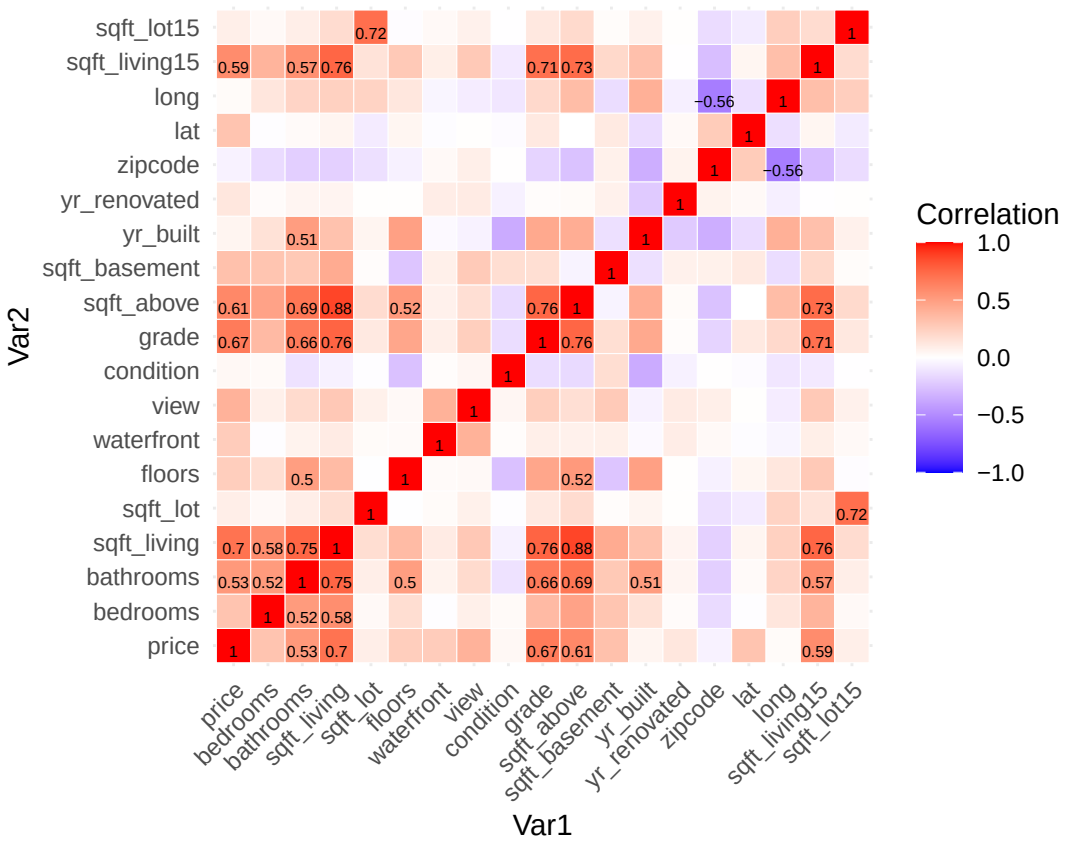
The distribution of grades reflects a diverse range of housing quality. A significant portion of houses falls within Grade 7 (41.6%) and Grade 8 (28.1%), indicating properties with a higher level of construction and design. Grades 9 and 10 together contribute 17.3%, highlighting a considerable proportion of houses with superior construction and design quality.The dataset includes a limited number of houses with lower grades (1-6), with most grades in this range having negligible representation (close to 0%).The distribution is skewed towards higher grades, emphasizing the prevalence of houses with above-average construction and design quality in the dataset.

Handling Zero Values in Bedroom & Bathroom

```
# Function to impute missing values using the median based on non-zero values
impute_nonzero <- function(var) {
  non_zero_values <- as.numeric(var[var != 0])
  if (length(non_zero_values) > 0) {
    imputed_value <- median(non_zero_values)
    var[var == 0] <- imputed_value
  }
  return(var)
}

# Convert the variables to numeric again
house_data2$bedrooms <- as.numeric(as.character(house_data2$bedrooms))
house_data2$bathrooms <- as.numeric(as.character(house_data2$bathrooms))
house_data2$floors <- as.numeric(as.character(house_data2$floors))

# Apply the imputation function to bedrooms and bathrooms
house_data2$bedrooms <- impute_nonzero(house_data2$bedrooms)
house_data2$bathrooms <- impute_nonzero(house_data2$bathrooms)
```

Correlation Analysis

```
# Calculate correlation for all variables
cor_matrix <- cor(all_vars)

# Create a heatmap for correlation values
melted_correlation <- melt(cor_matrix)
ggplot(melted_correlation, aes(x = Var1, y = Var2, fill = value)) +
  geom_tile(color = "white") +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                       midpoint = 0, limit = c(-1, 1), space = "Lab",
                       name = "Correlation") +
  geom_text(aes(label = ifelse(abs(value) > 0.5, round(value, 2), "")), vjust = 1,
    size= 2) + #  filter the results that are highly correlated to interpret easy
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  coord_fixed()
```



Initially, we constructed a correlation matrix to discern relationships among variables in the dataset. To enhance interpretability, we applied a filter, selecting correlations with an absolute value greater than 0.5. This focused approach facilitates easier interpretation by highlighting strong correlated variables. The choice of the cutoff level for correlation analysis depends on the specific goals of the analysis and the nature of the data. Commonly used cutoff values for correlation coefficients between 0.5-0.7 for moderate correlation and above 0.7 for strong correlation. Eventhough we in the begging chooses 0.7 cutoff , the results showed that The price variable stands out with a strong positive correlation of 0.70 with only the square footage of living space (sqft_living). While this suggests a notable linear relationship between these two factors,we recognized it may be beneficial to also consider variables with moderate correlations to price, as they could provide additional insights into determinants of house prices beyond just living space. So we decreased the cutoff level according that.

According to final results (with 0.5 cutoff):

The Price (dependent variable): Strongly correlated with the square footage of living space (sqft_living) at 0.70, indicating that larger living spaces tend to command higher prices. And also having moderate correlation with other features such as bathrooms (0.53), sqft_above(0.61), sqft_living15(0.59) and grade(0.67)

The number of bathroom (bathrooms) have strong correlation with only sqft_living(0.75), also having moderate correlation with multiple variables ; price(0.53), bedrooms(0.52), floors (0.5), sqft_above(0.69),sqft_living15 (0.57), grade (0.66),sqft_living15 (0.51)

The square footage of living space (sqft_living) demonstrates strong positive correlations with price (0.70), bathrooms (0.75), sqft_above (0.88), sqft_living15 (0.76), and grade (0.76), highlighting its multifaceted influence on house features and value. and it have moderate correlation with bedroom (0.58)

The square footage above ground (sqft_above) has the highest correlation with sqft_living (0.88) and substantial correlations with sqft_living15 (0.73) and grade (0.76), And it have moderate correlation with price (0.61), bathroom (0.69) and floors (0.52) underscoring its significance in determining overall property grades.

The overall grade (grade) exhibits a strong positive correlation with various measures of house size, including sqft_living (0.76), sqft_above (0.76), and sqft_living15 (0.71). Additionally, it shows a moderate correlation with price (0.67) and bathrooms (0.66), suggesting that houses with higher grades tend to be larger, have more bathrooms, and command higher prices.

In conclusion, the correlation analysis has uncovered intricate relationships among various features in the dataset, emphasizing the strong correlation of house prices with the square footage of living space (sqft_living). Additionally, moderate correlations with other features such as bathrooms (0.53), sqft_above (0.61), sqft_living15 (0.59), and grade (0.67) suggest the presence of diverse factors influencing property values, warranting further in-depth analysis in later stages.

Moreover, the identified potential multicollinearity issue highlights the need for careful feature selection to enhance the stability and interpretability of the regression model. Specifically, considering the strong correlations among Sqft_living, Sqft_living15, and Sqft_above, it is advisable to include only one of them in the model to avoid multicollinearity and ensure the model's robustness.

In addition it's crucial to remember that correlation does not imply causation. While these variables are correlated, further analysis and domain knowledge are needed to understand the causal relationships and make informed predictions.

As we move forward, advanced visualizations will serve as valuable tools to unravel these complex relationships, offering a more nuanced understanding of the dynamics shaping the real estate market in King County, Washington State, USA.

Feature Selection: Given the multicollinearity observed among sqft_living, sqft_living15, and sqft_above, we select one of these variables that best represents the living space in the model. For our case, sqft_living has a strong correlation with the target variable price and other predictors, making it a suitable choice.

Advanced Visualization Techniques

Scatter plots

```
# Create an empty list to store plots
plots_list <- list()

# Iterate through variables and create scatter plots
for (variable in cont_vars[-1]) {
  # Create scatter plot with regression line
  scatter_plot <- ggplot(house_data2, aes_string(x = variable, y = "price")) +
    geom_point(color = "orange") +
    geom_smooth(method = "lm", se = FALSE, color = "red") +
    geom_encircle(data = house_data2 %>% filter(price > 6000000),
                  color = "red", size = 2, expand = 0.05) +
     geom_encircle(data = house_data2 %>% filter(bedrooms == 33),
                  color = "green", size = 2, expand = 0.05) +
    labs(title = paste(variable, "vs. Price"), x = variable, y = "Price") +
    theme_minimal()

  # Add the plot to the list
  plots_list[[variable]] <- scatter_plot
}
```

Continuous variables vs. "Price"

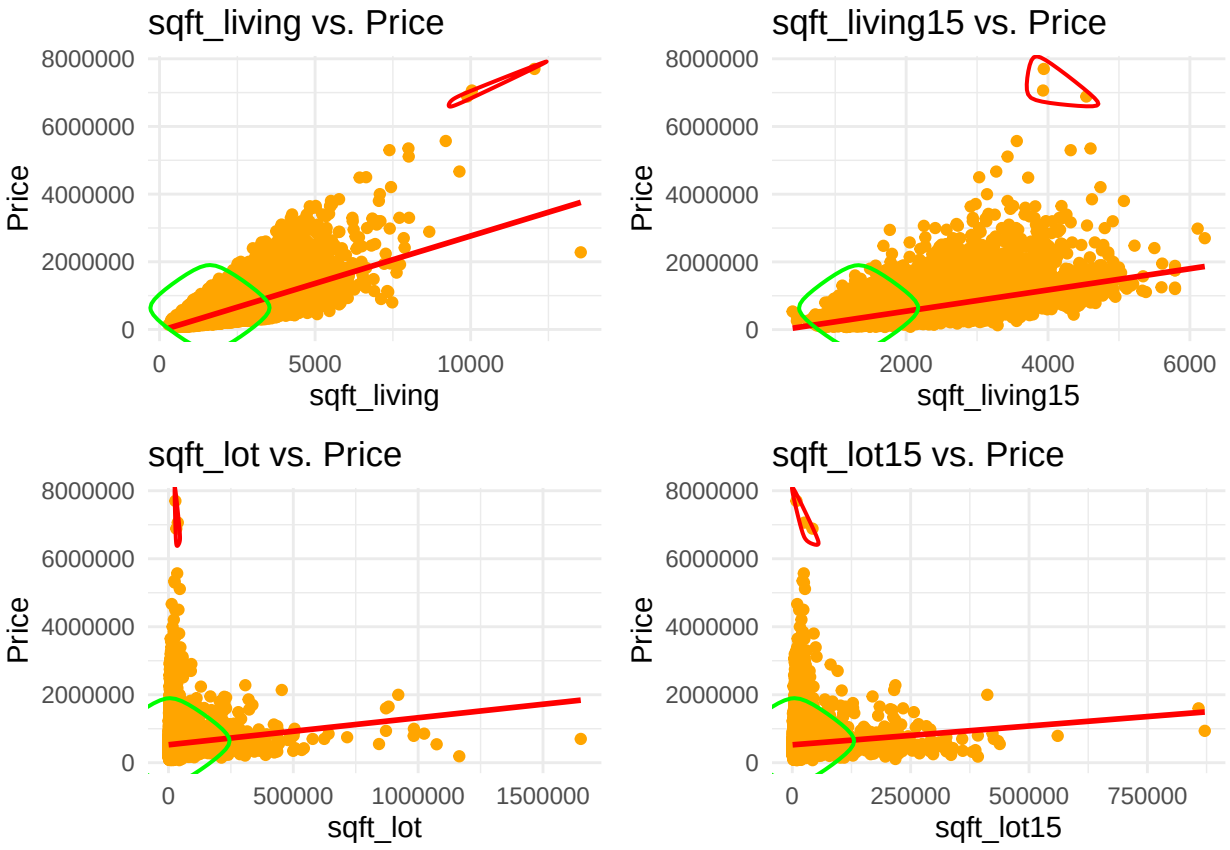
```
## Warning: ‘aes_string()‘ was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with ‘aes()‘.
## i See also ‘vignette("ggplot2-in-packages")‘ for more information.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this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

```
# Arrange the plots in a grid
advanced_plots <- ggarrange(plotlist = plots_list, ncol = 2, nrow = 2)
```

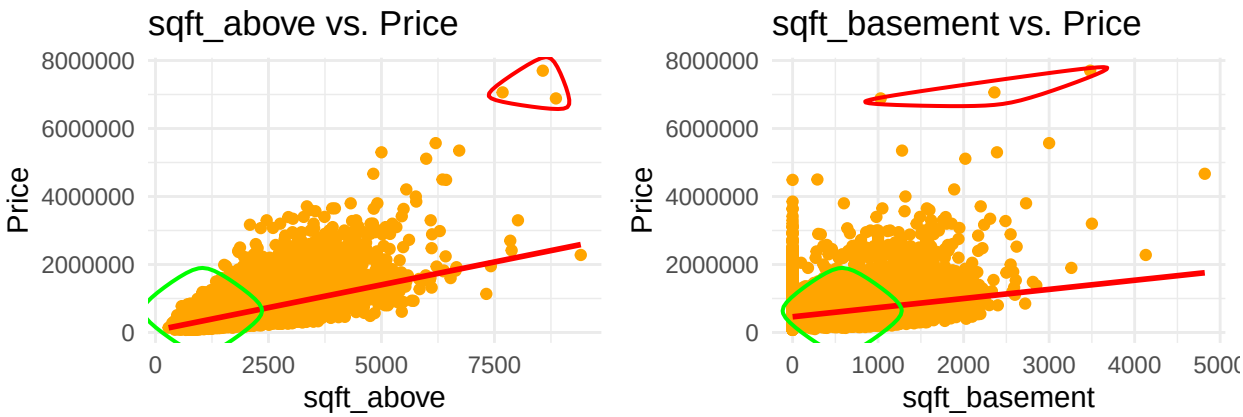
```
## ‘geom_smooth()‘ using formula = ’y ~ x’
## ‘geom_smooth()‘ using formula = ’y ~ x’
## ‘geom_smooth()‘ using formula = ’y ~ x’
## ‘geom_smooth()‘ using formula = ’y ~ x’
## ‘geom_smooth()‘ using formula = ’y ~ x’
## ‘geom_smooth()‘ using formula = ’y ~ x’
```

```
# Display the arranged plots
print(advanced_plots)
```

```
## $‘1‘
```




```
##
## $‘2‘
```




```
##
## attr(,"class")
## [1] "list"      "ggarrange"
```

We generate scatter plots for various continuous variables against housing prices, utilizing light green points and red regression lines for visualization. A red circle is incorporated to emphasize observations where housing prices exceed $6,000,000, indicating potential outliers. The scatter plots collectively underscore similarities in the distribution patterns of all continuous variables concerning price. This graphical exploration enhances the understanding of the correlation between each continuous variable and housing prices, with the filter for prices drawing attention to three potential outliers. Identifying and comprehending such outliers is crucial for robust data analysis, aiding in informed decisions regarding their impact on statistical models and subsequent analyses. Further investigation and domain knowledge are typically required to interpret these outliers within the dataset's context.

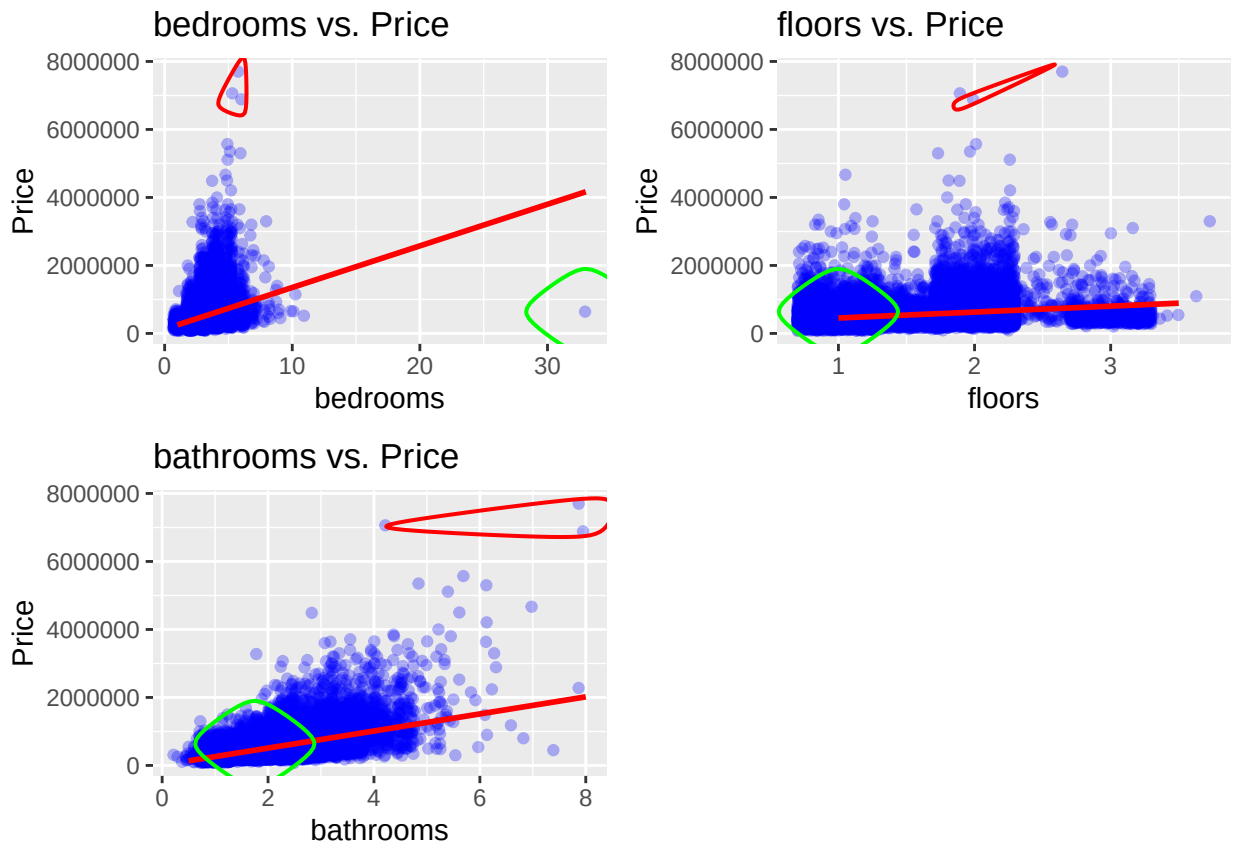
```
# Create an empty list to store plots
plots_list <- list()

# Iterate through variables and create scatter plots
for (variable in disc_vars) {
  # Create scatter plot with regression line
  scatter_plot <- ggplot(house_data2, aes_string(x = variable, y = "price")) +
    geom_jitter(width = .3, alpha = .3, color = "blue") + # Introduce a noise
```

```
    geom_smooth(method = "lm", se = FALSE, color = "red") +
    geom_encircle(data = house_data2 %>% filter(price > 6000000),
     color = "red", size = 2, expand = 0.05) + # Add an encircling  for high prices
    geom_encircle(data = house_data2 %>% filter(bedrooms == 33),
                  color = "green", size = 2, expand = 0.05) + # Add an  encirling for
    #bedrooms == 33
    labs(title = paste(variable, "vs. Price"), x = variable, y = "Price")

  # Add the plot to the list
  plots_list[[variable]] <- scatter_plot
}

# Arrange the plots in a grid
advanced_plots <- ggarrange(plotlist = plots_list, ncol = 2, nrow = 2)
```

Discreate variables vs "Price"

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

```
# Display the arranged plots
print(advanced_plots)
```



In this code update, we continued our exploration of the dataset by examining the relationship between

housing prices and discrete variables. We introduced noise for a more nuanced view and identified high-priced outliers, visualizing them with encircling shapes.

As an additional step, we focused on the unusual case where the number of bedrooms equals 33. We specifically circled these observations using a distinctive green color. This targeted analysis aims to spotlight and investigate unique patterns and outliers within the data, enhancing our understanding of their impact on housing prices. Our approach reflects an iterative process, adapting visualizations to reveal hidden insights in the dataset.

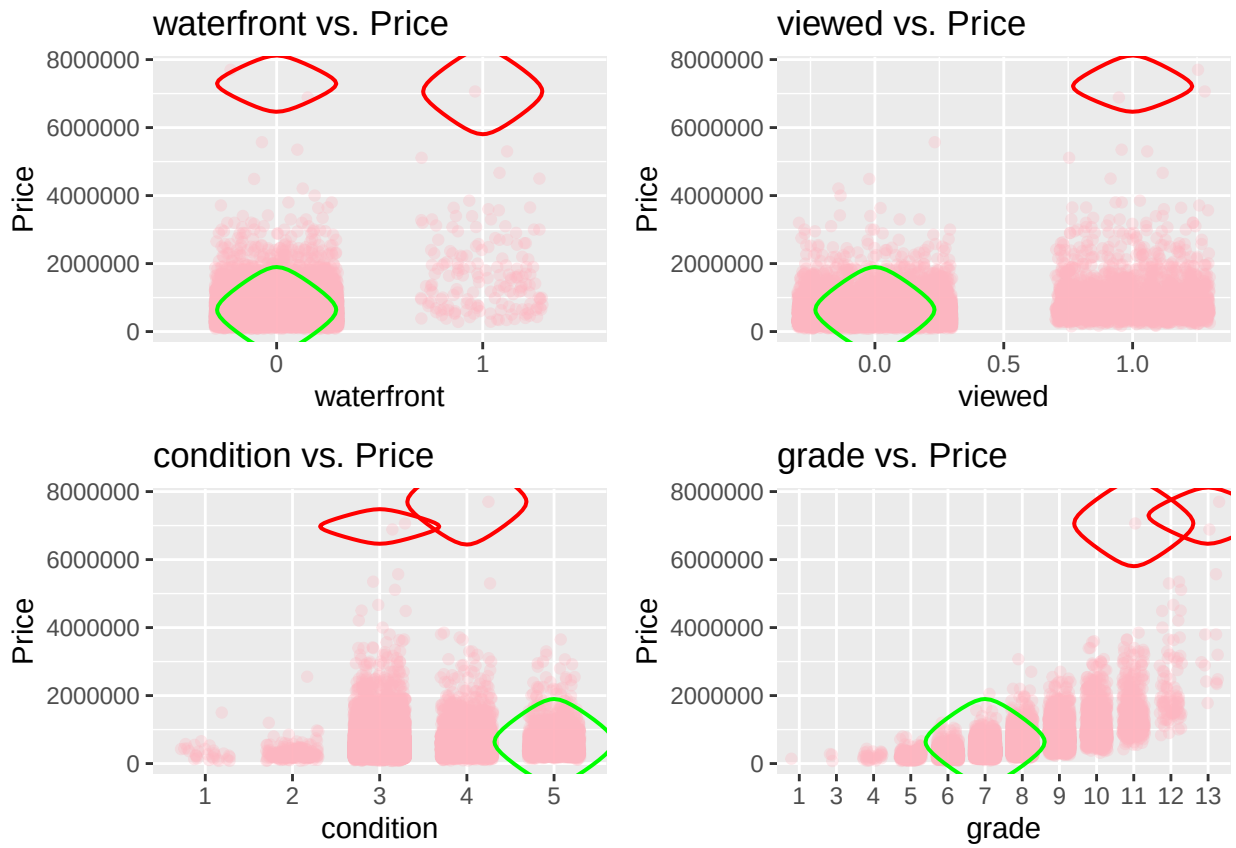
```
# Create an empty list to store plots
plots_list <- list()

# Iterate through categorical variables and create scatter plots
for (variable in cat_vars) {
  # Create scatter plot with regression line, colored points, jitter for density, and
  ##circle for high-priced outliers
  scatter_plot <- ggplot(house_data2, aes_string(x = variable, y = "price")) +
    geom_jitter(width = .3, alpha = .3, color= "lightpink") +
    geom_encircle(data = house_data2 %>% filter(price > 6000000),
                  color = "red", size = 2, expand = 0.05) +
    geom_encircle(data = house_data2 %>% filter(bedrooms == 33),
                  color = "green", size = 2, expand = 0.05) +
    geom_encircle(data = house_data2 %>% filter(grade < 4),
                  color = "blue", size = 2, expand = 0.05) +
    labs(title = paste(variable, "vs. Price"), x = variable, y = "Price")

  # Add the plot to the list
  plots_list[[variable]] <- scatter_plot
}
```

Categorical variables vs. "Price"

```
## Warning: There was 1 warning in ‘filter()‘.
## i In argument: ‘grade < 4‘.
## Caused by warning in ‘Ops.factor()‘:
## ! ’<’ not meaningful for factors
## There was 1 warning in ‘filter()‘.
## i In argument: ‘grade < 4‘.
## Caused by warning in ‘Ops.factor()‘:
## ! ’<’ not meaningful for factors
## There was 1 warning in ‘filter()‘.
## i In argument: ‘grade < 4‘.
## Caused by warning in ‘Ops.factor()‘:
## ! ’<’ not meaningful for factors
## There was 1 warning in ‘filter()‘.
## i In argument: ‘grade < 4‘.
## Caused by warning in ‘Ops.factor()‘:
## ! ’<’ not meaningful for factors
```

```
# Arrange the plots in a grid
advanced_plots <- ggarrange(plotlist = plots_list, ncol = 2, nrow = 2)
```

```
# Display the arranged plots
print(advanced_plots)
```



In this visualization, we explored various categorical variables in relation to housing prices. Each scatter plot includes a regression line, light pink points with added jitter for better density visualization, and encircling of specific observations. The red circles highlight houses with prices exceeding $6,000,000, signaling potential outliers in the dataset. Additionally, green circles indicate properties with an unusually high number of 33 bedrooms, drawing attention to this unique characteristic. Moreover, blue circles represent homes with a grade lower than 3, suggesting those with the lowest grading. The use of color-coded encircling helps emphasize distinct patterns and potential anomalies in the relationships between categorical variables and housing prices.

The blue circles in the scatter plots indicate houses with the lowest grades (grade < 3). these house is not waterfront , have zero view, and is in a poor condition (condition 1).

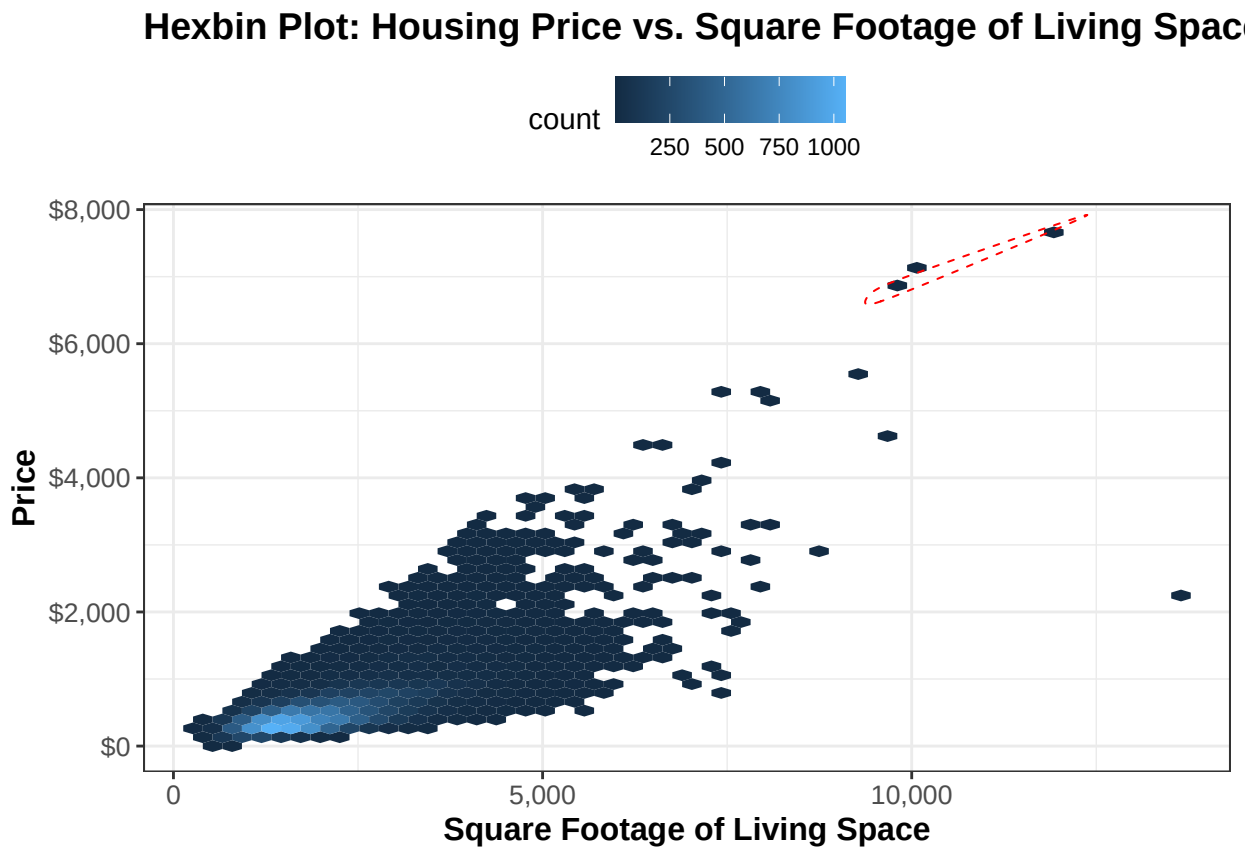
The green-circled houses with 33 bedrooms present intriguing attributes, notably lacking waterfront features, having zero views, a condition rating of 5, and a grade higher than 5. While such characteristics are conceivable, the observed data challenges expectations, particularly in terms of the square footage of living space. The discrepancy between the expected and actual living space raises questions about potential anomalies or recording errors. To refine the accuracy and reliability of the analysis, reassessing or potentially excluding these variables is advisable. Upon detailed examination, anomalies in houses with 33 bedrooms, such as a single floor, less than 2000 sqft_living, and around 2 bathrooms, were identified as potential errors. To rectify this issue, the number of bedrooms was replaced with the median value, resulting in a more reasonable representation aligned with domain knowledge and realistic expectations.

```r
# Find indices where 'bedrooms' is 33
index_bedrooms_33 <- which(house_data2$bedrooms == 33)


# Replace the 'bedrooms' value of 33 with the median value of bedrooms
house_data2$bedrooms[index_bedrooms_33] <- median(house_data2$bedrooms, na.rm = TRUE)
```

Hexbin Visualization: Housing Prices and Subset Encircling

```r
# Other advanced visualizations for analysis (e.g., hexbin, density plot, etc.)
p<- ggplot(house_data2, aes(x = sqft_living, y = price)) +
  geom_hex(bins = 50) +
  labs(title = 'Hexbin Plot: Housing Price vs. Square Footage of Living Space',
       x = 'Square Footage of Living Space', y = 'Price')  +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Encircle specific data points

# Filter data to select specific data points (sqft_living > 5000 and price > 8000)
house_sel <- house_data2 %>%
  filter( price > 6000000)

# Add encircling around selected data points with improved aesthetics
p +
  geom_encircle(data = house_sel, color = "red", size =  1, expand = 0.05,
                linetype ="dashed") +
    geom_encircle(data = house_data2 %>% filter(bedrooms == 33),
                  color = "green", size = 2, expand = 0.05) +
    geom_encircle(data = house_data2 %>% filter(grade < 3),
                  color = "blue", size = 2, expand = 0.05) +
  theme_bw() +
  scale_x_continuous(labels = scales::comma) +
  scale_y_continuous(labels = scales::dollar_format(scale = 0.001)) +
  theme(axis.text = element_text(size = 10),
        axis.title = element_text(size = 12, face = "bold"),
        plot.title = element_text(size = 14, face = "bold"),
        legend.position = 'top')
```

```
## Warning: There was 1 warning in `filter()`.
## i In argument: `grade < 3`.
## Caused by warning in `Ops.factor()`:
## ! '<' not meaningful for factors
```



In this advanced visualization, we employed a hexbin plot to explore the relationship between housing price and square footage of living space. The hexbin plot provides a clear overview of the density of data points, offering insights into the distribution of housing prices concerning living space. We continued the analysis by encircling specific data points of interest. The red dashed circle encompasses houses with prices exceeding $6,000,000. Additionally, we maintained the encircling of the green circle around properties with 33 bedrooms and the blue circle around those with a grade less than 3. These encirclings help to highlight and differentiate distinct subsets within the dataset, providing a nuanced understanding of the data distribution. The refined aesthetics, including color-coded circles and improved line types, enhance the visual appeal and interpretability of the plot. This visualization strategy builds upon the previous stages, offering a comprehensive exploration of the dataset's intricate patterns and outliers.

```
house_data2[duplicated(house_data2), ]
```

Checking Dublicates

```
##  [1] id            date          price         bedrooms      bathrooms
##  [6] sqft_living   sqft_lot      floors        waterfront    condition
## [11] grade         sqft_above    sqft_basement yr_built      yr_renovated
## [16] zipcode       lat           long          sqft_living15 sqft_lot15
## [21] viewed
## <0 rows> (or 0-length row.names)
```

```r
duplicates <- house_data2[duplicated(house_data2$id), ]
dim(duplicates)
```

```
## [1] 177  21
```

The dataset analysis provided two distinct findings regarding duplicates. First, a comprehensive scan across all columns of the dataset did not reveal any duplicate entries, suggesting the entire dataset is unique in its entirety. However, when focusing specifically on the 'id' column—a unique identifier for each home sold—it was discovered that 177 homes were listed with duplicate 'id' values. This suggests that while the dataset itself is unique, there were instances where individual homes appeared to have been sold more than once during the observed period. Such duplicates in the 'id' column indicate potential anomalies in the data, suggesting that certain homes may have been recorded multiple times or sold more than once, warranting a closer examination into the sales records to ensure data accuracy and integrity.

To handle it 3 approaches :

Remove Duplicates: You can choose to remove the rows with duplicate 'id' values. This ensures that each home is represented only once in the dataset. However, you need to carefully consider the implications of removing data, as it may result in a loss of information.

```r
# Remove rows with duplicate 'id' values
house_data2_unique <- house_data2[!duplicated(house_data2$id), ]
```

Aggregate Data: If the duplicates in the 'id' column represent different transactions or sales for the same home, you might want to aggregate the data. For example, you could calculate the average price, total number of bedrooms, or other relevant statistics for each unique 'id'.

```r
# Aggregate data by 'id'
house_data2_agg<- house_data2 %>%
  group_by(id) %>%
  summarize(avg_price = mean(price),
            total_bedrooms = sum(bedrooms),
            # Add other relevant aggregations
            )
```

Feature Engineering: Instead of directly removing or aggregating duplicates, you can create new features to capture the information. For example, you might create a new binary feature indicating whether a home has been sold more than once.

```r
# Create a binary feature indicating if 'id' has duplicates
house_data2$has_duplicates <- duplicated(house_data2$id)
```

Geospatial Visualization

We conducted a geospatial analysis to visually represent the distribution of house prices within King County and Seattle. The aim was to identify regional patterns and highlight areas with notable property values.

Data Acquisition and Integration To initiate this exploration, we obtained the shapefile for King County from the website [link: https://gis-kingcounty.opendata.arcgis.com/]. Subsequently, we merged this shapefile with our existing dataset, house_data2. This integrated dataset serves as the foundation for our geospatial visualization.

```r
shape<-read_sf("zipcodeSHP/")
head(shape, 3)
```

```
## Simple feature collection with 3 features and 8 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: 1266411.88 ymin: 97164.5685 xmax: 1303261.54 ymax: 134079.634
## Projected CRS: NAD83(HARN) / Washington North (ftUS)
## # A tibble: 3 x 9
##     ZIP ZIPCODE COUNTY ZIP_TYPE COUNTY_NAM  PREFERRED_  Shape_STAr Shape_STLe
##   <dbl> <chr>   <chr>  <chr>    <chr>       <chr>            <dbl>      <dbl>
## 1 98001 98001   033    Standard King County AUBURN      525368923.    147537.
## 2 98002 98002   033    Standard King County AUBURN      205302741     104440.
## 3 98003 98003   033    Standard King County FEDERAL WAY 316942614.    123734.
## # i 1 more variable: geometry <MULTIPOLYGON [US_survey_foot]>
```

```r
merged_data <- merge(house_data2, shape, by.x = "zipcode", by.y = "ZIPCODE",
                     all.x = TRUE)
```

```r
str(merged_data)
```

```
## 'data.frame':    23307 obs. of  30 variables:
##  $ zipcode       : int  98001 98001 98001 98001 98001 98001 98001 98001 98001 98001 ...
##  $ id            : num  6699300330 3750605247 3353401710 2005950050 3522049063 ...
##  $ date          : Date, format: "2015-05-13" "2014-08-04" ...
##  $ price         : num  372000 255000 227950 260000 639900 ...
##  $ bedrooms      : num  5 3 3 3 4 3 5 3 2 3 ...
##  $ bathrooms     : num  2.5 1 1.5 2 2.5 1 2.5 1 1 1.75 ...
##  $ sqft_living   : int  2840 1710 1670 1630 3380 1370 3597 1540 1780 1840 ...
##  $ sqft_lot      : int  6010 12000 8230 8018 75794 10708 4972 37950 81021 16679 ...
##  $ floors        : num  2 1 1 1 2 1 2 1 1 1 ...
##  $ waterfront    : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ condition     : Factor w/ 5 levels "1","2","3","4",..: 3 4 5 3 3 3 3 4 4 3 ...
##  $ grade         : Factor w/ 12 levels "1","3","4","5",..: 7 6 6 6 9 6 6 6 8 7 ...
##  $ sqft_above    : int  2840 1710 1670 1630 3380 1370 3597 1090 1780 1840 ...
##  $ sqft_basement : int  0 0 0 0 0 0 0 450 0 0 ...
##  $ yr_built      : int  2003 1972 1954 2003 1997 1969 2006 1959 1954 1989 ...
##  $ yr_renovated  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ lat           : num  47.3 47.3 47.3 47.3 47.4 ...
##  $ long          : num  -122 -122 -122 -122 -122 ...
##  $ sqft_living15 : int  2740 1310 2077 1610 3710 1770 3193 1820 1780 1910 ...
##  $ sqft_lot15    : int  5509 9600 4910 8397 17913 14482 6000 24375 26723 15571 ...
##  $ viewed        : num  0 0 0 0 0 0 0 0 1 0 ...
##  $ has_duplicates: logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ ZIP           : num  98001 98001 98001 98001 98001 ...
##  $ COUNTY        : chr  "033" "033" "033" "033" ...
##  $ ZIP_TYPE      : chr  "Standard" "Standard" "Standard" "Standard" ...
##  $ COUNTY_NAM    : chr  "King County" "King County" "King County" "King County" ...
##  $ PREFERRED_    : chr  "AUBURN" "AUBURN" "AUBURN" "AUBURN" ...
##  $ Shape_STAr    : num  525368923 525368923 525368923 525368923 525368923 ...
##  $ Shape_STLe    : num  147537 147537 147537 147537 147537 ...
##  $ geometry      :sfc_MULTIPOLYGON of length 23307; first list element: List of 1
```

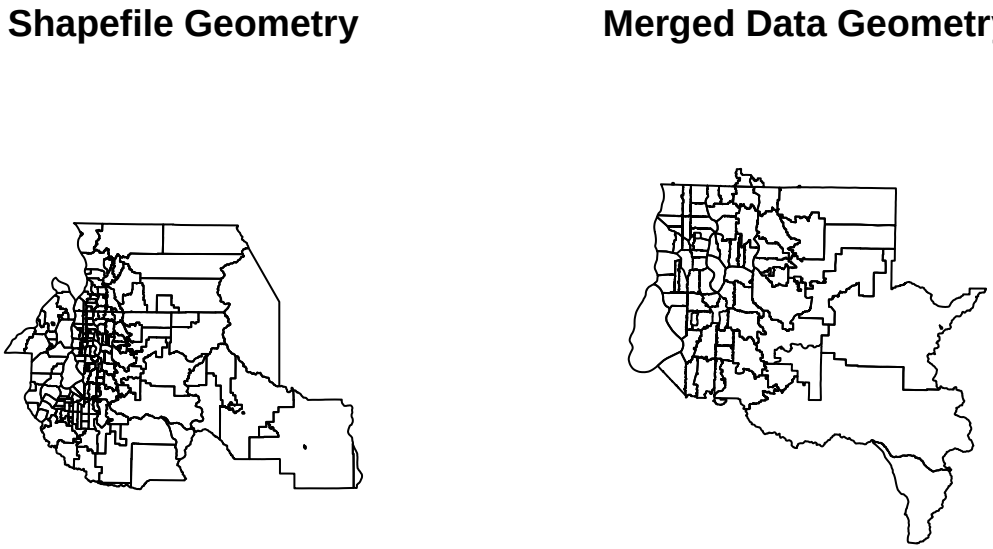
```
##   ..$ :List of 1
##   .. ..$ : num [1:1563, 1:2] 1279285 1279733 1280459 1281079 1281153 ...
##   ..- attr(*, "class")= chr [1:3] "XY" "MULTIPOLYGON" "sfg"
```

```
house_data2_avg <- house_data2 %>%
  group_by(zipcode) %>%
  summarise_all(mean, na.rm = TRUE)
```

```
## Warning: There were 210 warnings in `summarise()`.
## The first warning was:
## i In argument: `waterfront = (function (x, ...) ...`.
## i In group 1: `zipcode = 98001`.
## Caused by warning in `mean.default()`:
## ! argument is not numeric or logical: returning NA
## i Run `dplyr::last_dplyr_warnings()` to see the 209 remaining warnings.
```

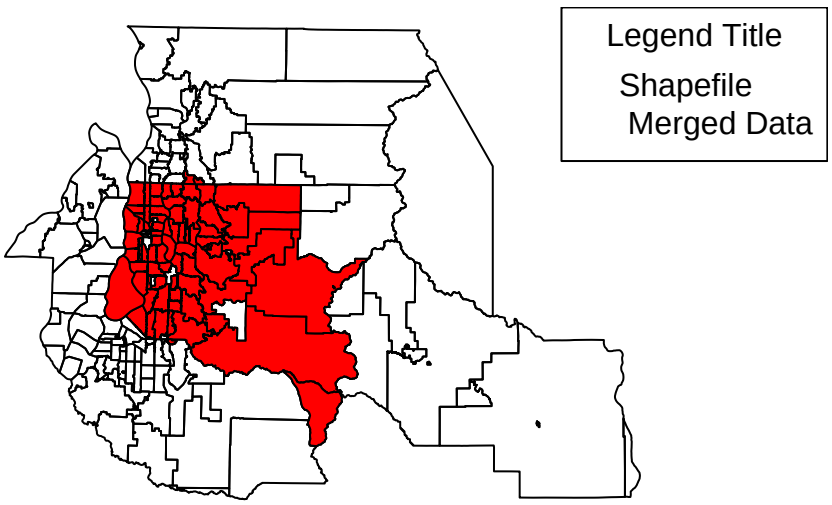
```
# Merge the averaged datasets
merged_data_avg <- merge(house_data2_avg, shape, by.x = "zipcode",
                         by.y = "ZIPCODE", all.x = TRUE)
```

```


# Set up a 1x2 plotting layout
par(mfrow = c(1, 2))
# Plot the geometry of the shapefile
plot(shape$geometry, main = "Shapefile Geometry")
# Plot the geometry of the merged dataset
plot(merged_data_avg$geometry, main = "Merged Data Geometry")
```



In the first set of plots, we visualize the original shapefile geometry in light blue and the merged dataset geometry in red. This comparison allows us to observe the integration of the two datasets.

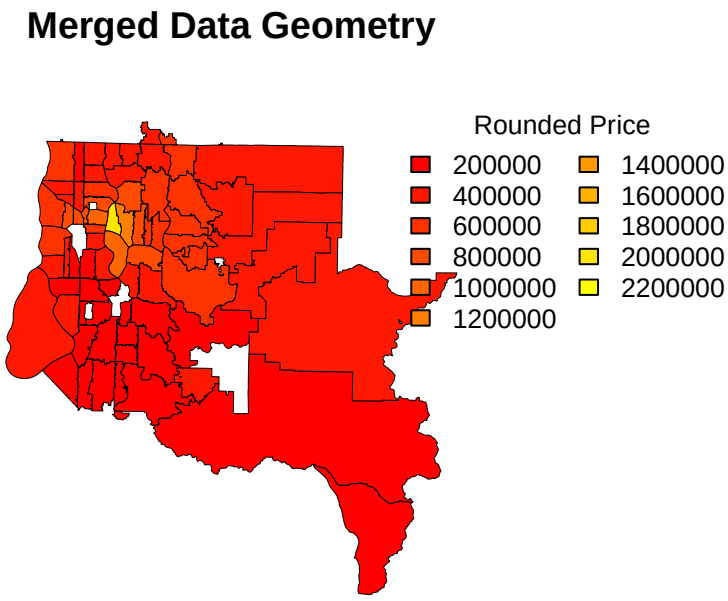
```
# Set up a 1x2 plotting layout
par(mfrow = c(1, 1))

# Plot the geometry of the shapefile with one color (e.g., black)
plot(shape$geometry, main = "Shapefile Geometry")


# Plot only the subset within the shapefile geometry with a different color (e.g., red)
plot(merged_data_avg$geometry, col = "red", add = TRUE)

# Add a legend
legend("topright", legend = c("Shapefile", " Merged Data"),
       col = c("lightblue", "red"), title = "Legend Title")
```



In the second set of plots, we refine the visualization by representing the original shapefile geometry in black and highlighting the subset within the merged dataset in red. The legend aids in distinguishing the components of the plot.

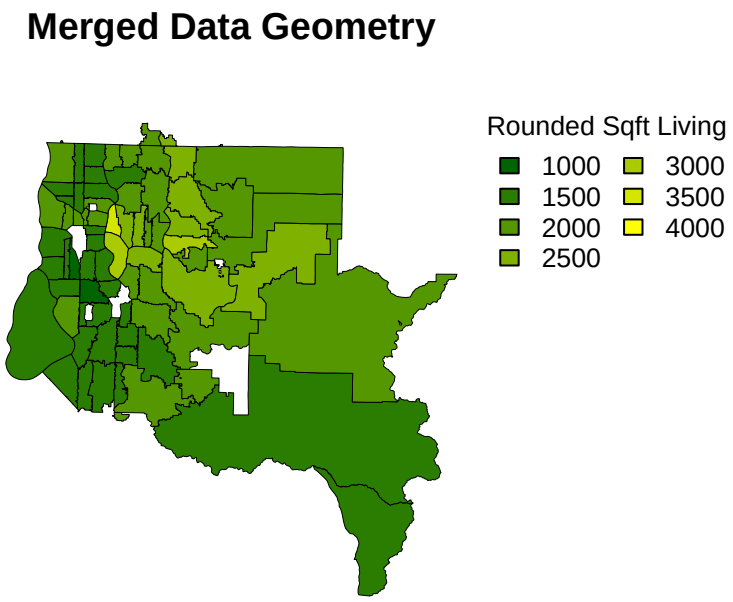
```
# Generate a continuous blue color palette
custom_palette <- colorRampPalette(c("red", "yellow"))

# Round the 'price' variable
merged_data_avg$rounded_price <- round(merged_data_avg$price)

# Determine the number of breaks for the legend
num_breaks <- 11

# Generate a scaled sequence of rounded prices
scaled_prices <- pretty(merged_data_avg$rounded_price, n = num_breaks)

# Plot the geometry with custom colors based on the rounded 'price' variable
plot(
  merged_data_avg$geometry,
  main = "Merged Data Geometry",
  col = custom_palette(num_breaks)[cut(merged_data_avg$rounded_price,
                                       breaks =scaled_prices)],
  border = "black",  # Add black borders for better visualization
  lwd = 0.2          # Adjust the line width of borders
)
```

```
# Add legend with a scaled sequence of rounded prices
legend(
  "topright",
  legend = scaled_prices,
  fill = custom_palette(num_breaks),
  title = "Rounded Price",
  cex = 0.8,
  bty = "n",          # Remove box around the legend
  ncol = 2            # Set the number of columns in the legend
)
```




```
# Generate a continuous blue color palette
custom_palette_sqft <- colorRampPalette(c("darkgreen", "yellow"))

# Round the 'sqft_living' variable
merged_data_avg$rounded_sqft_living <- round(merged_data_avg$sqft_living)

# Determine the number of breaks for the legend
num_breaks_sqft <- 7

# Generate a scaled sequence of rounded sqft_living values
scaled_sqft_living <- pretty(merged_data_avg$rounded_sqft_living, n = num_breaks_sqft)

# Plot the geometry with custom colors based on the rounded 'sqft_living' variable
plot(
```

```
  merged_data_avg$geometry,
  main = "Merged Data Geometry",
  col = custom_palette_sqft(num_breaks_sqft)[cut(merged_data_avg$rounded_sqft_living,
                                                 breaks = scaled_sqft_living)],
  border = "black",  # Add black borders for better visualization
  lwd = 0.2          # Adjust the line width of borders
)

# Add legend with a scaled sequence of rounded sqft_living values
legend(
  "topright",
  legend = scaled_sqft_living,
  fill = custom_palette_sqft(num_breaks_sqft),
  title = "Rounded Sqft Living",
  cex = 0.8,
  bty = "n",         # Remove box around the legend
  ncol = 2           # Set the number of columns in the legend
)
```



Finally, we employ a custom color palette to map the average price distribution within the merged data, providing insights into the spatial variations in property values. The legend assists in interpreting the color-coded price ranges. We observed that centered areas have higher prices.

Conclusion

In conclusion, the exploration of housing dynamics in King County, WA, reveals a dataset rich in diverse variables that influence property prices. Through rigorous data preprocessing, including addressing missing values and outliers, and transforming variables for robust analysis, we gained valuable insights into the relationships among key features. Correlation analysis highlighted the strong positive correlation between house prices and factors such as square footage of living space, bathrooms, and overall grade. Visualizations, ranging from scatter plots to hexbin plots and geospatial analyses, provided nuanced perspectives on the dataset, emphasizing potential outliers and unique patterns. Noteworthy findings include high-priced outliers, a peculiar property with 33 bedrooms, and homes with the lowest grades exhibiting distinct characteristics. The geospatial visualization further unveiled regional patterns in property values across King County. Moving forward, these insights lay the groundwork for more sophisticated modeling and predictive analyses, with a keen awareness of potential multicollinearity and the need for careful feature selection. The iterative nature of the analysis underscores the importance of continuously refining our understanding to uncover hidden dynamics and anomalies within the real estate market.

Machine Learning

##Resources

R Studio and associated libraries

Kaggle dataset (provided under CC0: Public Domain). https://www.kaggle.com/datasets/shivachandel/kc-house-data/data

Shapefile for mapping https://gis-kingcounty.opendata.arcgis.com/