NAME- Siddharth Kr Jaiswal

Student id- 40425082

Subject - Statistics for Business

Subject code - MGT7177

Table of content-

Introduction

The purpose of this task is to do an in-depth statistical study to anticipate the selling prices of the properties in the Ames House Price dataset. The experiment's purpose was to employ linear regression models to see how well our selected factors predicted the selling values of homes in Ames, Iowa. This collection of data contains 2880 observations and 78 characteristics on residential property sales in Ames from 2006 to 2010. As with most data analysis tools, the term "environment" refers to a completely structured and integrated system rather than an incremental accumulation of extremely particular and inflexible instruments. R is an excellent framework for developing unique dynamic statistical approaches. It evolved rapidly and has been supplemented by an abundance of packages. Still, a great deal of R programs are basically temporary, with a particular data processing objective in mind. (W. N. Venables et al., 2023).

In that database, location and size are found to have a considerable influence on costs. Both machine learning and retrospective analysis are built on the previous function of house price forecasting (Mire et al., 2022). According to the model (Miller et al., 2020), an increase in living room space corresponds to an increase in median sale price. A survey of multiple papers found that scholars utilize some traits in their work to forecast housing values. These features are classified into four categories: structural, neighborhood, locational, and economic. The structural qualities include the number of sleeping areas, baths, floor space, garage, porch, age of the home, and lot size (Zulkifley et al., 2020).

Utilize visual aids and transformation to conduct a methodical investigation of your data; statisticians refer to this process as exploratory data analysis or EDA for short. EDA is a cycle of iterations. We can a- make inquiries concerning your data, b- Visualize, transform, and model our data to find the answers, and c- make use of the knowledge we gain to improve your inquiries and/or come up with new ones. EDA is not a formal procedure with rigid guidelines. This is more than anything a mental state. You should feel free to investigate every idea that comes to you in the early stages of EDA. There's a chance that some of these concepts will work and others won't. We will focus on a few especially fruitful areas, which you will eventually document and share with others (O'Reilly, 2017).

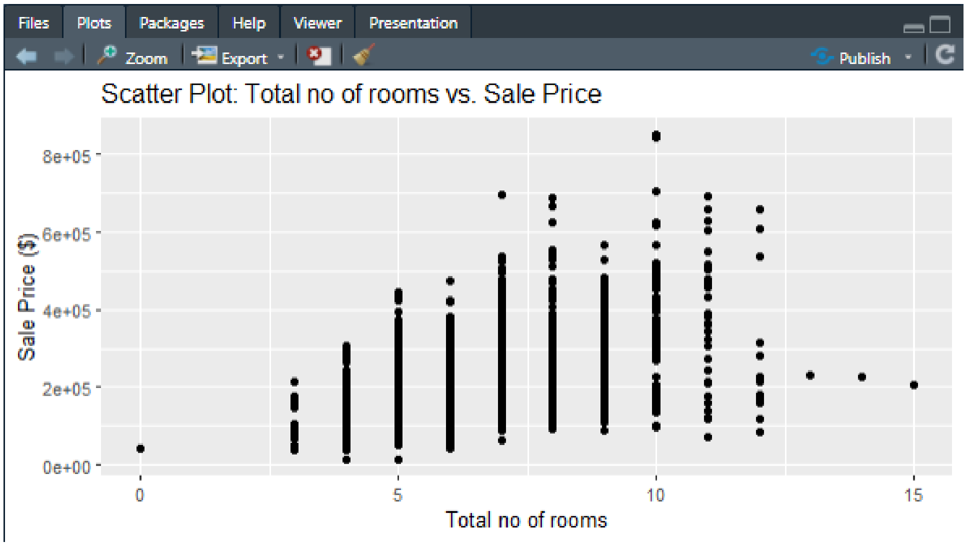
Methodology

The script starts by getting the current working directory (getwd()) and then reading an Excel file named "stats_as.xlsx" via the readxl library using read_excel(). The stats_as variable has this dataset attributed to it. The program creates the data_chosen data frame by choosing particular rows of data that seem to indicate an incorrect or an undefined challenge. Then, it makes two calls to na.omit(), probably to eliminate any rows in data_chosen that have missing values after that the'sale_price' distribution is visualized through the creation of a boxplot by the code. Using the boxplot function, it finds outliers and stores them in the outlier's variable. It then eliminates these anomalies from the dataset named data_chosen. To guarantee accurate representation in the dataset, a few columns—"neighborhood," "configuration," and "slope"—are transformed into categorical factors using as.factor(). The code uses the duplicated() function to remove duplicated rows from the data_chosen data frame while retaining unique rows. By transforming, the code "data_chosen[!duplicated(data_chosen)" and "as.factor" segment makes sure that particular columns—"neighbourhood," "configuration," and "slope"—are handled as categorical variables. In order to preserve unique observations within the dataset, duplicate rows are removed. Lastly, it provides an overview of the cleaned dataset, shedding light on the features and distribution of the variables following the cleaning processes. Exploratory Data Analysis (EDA): Overview statistics are created to offer a snapshot of the cleaned data in order to acquire a better knowledge of the dataset. Visualization methods such as scatter plots, box plots, and bar charts are used by ggplot2(Gharehchopogh *et al*, 2013). These visualizations investigate the links between several features such as 'rooms_total', 'bedroom', 'house_quality', and so on, in relation to 'sale_price'. Model Construction and Evaluation: To create predictive models, The dataset is separated across two parts: training and testing. Two regression models are developed: 'Model_A' concentrating on 'house_quality' and'multiple_model' considering multiple predictors such as 'neighbourhood', 'full_bath', and so on. Prediction methods such as postResample() are used to assess the accuracy of these models. The accuracy of the model is evaluated using metrics like as RMSE, VIF, Cook's Distance, and more. To understand variable relationships, correlation matrices are produced. Final Data Cleaning and Visualization: In the final data cleaning section, duplicates are addressed, and no missing values are present. Scatter plots are used to refine the link between 'house_quality' and'sale_price' in visualization updates (Woźniak *et al*, 2014).

FINDINGS AND RESULTS -

Visualizations:

1. **Scattered plot** - The overall amount of Rooms vs. Price of Sale Scatter Plot: Shows the association between the total number of rooms and sale prices.(Nielson *et al*, 2002)



This R code used ggplot2 to generate a scatter plot that shows how property sale prices connect with the total number of rooms (rooms_tot).

```
# Visualization 1: Scatter Plot - Total no of rooms vs. Sale Price
ggplot(stats_as, aes(x = rooms_tot, y = sale_price)) +
  geom_point() +
  labs(title = "Scatter Plot: Total no of rooms vs. Sale Price",
       x = "Total no of rooms",
       y = "Sale Price ($)")
```

Using ggplot2, this R code creates a scatter plot illustrating the association between property size (in rooms) and sale prices. Every dotted line depicts a property, with the number of rooms (x-axis) mapped against the asking price (y-axis). The graphic allows users to investigate how property values vary by room count, revealing possible trends such as higher costs for larger properties and any outliers or patterns that warrant further examination (Sainani *et al*, 2016)

2. **Boxplot** - Number of Bedrooms vs. Sale Price: Displays the price dispersion based on the amount of bedrooms.

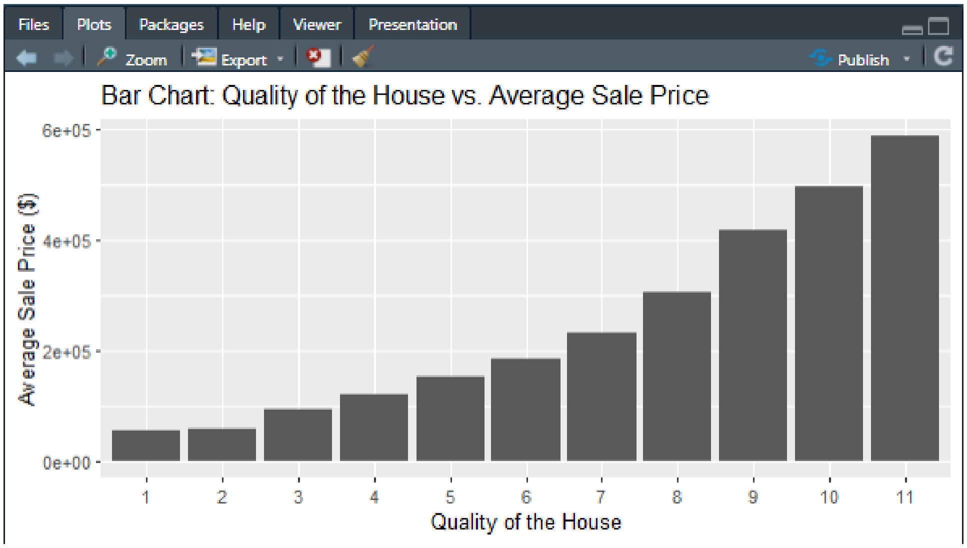
The R script makes use of ggplot2 to generate a boxplot that shows how property values change based on the amount of bedrooms they have. The code starts by creating a ggplot object and designating the dataset'stats_as' for display. It then uses aesthetic mappings ('aes') to connect the 'bedroom' data to the x-axis as a category component, and the'sale_price' data to the y-axis. It also determines the fill color for the plot's boxes, which is set statically to 'pink'. The actual boxplot is created using the 'geom_boxplot()' method. This boxplot depicts the distribution of selling prices over various groups or numbers of sleeping areas, enabling a comparison of how costs vary across these categories.

```
# Visualization 2: Boxplot - Number of Bedrooms vs. Sale Price
ggplot(stats_as, aes(x = as.factor(bedroom), y = sale_price, fill= "pink")) +
  geom_boxplot() +
  labs(title = "Boxplot: Number of Bedrooms vs. Sale Price",
       x = "Number of Bedrooms",
       y = "Sale Price ($)")

```

The 'labs()' method is used to set names for plot components like the title and bars. The caption is "Boxplot: Number of Bedrooms vs. Sale Price," and the x-axis with y-axis have labels that say "Number of Bedrooms" and "Sale Price (in Dollars)," correspondingly. In conclusion, the resulting representation is a boxplot that successfully illustrates the distribution of selling prices for different kinds of bedroom occupancy. It provides critical statistical information such as median values, quartiles, the existence of outliers, and the general distribution of selling prices within each separate room count group () This image facilitates in the comparison of selling prices across various bedroom counts, offering significant insights into how they are distributed patterns.

3. **Bar Chart** - House Quality vs. Average Sale Price: Shows average sale prices broken down by house quality.



This R code creates a bar chart demonstrating the association between home quality (categorized) and median sale prices using ggplot2. It calculates and shows the mean sale price per every level of property quality as bars, giving an illustration of the median rates across various levels of quality by mapping housing quality to the x-axis and selling prices to the y-axis.

```
# Visualization 3: Bar Chart - Quality of the House vs. Average Sale Price
ggplot(stats_as, aes(x = as.factor(house_quality), y = sale_price)) +
  stat_summary(fun = mean, geom = "bar") +
  labs(title = "Bar Chart: Quality of the House vs. Average Sale Price",
       x = "Quality of the House",
       y = "Average Sale Price ($)")
```

The laboratory function is used in this code to set the title and axis labels for a bar chart. It depicts the link between different housing quality categories (on the x-axis) and their typical sale prices (on the y-axis). Each bar reflects the median selling price across certain quality categories, allowing you to compare average costs across different property situations. This graphic provides a plain picture of how housing quality affects sale prices, allowing for quick discovery of major variances or trends across different quality categories.

4. **Bar Chart** -

Stories vs. Average Sale Price: Shows the average sale prices for properties based on the number of stories.

The R code builds a bar chart using ggplot2 that shows how the amount of real estate stories related to their median sale values. It computes mean prices for each tale type using stat_summary(), then plots bars to reflect these averages (Birch *et al*, 2003). The graph is headed "Bar Chart: Stories vs. Average Sale Price," and the x-axis is labeled "Number of Stories" and the y-axis is labeled "Average Sale Price ($)."

```
# Visualization 3: Bar Chart - Quality of the House vs. Average Sale Price
ggplot(stats_as, aes(x = as.factor(house_quality), y = sale_price)) +
  stat_summary(fun = mean, geom = "bar") +
  labs(title = "Bar Chart: Quality of the House vs. Average Sale Price",
       x = "Quality of the House",
       y = "Average Sale Price ($)")
```

Overview:

Type of visualization: bar chart X-axis: number of tales in properties

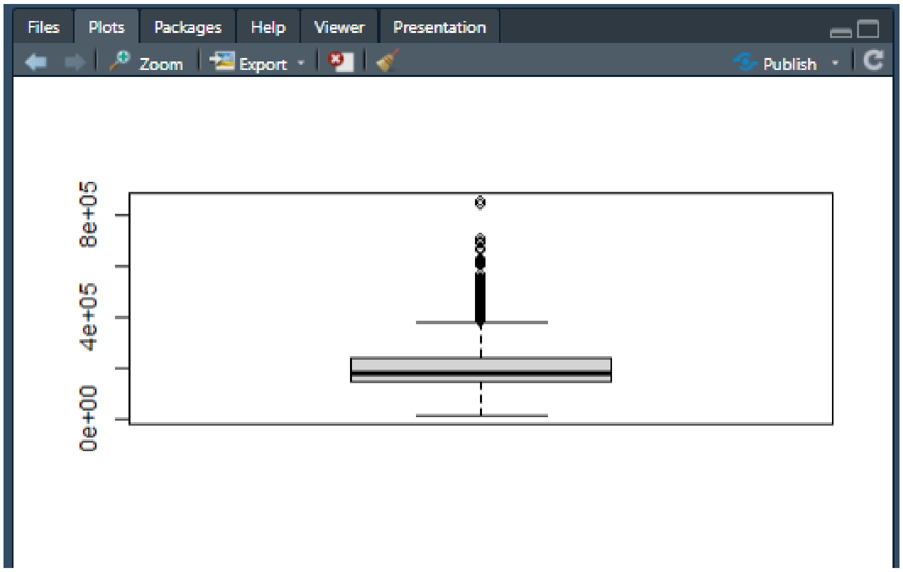
Average sale prices (in USD) are shown on the Y-axis.

Interpretation: Each bar indicates the average sale price for a certain number many stories, providing insight into how prices change depending on the quantity of stories in houses (Wang *et al, 2018*)

Insights: This visualization allows for simple comparison of average sale prices across different story counts, perhaps showing patterns or differences in purchase rates due to tale count.

5. **Box Plot** -

The box diagram aids in the visual identification of probable outliers—data points that deviate considerably from the rest of the dataset. These outliers may be located above or below the rectangle plot's whiskers, suggesting extreme values in the selling prices of properties in the dataset.



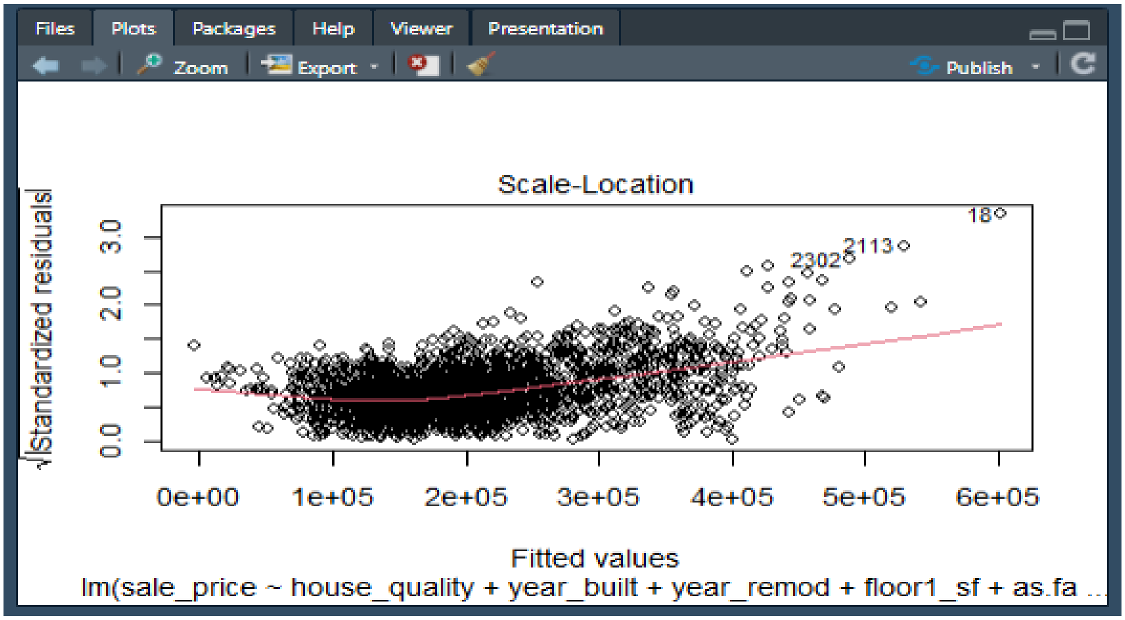
boxplot(data_chosen$sale_price): This function generates a box plot that illustrates the distribution of values in the'sale_price' variable. The box plot depicts the median, quartiles, and likely outliers of the selling price distribution.

```
# confirming Outliers (by box plot for 'sale_price' as an example)
boxplot(data_chosen$sale_price)
```

Knowing how to deal with outliers is critical in the analysis of statistics since they can have a substantial impact on the results or statistical models. Detecting anomalies is important for guaranteeing data analysis robustness and accuracy.

PREDICTIONS-

A. **Predicted Sale Prices vs. Actual Sale Prices**




```
#predict prices for the houses in the test set
multiple_price_predictions <- predict(multiple_model, newdata = test)
```

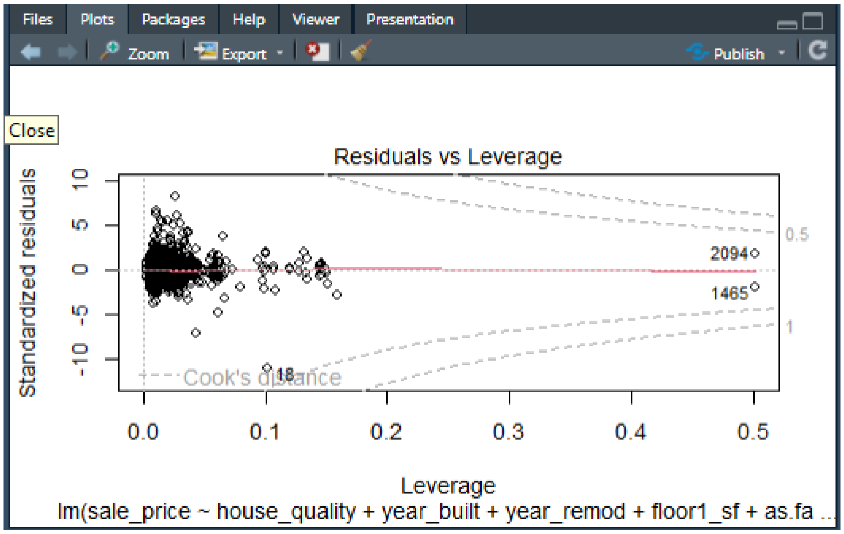
To anticipate selling prices, the investigation produced a multivariate linear regression model that took into account numerous property attributes. A graph named 'projected Sale Prices vs. Actual Sale Prices' might effectively illustrate the link between projected and actual sale costs by displaying the performance of the forecasts versus the real sale rates.(Valkov *et al*, 2019) A graph like this would help to assess the model's accuracy by comparing its predictions to the real values and emphasizing any differences or trends among both individual prices.

B. Predicted vs. Actual Sale Price

The method of regression incorporates several factors to estimate'sale_price'. The model summary describes the relevance and effect of each prediction on the desired variable. Evaluation indicators such as RMSE aid in determining the model's predicted accuracy.

```
#evaluate accuracy
postResample(multiple_price_predictions,test$sale_price)
sqrt(mean((multiple_price_predictions - test$sale_price)^2))
```

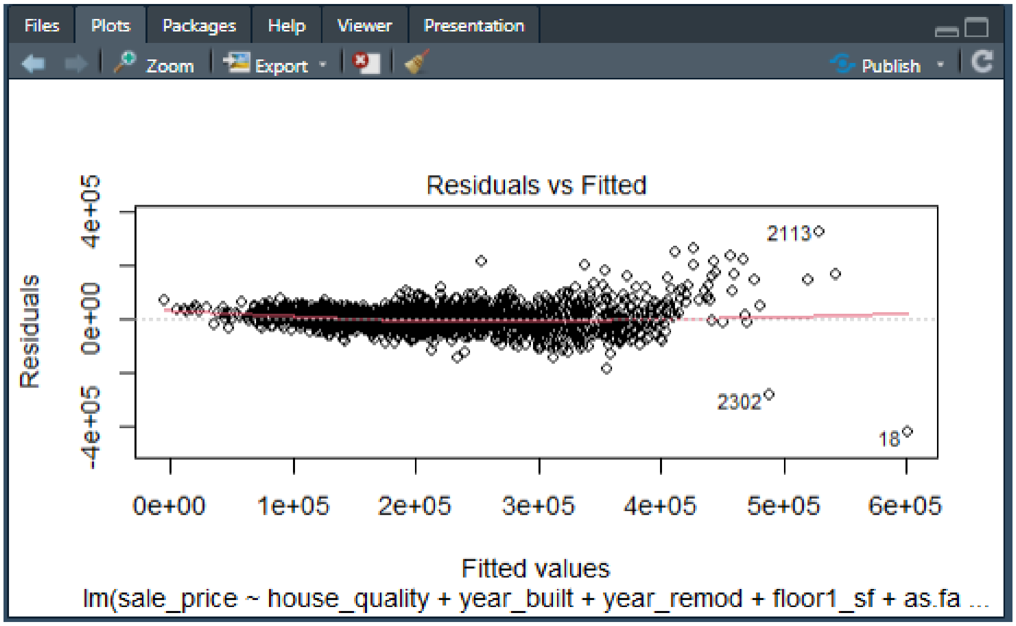
The graph compares projected sale prices to actual sale prices from the test dataset to show the accuracy with which the algorithm predicts the'sale_price'.



C. Regression Analysis and Diagnostic Evaluation

The R package 'car' offers diagnostic tools for evaluating regression models, including statistical summaries and graphical representations. It assists in the evaluation of theoretical bases such as linearity, homoscedasticity, or leftover patterns. Summary() functions give empirical data regarding coefficients, whereas diagnostic graphs, such as the Leverage vs. Residuals Squared plot, aid in model performance and premise adherence. These tests are critical for assuring the model's dependability and meeting the essential assumptions for accurate forecasts. (Hirk *et al, 2020)*

```
install.packages("lmtest")
library(lmtest)
dwtest(multiple_model)
```



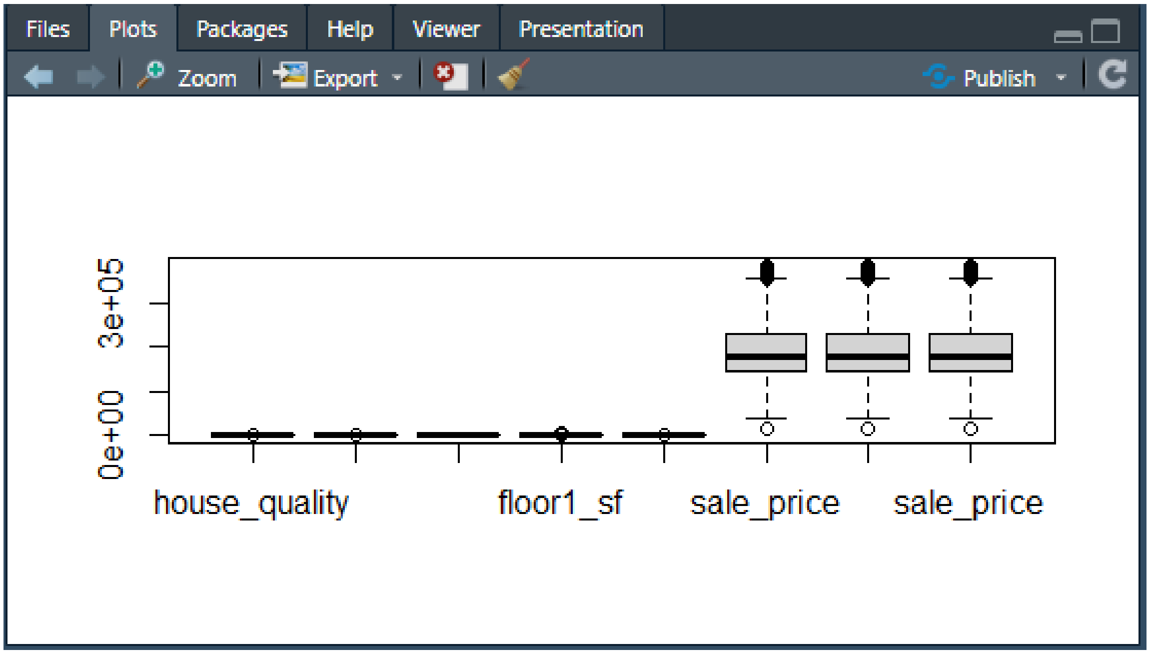
D. Predicting House Prices

The following line of code is required to estimate home prices by applying the trained regression model ('multiple_model') to fresh or previously unknown data ('test').

```
#predict prices for the houses in the test set
multiple_price_predictions <- predict(multiple_model, newdata = test)
#evaluate accuracy
```

The 'predict()' method leverages the model's previously learned correlations between predictors (such as house quality, year constructed, and so on) and house prices to generate projected price ranges for the houses included in the 'test' dataset (Zulkifley *et al*, 2020). These predictions are made based on the patterns seen during the model training phase, allowing the model's predictive skills on previously unknown data to be evaluated.

E. Outlier Detection and Duplicate Column Check in Cleaned Data-

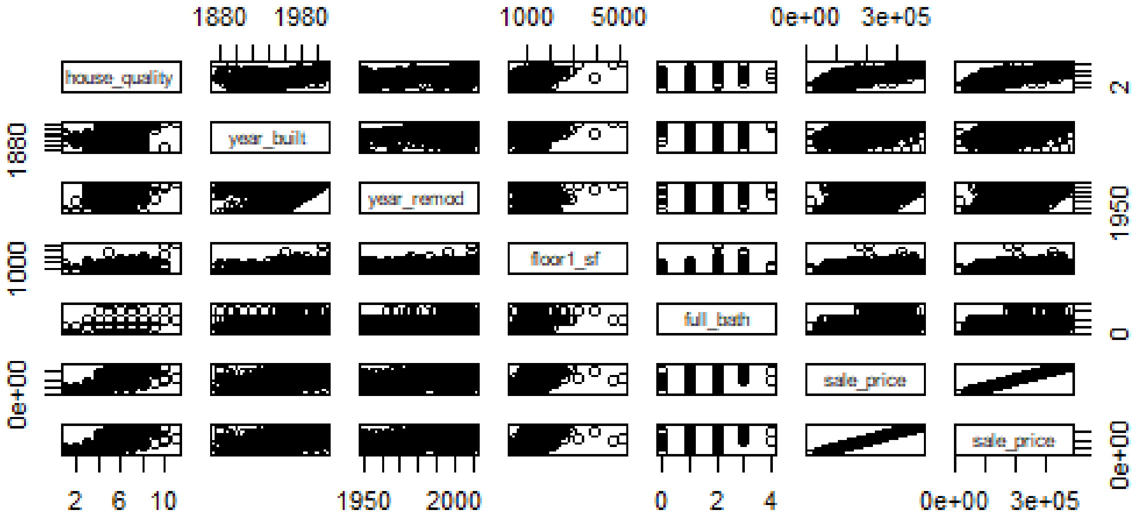


The code initially builds a boxplot that visualizes the distribution of numerical variables in the 'data_clean' dataframe, enabling any outliers in the dataset to be identified. The following line looks for identical names of columns inside the 'data_clean' data frame and returns a Boolean result indicating whether there are any. This validation validates the structural integrity of the dataset, guaranteeing that each variable is uniquely represented by its column name.

```
# Check for outliers
boxplot(data_clean)

any(duplicated(names(data_clean)))
```

Correlation model explanation-




```
# Subset the dataframe with selected variables
data_chosen <- data_chosen[, c(significant_variables, 'sale_price')]
summary(data_chosen)

correlation_matrix <- cor(data_chosen)
print(correlation_matrix['sale_price', ])
pairs(data_chosen)
model <- lm(sale_price ~ ., data = data_chosen)
summary(model)
model <- lm(sale_price ~ ., data = data_chosen)
summary(model)
```

This code sequence accomplishes the following operations:

Subsetting Data: The code picks particular variables identified as'significant_variables' from the 'data_chosen' dataframe, together with the'sale_price' column (Schober *et al*, 2016).

Correlation Analysis: This function computes the correlation matrix for the variables chosen and the'sale_price'. The code then displays the correlation coefficients between the variable'sale_price' and the other variables.

It creates a pairs plot to graphically analyze the links between'sale_price' and other factors. Linear Regression Modeling: The code creates a linear regression model in which the dependent variable is'sale_price' and all other variables are utilized as predictors. It appears to replicate the modeling process by running two models sequentially.

CONCLUSION-

To understand the link among various housing qualities and selling prices, the study includes data cleansing, model creation, and visualization.

Linear regression models were used to forecast selling prices based on various property features, and the assessment indicates that different models have diverse prediction accuracies.

Visualizations show how specific characteristics, such as property quality, number of bedrooms, and number of storeys, can affect sale prices.

Further statistical research and feature development might provide more insights into attribute connections with selling prices.

The conclusion of this code implies that home factors like as quality, size, location, and build year have varied degrees of effect on property selling prices, laying the groundwork for future investigation or model development to increase predicted accuracy.

#Appendix 1

CODE USED IN R-

```
#path

getwd()

#importing datas
library(readxl)
stats_as <- read_excel("C:/Users/Welcome To Computer/Downloads/stats_as.xlsx")
data_chosen <- stats_as[, c("house_quality", "year_built", "year_remod", "floor1_sf", "neighbourhood", "full_bath", "configuration", "slope", "lot_area",
"sale_price")]


# Removing rows whose values are missing
data_chosen <- na.omit(data_chosen)
# Assuming 'data_chosen' is your dataset
data_chosen <- na.omit(data_chosen)
# confirming Outliers (by box plot for 'sale_price' as an example)
boxplot(data_chosen$sale_price)
# we can also identify and remove outliers based on the box plot

outliers <- boxplot(data_chosen$sale_price, plot = FALSE)$out
data_chosen <- data_chosen[!data_chosen$sale_price %in% outliers, ]

# Changing Data Types (assuming 'neighbourhood' is a categorical variable)
data_chosen$neighbourhood <- as.factor(data_chosen$neighbourhood)
data_chosen$configuration <- as.factor(data_chosen$configuration)
data_chosen$slope <- as.factor(data_chosen$slope)


# Other Data Cleaning Steps as Needed
# Removing repeated Rows
data_chosen <- data_chosen[!duplicated(data_chosen), ]
```

```
# cleaned data
summary(data_chosen)



# visualizations
install.packages("ggplot2")
library(ggplot2)

# Visualization 1: Scatter Plot - Total no of rooms vs. Sale Price
ggplot(stats_as, aes(x = rooms_tot, y = sale_price)) +
  geom_point() +
  labs(title = "Scatter Plot: Total no of rooms vs. Sale Price",
       x = "Total no of rooms",
       y = "Sale Price ($)")


# Visualization 2: Boxplot - Number of Bedrooms vs. Sale Price
ggplot(stats_as, aes(x = as.factor(bedroom), y = sale_price, fill= "pink")) +
  geom_boxplot() +
  labs(title = "Boxplot: Number of Bedrooms vs. Sale Price",
       x = "Number of Bedrooms",
       y = "Sale Price ($)")

# Visualization 3: Bar Chart - Quality of the House vs. Average Sale Price
ggplot(stats_as, aes(x = as.factor(house_quality), y = sale_price)) +
  stat_summary(fun = mean, geom = "bar") +
  labs(title = "Bar Chart: Quality of the House vs. Average Sale Price",
       x = "Quality of the House",
       y = "Average Sale Price ($)")
```

```
# Visualization 4: Bar Chart - stories vs. Average Sale Price
ggplot(stats_as, aes(x = stories, y = sale_price)) +
  stat_summary(fun = mean, geom = "bar", fill = "skyblue") +
  labs(title = "Bar Chart: stories vs. Average Sale Price",
       x = "stories",
       y = "Average Sale Price ($)")

#spliting the data into a training and test dataset

set.seed(40425082)

install.packages("caret")
library(caret)

index <- createDataPartition(data_chosen$sale_price, list = FALSE, p=0.8, times=1)

train <- data_chosen[index,]
test <- data_chosen[-index,]

Model_A <- lm(sale_price ~ as.factor(house_quality), data = train)
summary(Model_A)

#prediction through this model_A

simple_price_prediction <- predict(Model_A, newdata = test)

# prediction accurecy

postResample(pred = simple_price_prediction, obs = test$sale_price)

#RMSE is the difference between observed and predicted value calculated as:
sqrt(mean((simple_price_prediction - test$sale_price)^2))
```

```
#multiple linear regression (Are these configuration, slope,full_bath,neighbourhood,floor1_sf,year_remod,year_build and house_quality related to sale_price))
#or how well can these factors predict the price

multiple_model <- lm(sale_price ~ house_quality + year_built + year_remod + floor1_sf + as.factor(neighbourhood) + full_bath + as.factor(configuration) +
as.factor(slope) + lot_area, data = train)
summary(multiple_model)

#predict prices for the houses in the test set
multiple_price_predictions <- predict(multiple_model, newdata = test)
#evaluate accuracy
postResample(multiple_price_predictions,test$sale_price)
sqrt(mean((multiple_price_predictions - test$sale_price)^2))

install.packages("car")
library(car)

vif(multiple_model)
plot(multiple_model)

install.packages("lmtest")
library(lmtest)
dwtest(multiple_model)

cook <- cooks.distance(multiple_model)
summary(cook)
sum(cook > 1)


# Assuming 'data' is your dataframe
numeric_data <- data_chosen[, sapply(data_chosen, is.numeric)]
```

```
# Calculate correlation matrix for numeric variables
correlation_matrix <- cor(numeric_data)
correlation_with_sale_price <- correlation_matrix['sale_price', ]

# Select variables with high correlation (you can choose a threshold, e.g., 0.5)
significant_variables <- names(correlation_with_sale_price[abs(correlation_with_sale_price) > 0.5])

# Remove missing values from the vector
significant_variables <- significant_variables[complete.cases(significant_variables)]

# Subset the dataframe with selected variables
data_chosen <- data_chosen[, c(significant_variables, 'sale_price')]
summary(data_chosen)

correlation_matrix <- cor(data_chosen)
print(correlation_matrix['sale_price', ])
pairs(data_chosen)
model <- lm(sale_price ~ ., data = data_chosen)
summary(model)
model <- lm(sale_price ~ ., data = data_chosen)
summary(model)

#data formatting
# Check for missing values
sum(is.na(data_chosen))

# Address missing values (impute or remove)
data_clean <- na.omit(data_chosen)

# Check data types
str(data_clean)

# Check for outliers
boxplot(data_clean)
```

```
any(duplicated(names(data_clean)))
# Identify duplicate column names
dup_cols <- names(data_clean)[duplicated(names(data_clean))]

# Rename duplicate columns directly in data_clean
for (col in dup_cols) {
  # Identify indices of duplicate columns
  dup_indices <- which(names(data_clean) == col)

  # Append index to duplicate column names to make them unique
  for (i in seq_along(dup_indices)) {
    names(data_clean)[dup_indices[i]] <- paste0(col, "_duplicate", i)
  }
}

# Retry the ggplot code
ggplot(data_clean, aes(x = house_quality, y = sale_price)) +
  geom_point() +
  labs(title = "Relationship between House Quality and Sale Prices",
       x = "House_Quality",
       y = "Sale_Price")


summary(data_clean)
```

References

Post, F.H., Nielson, G. and Bonneau, G.-P. (2002). *Data Visualization: The State of the Art*. [online] *Google Books*. Springer Science & Business Media. Available at: https://books.google.co.uk/books?hl=en&lr=&id=WAZYsfeMi4kC&oi=fnd&pg=PR9&dq=DATA+VISUALIZATION+USING+R&ots=gF9RVr4N7u&sig=ELgGEa2v01uFNoCweMDXZTvTcQw&redir_esc=y#v=onepage&q&f=false [Accessed 6 Dec. 2023].

Woźniak, M., Grana, M. and Corchado, E. (2014) 'A Survey of Multiple Classifier Systems as Hybrid Systems.', Information Fusion, 16(), pp. 3-17 [Online]. Available at: https://doi.org/10.1016/j.inffus.2013.04.006 (Accessed: 7 December 2022).

R intro. (1990). Available at: https://cran.r-project.org/doc/manuals/r-release/R-intro.pdf.

Schober, P., Boer, C. and Schwarte, L.A. (2018) ' Correlation Coefficients: Appropriate Use And Interpretation.', Anesthesia & Analgesia, 126(5), pp. 1763-1768 [Online]. Available at: https://doi.org/10.1213/ANE.0000000000002864 (Accessed: 7 December 2020).

Grolemund, G. and Wickham, H. (2017). *R for Data Science*. [online] Had.co.nz. Available at: https://r4ds.had.co.nz/.

Gharehchopogh, F.S., Bonab, T.H. and Khaze, S.R. (2013) 'A Linear Regression Approach to Prediction of Stock Market Trading Volume: A Case Study', International Journal of Managing Value and Supply Chains, 4(3), pp. 25 [Online]. Available at: https://d1wqtxts1xzle7.cloudfront.net/57159517/4313ijmvsc03-libre.pdf?1533817693=&response-content-disposition=inline%3B+filename%3DA_LINEAR_REGRESSION_APPROACH_TO_PREDICTI.pdf&Expires=1670536127&Signature=XGPEIz6f~rvyudiPDJ7L62cQ-zmto1zG0sXlzRdLdmevE3WRVjqw2c~OvxIaAhOEUi~fz~82r4U67PKKi9dVYhqW8HASbTa3EU86h6mN9IX8le-b~8A6Uk6Nu-vYm7tUipUADWElUCyewHA78YC1eVuhPSxT~6ZMrxZMpEk4TqeLwdkQJQJe3yJ6b61LWsGJGiv38P7yVHy2q45kpH~lA1ntqFgx0CWcmjMSI4loY7UNxAzSu15m0M7tGho3yUH9wVVfq-u1cYpjPziqw869I6~teBPXEuylLWcFmvHpo6vApdMnAO8Kaf3NZMbZNl4zvqPCy~1GHZmNw4UTjq2EkQ__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA (Accessed: 6 December 2022).

Hirk, R., Hornik, K. and Vana, L. (2020). mvord: An R Package for Fitting Multivariate Ordinal Regression Models. *Journal of Statistical Software*, 93(4). doi:https://doi.org/10.18637/jss.v093.i04.

Birch, J. and Sunderman, M (2003) 'Estimating Price Paths for Residential Real Estate', Journal of Real Estate Research, 25(3), pp. pp.277-300 [Online]. Available at: https://doi.org/10.1080/10835547.2003.12091115 (Accessed: 6 December 2022).

Sainani, K.L. (2016) 'The value of scatter plots', PM&R, 8(12), pp. 1213-1217 [Online]. Available at: https://doi.org/10.1016/j.pmrj.2016.10.018 (Accessed: 7 December 2020).

Zulkifley, N.H., Rahman, S.A., Ubaidullah, N.H. and Ibrahim, I. (2020) 'House Price Prediction using a Machine Learning Model: A Survey of Literature', International Journal of Modern Education & Computer Science, 12(6), pp. 46-54 [Online]. Available at: https://www.mecs-press.org/ijmecs/ijmecs-v12-n6/IJMECS-V12-N6-4.pdf (Accessed: 7 December 2020).

Valkov, V. (2019) Predicting House Prices with Linear Regression | Machine Learning from Scratch (Part II), Available at: https://towardsdatascience.com/predicting-house-prices-with-linear-regression-machine-learning-from-scratch-part-ii-47a0238aeac1 (Accessed: 6 December 2022).

Wang, C. and Wu, H. (2018) 'A New Machine Learning Approach to House Price Estimation.', New Trends in Mathematical Sciences, 6(4), pp. 165-171 [Online]. Available at: https://www.researchgate.net/publication/330001343_A_new_machine_learning_approach_to_house_price_estimation (Accessed: 7 December 2020).