Predicting house price of Ames, Iowa using the Regression model

Machine Learning Data Science

I. Introduction

House is very important in people's life; everybody wants to buy a house as it is a place where people feel safe and relaxed after intense work from their works. The price of houses depends on the different features of the houses. Home price predictions will help them to decide whether the house they want to buy is worth that price or not. House price prediction not only helps the buyer but also it will be helpful for the seller, as they could decide what is the best price for the house, they want to sell keeping in mind all the features of the house and the neighborhood where it is located.

The goal of the project was to build machine learning models to predict house prices located in Ames, Iowa by using Multiple Linear Regression. We will predict the house prices in any Neighborhood of Ames, Iowa.

It is an interesting project as people want to buy houses but do not know how to determine the prices, every year the prices of houses go on the increase and people do not have reliable sources to believe whether the prices determined by the houses are correct or not, by determining and analyzing all the parameters of the dataset, people themselves can predict the houses prices. It will be useful for both the buyers and sellers around Ames, Iowa. Data scientists can use this model as a base and predict house prices for other places as well.

For a large dataset like this one, calculating the prices of the houses manually is very difficult, if we design the model using machine learning approaches, then by using the multiple regression model which has the best accuracy by analyzing RMSE(Root Mean Square Error), we can automatically calculate the prices of the houses within less time. We can use the machine learning model to calculate the prices of the number of houses at a time, which will save time and money.

Keywords: Machine learning, Multiple Linear Regression, Ames House, House price prediction, RMSE

II. Background

The data set contains information from the Ames Assessor's Office used in computing assessed values for individual residential properties sold in Ames, IA from 2006 to 2010. The data has 2930 rows and 82 variables, 82 columns include 23 nominal, 23 ordinal, 14 discrete, and 20 continuous variables (and 2 additional observation identifiers). The main objective behind this project is to develop a model to predict the selling price of a given home in Ames, Iowa. Real estate investors can use this information to help assess whether the asking price of a house is higher or lower than the true value of the house. If the home is undervalued, it may be a good investment for the firm. The overall message I need to convey after analyzing this data set will be to determine the key features among 82 features that are mainly responsible to determine the price of the houses, and uses it to develop the regression model, finally using this reliable model we can predict the price of the house for future data set of Ames, Iowa.

The research question will be- Are the features like overall quality, area, build year, neighborhood, and external quality of the house determine the price of the Ames, Iowa houses?

The dataset contains 2930 observations and 82 features of the Ames, Iowa houses. The 82 features of the houses consist of area, external quality, overall quality, build year, neighborhood, floor square foot, etc., The 82 features are more than enough to determine the price of the houses using the regression model I am using for this dataset. The available data is sufficient to predict the price of the houses.

III. Dataset description

I have downloaded this Ames, Iowa house prediction dataset from Kaggle. (https://www.kaggle.com/datasets/marcopale/housing) . The source of the data is Ames, Iowa Assessor's Office. The data set contains information from the Ames Assessor's Office used in computing assessed values for individual residential properties sold in Ames, IA from 2006 to 2010.

The dataset contains 2930 observations and 82 feature variables of houses that are present in Ames, Iowa. By analyzing the dataset, we can develop reliable machine learning models, and using these models, we can predict the prices of the houses that are beneficial for both the customers and the people who want to sell their houses. It means a lot to me as I am a data science student who wants to learn the machine learning model, this dataset provides me an opportunity to learn the regression model which I am using for this project, not only that, but I will also have good knowledge about the regression model which will be good skills for my future data science career.

The dataset has 2930 rows and 82 variables, 82 columns include 23 nominal, 23 ordinal, 14 discrete, and 20 continuous variables (and 2 additional observation identifiers). Out of 82 variables, 81 variables are independent variables, and the target or dependent variable is the price. With the help of independent variables, we will predict the price of the houses. Some of the important independent variables included in the dataset are area, external quality, overall quality, build year, neighborhood, floor square foot, etc., While exploring the data, the minimum price of the house was found 12,789 while the maximum price of the house was 755,000, whereas the average price was 180,796. After visualizing the data, it seems that there has been a housing boom during the early 2000s. After analyzing the overall condition variable, it was found that the overall condition of the houses were more than the average condition and that there are more well-maintained houses and the most houses were under 2000 square feet. From the descriptive statistics, it seems that the mean value is 1500 sq.ft. and the median is 1442 sq.ft. After observing the neighborhood parameters, the neighborhood influences the prices of houses - the most expensive parts of the city have prices three times higher than the cheapest areas. After calculating the correlation coefficient of price with different variables, it is found that the enclosed porch and PID were negatively correlated. The variables that were positively correlated with the prices were overall quality, area, first-floor sq ft, full Bath, Total room above ground, and year built after calculating correlation. The External Quality of the material of the houses may also increase the prices of the houses. We will not use all the features of the dataset to build the model. Rather, we will only use important features (independent variables) to build our multiple regression models.

There were 2930 observations and 82 variables in the dataset. It was not possible and appropriate to use all the independent variables while building the model. So, I selected the important features that influence the prices of the houses. For the numeric variables, I used the correlation to check the relationship between the independent variables and the target variable(price). And for the character variables, I used a scatter plot and bar graph to determine the relationship between the variables. While building the model, I used only those variables that are highly correlated with the target variable of the dataset.

The variables I used for building the multiple regression model were- overall quality, first-floor square feet, Garage area, year build, external quality, and full bath, Masonry veneer area in square feet. For the variables which do not show a strong relationship with the target variable, I omitted those variables while building the model. I have used the correlation coefficient and heat map visualization to determine the strength of the relationship between independent variables and the price of the houses. While checking the null values of the features of the dataset, I found that some of the variables have null values. Those variables were not important for building the model, as they have no positive strong correlation with the price of the house except the Garage area variable and the Masonry veneer area. I have replaced those null values with the mean values of the variables respectively.

IV. Model's Baseline

In this model, I am using a multiple regression model. The regression method is used to forecast the numeric data and it is one of the easy ways to find trends in the data. For example, in our model, I am using different variables to find the connection between those independent variables with the price of the houses. Regression provides an equation for a graph so that we can make a prediction from the data. It gives statistics (p-value and correlation coefficient) to tell how your model predicts. Multiple regression is an extension of simple linear regression models that allow predictions of systems with multiple variables.

$Y=a + b_1X_1 + b_2X_2 + b_3X_3$

In multiple regression, there is more than one independent variable as shown in the equation above. Both the simple and multiple regression techniques assume that the dependent variable is measured on a continuous scale.

Multiple linear regression is a wonderful algorithm for predicting continuous values. The house price is also a continuous value. The great advantage of regression models is that they can be used to capture important relationships between the forecast variable of interest and the predictor variables. Building this model is very easy and effective once you know the relationship between the independent variables and the dependent variable. We can use all the numeric and character-independent variables while building the model, for the character nominal variables, it will treat as numeric variables by creating dummy variables while building the multiple regression model.

The pros of the multiple linear algorithms are:

- By far the most appropriate for modeling numeric data.
- The ability to determine the relative influence of one or more predictor variables on the criterion value.
- The ability to identify outliers, or anomalies
- Can be adapted to model almost any modeling task.
- Provides the estimates of both strength and size of the relationships among the features and the outcomes.

The cons of the multiple linear algorithms are:

- Multiple linear regression makes strong assumptions about the data.
- The regression model will not be good at explaining the relationship of the independent variables to the dependent variables if those relationships are not linear
- Does not handle missing data.
- Only works with numeric features, so categorical data requires extra processing, (but done by the model itself).
- Requires some knowledge of statistics to understand the model.

V. Building your machine learning model

- Step 1: Import required libraries.

```
> library(dplyr)

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union

> library(ggplot2)
Want to understand how all the pieces fit together? Read R for Data Science:
https://r4ds.had.co.nz/
> options(scipen = 999)
> install.packages("GGally")
trying URL 'https://cran.rstudio.com/bin/macosx/contrib/4.2/GGally_2.1.2.tgz'
Content type 'application/x-gzip' length 1636473 bytes (1.6 MB)
==================================================
downloaded 1.6 MB


The downloaded binary packages are in
        /var/folders/r7/32bm9v6164j0k8z2qvjcfrq40000gn/T//RtmpzYyqaa/downloaded_packages
> library(GGally)
Registered S3 method overwritten by 'GGally':
  method from
  +.gg   ggplot2
> library(caTools)
```

```
> install.packages("Metrics")
trying URL 'https://cran.rstudio.com/bin/macosx/contrib/4.2/Metrics_0.1.4.tgz'
Content type 'application/x-gzip' length 81807 bytes (79 KB)
==================================================
downloaded 79 KB


The downloaded binary packages are in
        /var/folders/r7/32bm9v6164j0k8z2qvjcfrq40000gn/T//RtmpzYyqaa/downloaded_packages
> library(Metrics)
```

The libraries I used were- library dplyr: It is a grammar of data manipulation, providing a consistent set of verbs that helps to solve the most common data manipulation challenges. To avoid all scientific notation options(scipen=999) is used. The ggcorr function is a visualization function to plot correlation matrixes as ggplot2 objects. The ggcorr is available through the GGally package. The library caTools is needed for splitting data (for importing sample()function). The library Metrics is needed for calculating the root mean square error.

- Step 2: Load the data set.

```
> setwd("/Users/nirajkc/Desktop")
> sales= read.csv("AmesHousing.csv")
> head(sales)
  Order       PID area  price MS.SubClass MS.Zoning Lot.Frontage Lot.Area Street Alley Lot.Shape Land.Contour
1     1 526301100 1656 215000          20        RL          141    31770   Pave  <NA>       IR1          Lvl
2     2 526350040  896 105000          20        RH           80    11622   Pave  <NA>       Reg          Lvl
3     3 526351010 1329 172000          20        RL           81    14267   Pave  <NA>       IR1          Lvl
4     4 526353030 2110 244000          20        RL           93    11160   Pave  <NA>       Reg          Lvl
5     5 527105010 1629 189900          60        RL           74    13830   Pave  <NA>       IR1          Lvl
6     6 527105030 1604 195500          60        RL           78     9978   Pave  <NA>       IR1          Lvl
  Utilities Lot.Config Land.Slope Neighborhood Condition.1 Condition.2 Bldg.Type House.Style Overall.Qual
1    AllPub     Corner        Gtl        NAmes        Norm        Norm      1Fam      1Story            6
2    AllPub     Inside        Gtl        NAmes       Feedr        Norm      1Fam      1Story            5
3    AllPub     Corner        Gtl        NAmes        Norm        Norm      1Fam      1Story            6
4    AllPub     Corner        Gtl        NAmes        Norm        Norm      1Fam      1Story            7
5    AllPub     Inside        Gtl      Gilbert        Norm        Norm      1Fam      2Story            5
6    AllPub     Inside        Gtl      Gilbert        Norm        Norm      1Fam      2Story            6
  Overall.Cond Year.Built Year.Remod.Add Roof.Style Roof.Matl Exterior.1st Exterior.2nd Mas.Vnr.Type
1            5       1960           1960        Hip   CompShg      BrkFace      Plywood        Stone
2            6       1961           1961      Gable   CompShg      VinylSd      VinylSd         None
3            6       1958           1958        Hip   CompShg      Wd Sdng      Wd Sdng      BrkFace
4            5       1968           1968        Hip   CompShg      BrkFace      BrkFace         None
5            5       1997           1998      Gable   CompShg      VinylSd      VinylSd         None
6            6       1998           1998      Gable   CompShg      VinylSd      VinylSd      BrkFace
  Mas.Vnr.Area Exter.Qual Exter.Cond Foundation Bsmt.Qual Bsmt.Cond Bsmt.Exposure BsmtFin.Type.1 BsmtFin.SF.1
1          112         TA         TA     CBlock        TA        Gd            Gd            BLQ          639
2            0         TA         TA     CBlock        TA        TA            No            Rec          468
3          108         TA         TA     CBlock        TA        TA            No            ALQ          923
4            0         Gd         TA     CBlock        TA        TA            No            ALQ         1065
5            0         TA         TA      PConc        Gd        TA            No            GLQ          791
6           20         TA         TA      PConc        TA        TA            No            GLQ          602
  BsmtFin.Type.2 BsmtFin.SF.2 Bsmt.Unf.SF Total.Bsmt.SF Heating Heating.QC Central.Air Electrical X1st.Flr.SF
1            Unf            0         441          1080    GasA         Fa           Y      SBrkr        1656
2            LwQ          144         270           882    GasA         TA           Y      SBrkr         896
3            Unf            0         406          1329    GasA         TA           Y      SBrkr        1329
```

read.csv() function has used the read the AmesHousing dataset from folder where dataset is stored.

- Step 3: Check the structure of the dataset.

```
> str(sales)
'data.frame':   2930 obs. of  82 variables:
 $ Order          : int  1 2 3 4 5 6 7 8 9 10 ...
 $ PID            : int  526301100 526350040 526351010 526353030 527105010 527105030 527127150 527145080 527146030 52
7162130 ...
 $ area           : int  1656 896 1329 2110 1629 1604 1338 1280 1616 1804 ...
 $ price          : int  215000 105000 172000 244000 189900 195500 213500 191500 236500 189000 ...
 $ MS.SubClass    : int  20 20 20 20 60 60 120 120 120 60 ...
 $ MS.Zoning      : chr  "RL" "RH" "RL" "RL" ...
 $ Lot.Frontage   : int  141 80 81 93 74 78 41 43 39 60 ...
 $ Lot.Area       : int  31770 11622 14267 11160 13830 9978 4920 5005 5389 7500 ...
 $ Street         : chr  "Pave" "Pave" "Pave" "Pave" ...
 $ Alley          : chr  NA NA NA NA ...
 $ Lot.Shape      : chr  "IR1" "Reg" "IR1" "Reg" ...
 $ Land.Contour   : chr  "Lvl" "Lvl" "Lvl" "Lvl" ...
 $ Utilities      : chr  "AllPub" "AllPub" "AllPub" "AllPub" ...
 $ Lot.Config     : chr  "Corner" "Inside" "Corner" "Corner" ...
 $ Land.Slope     : chr  "Gtl" "Gtl" "Gtl" "Gtl" ...
 $ Neighborhood   : chr  "NAmes" "NAmes" "NAmes" "NAmes" ...
 $ Condition.1    : chr  "Norm" "Feedr" "Norm" "Norm" ...
 $ Condition.2    : chr  "Norm" "Norm" "Norm" "Norm" ...
 $ Bldg.Type      : chr  "1Fam" "1Fam" "1Fam" "1Fam" ...
 $ House.Style    : chr  "1Story" "1Story" "1Story" "1Story" ...
 $ Overall.Qual   : int  6 5 6 7 5 6 8 8 8 7 ...
 $ Overall.Cond   : int  5 6 6 5 5 6 5 5 5 5 ...
 $ Year.Built     : int  1960 1961 1958 1968 1997 1998 2001 1992 1995 1999 ...
 $ Year.Remod.Add : int  1960 1961 1958 1968 1998 1998 2001 1992 1996 1999 ...
 $ Roof.Style     : chr  "Hip" "Gable" "Hip" "Hip" ...
 $ Roof.Matl      : chr  "CompShg" "CompShg" "CompShg" "CompShg" ...
 $ Exterior.1st   : chr  "BrkFace" "VinylSd" "Wd Sdng" "BrkFace" ...
 $ Exterior.2nd   : chr  "Plywood" "VinylSd" "Wd Sdng" "BrkFace" ...
 $ Mas.Vnr.Type   : chr  "Stone" "None" "BrkFace" "None" ...
 $ Mas.Vnr.Area   : int  112 0 108 0 0 20 0 0 0 0 ...
 $ Exter.Qual     : chr  "TA" "TA" "TA" "Gd" ...
 $ Exter.Cond     : chr  "TA" "TA" "TA" "TA" ...
 $ Foundation     : chr  "CBlock" "CBlock" "CBlock" "CBlock" ...
 $ Bsmt.Qual      : chr  "TA" "TA" "TA" "TA" ...
```

```
 $ Foundation     : chr  "CBlock" "CBlock" "CBlock" "CBlock" ...
 $ Bsmt.Qual      : chr  "TA" "TA" "TA" "TA" ...
 $ Bsmt.Cond      : chr  "Gd" "TA" "TA" "TA" ...
 $ Bsmt.Exposure  : chr  "Gd" "No" "No" "No" ...
 $ BsmtFin.Type.1 : chr  "BLQ" "Rec" "ALQ" "ALQ" ...
 $ BsmtFin.SF.1   : int  639 468 923 1065 791 602 616 263 1180 0 ...
 $ BsmtFin.Type.2 : chr  "Unf" "LwQ" "Unf" "Unf" ...
 $ BsmtFin.SF.2   : int  0 144 0 0 0 0 0 0 0 0 ...
 $ Bsmt.Unf.SF    : int  441 270 406 1045 137 324 722 1017 415 994 ...
 $ Total.Bsmt.SF  : int  1080 882 1329 2110 928 926 1338 1280 1595 994 ...
 $ Heating        : chr  "GasA" "GasA" "GasA" "GasA" ...
 $ Heating.QC     : chr  "Fa" "TA" "TA" "Ex" ...
 $ Central.Air    : chr  "Y" "Y" "Y" "Y" ...
 $ Electrical     : chr  "SBrkr" "SBrkr" "SBrkr" "SBrkr" ...
 $ X1st.Flr.SF    : int  1656 896 1329 2110 928 926 1338 1280 1616 1028 ...
 $ X2nd.Flr.SF    : int  0 0 0 0 701 678 0 0 0 776 ...
 $ Low.Qual.Fin.SF: int  0 0 0 0 0 0 0 0 0 0 ...
 $ Bsmt.Full.Bath : int  1 0 0 1 0 0 1 0 1 0 ...
 $ Bsmt.Half.Bath : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Full.Bath      : int  1 1 1 2 2 2 2 2 2 ...
 $ Half.Bath      : int  0 0 1 1 1 1 0 0 0 1 ...
 $ Bedroom.AbvGr  : int  3 2 3 3 3 3 2 2 2 3 ...
 $ Kitchen.AbvGr  : int  1 1 1 1 1 1 1 1 1 1 ...
 $ Kitchen.Qual   : chr  "TA" "TA" "Gd" "Ex" ...
 $ TotRms.AbvGrd  : int  7 5 6 8 6 7 6 5 5 7 ...
 $ Functional     : chr  "Typ" "Typ" "Typ" "Typ" ...
 $ Fireplaces     : int  2 0 0 2 1 1 0 0 1 1 ...
 $ Fireplace.Qu   : chr  "Gd" NA NA "TA" ...
 $ Garage.Type    : chr  "Attchd" "Attchd" "Attchd" "Attchd" ...
 $ Garage.Yr.Blt  : int  1960 1961 1958 1968 1997 1998 2001 1992 1995 1999 ...
 $ Garage.Finish  : chr  "Fin" "Unf" "Unf" "Fin" ...
 $ Garage.Cars    : int  2 1 1 2 2 2 2 2 2 2 ...
 $ Garage.Area    : int  528 730 312 522 482 470 582 506 608 442 ...
 $ Garage.Qual    : chr  "TA" "TA" "TA" "TA" ...
 $ Garage.Cond    : chr  "TA" "TA" "TA" "TA" ...
 $ Paved.Drive    : chr  "P" "Y" "Y" "Y" ...
 $ Wood.Deck.SF   : int  210 140 393 0 212 360 0 0 237 140 ...
 $ Open.Porch.SF  : int  62 0 36 0 34 36 0 82 152 60 ...
 $ Enclosed.Porch : int  0 0 0 0 0 0 170 0 0 0
```

The dataset has 2930 rows and 82 variables. The 82 columns include 23 nominal, 23 ordinal, 14 discrete, and 20 continuous variables (and 2 additional observation identifiers). Out of 82 variables, 81 variables are independent variables, and the target or dependent variable is the price.

#Now checking the null values from the dataset.

```
> sort(colSums(is.na(sales)))
         Order            PID           area          price     MS.SubClass      MS.Zoning       Lot.Area         Street    Lot.Shape   Land.Contour      Utilities
             0              0              0              0              0              0              0              0              0              0              0
    Lot.Config     Land.Slope   Neighborhood    Condition.1    Condition.2     Bldg.Type    House.Style   Overall.Qual   Overall.Cond     Year.Built Year.Remod.Add
             0              0              0              0              0              0              0              0              0              0              0
    Roof.Style      Roof.Matl    Exterior.1st    Exterior.2nd    Mas.Vnr.Type     Exter.Qual     Exter.Cond     Foundation        Heating     Heating.QC    Central.Air
             0              0              0              0              0              0              0              0              0              0              0
    Electrical     X1st.Flr.SF    X2nd.Flr.SF Low.Qual.Fin.SF      Full.Bath      Half.Bath  Bedroom.AbvGr  Kitchen.AbvGr   Kitchen.Qual   TotRms.AbvGrd     Functional
             0              0              0              0              0              0              0              0              0              0              0
    Fireplaces     Paved.Drive    Wood.Deck.SF   Open.Porch.SF Enclosed.Porch    X3Ssn.Porch    Screen.Porch      Pool.Area       Misc.Val         Mo.Sold       Yr.Sold
             0              0              0              0              0              0              0              0              0              0              0
     Sale.Type  Sale.Condition    BsmtFin.SF.1    BsmtFin.SF.2    Bsmt.Unf.SF   Total.Bsmt.SF     Garage.Cars     Garage.Area Bsmt.Full.Bath Bsmt.Half.Bath    Mas.Vnr.Area
             0              0              1              1              1              1              1              1              2              2             23
     Bsmt.Qual      Bsmt.Cond  Bsmt.Exposure  BsmtFin.Type.1  BsmtFin.Type.2    Garage.Type  Garage.Finish    Garage.Qual    Garage.Cond   Garage.Yr.Blt    Lot.Frontage
            79             79             79             79             79            157            157            158            158            159            490
   Fireplace.Qu          Fence          Alley   Misc.Feature        Pool.QC
          1422           2358           2732           2824           2917
> sum(is.na(sales$Garage.Area))
[1] 1
> sales$Garage.Area[is.na(sales$Garage.Area)]<-mean(sales$Garage.Area,na.rm=TRUE)
> sales$Mas.Vnr.Area[is.na(sales$Mas.Vnr.Area)]<-mean(sales$Mas.Vnr.Area,na.rm=TRUE)
> sum(is.na(sales$Garage.Area))
[1] 0
> sum(is.na(sales$Mas.Vnr.Area))
[1] 0
```

As we can see that some of the variables have null values, we have replaced null values with of the variables Garage area and Mas. Vnr area as these variables are important for our models, which shows a correlation with the prices of the houses.

- Step 4: Checking the summary.

```
> summary(sales)
     Order            PID                area            price          MS.SubClass      MS.Zoning         Lot.Frontage      Lot.Area       Street
 Min.   :   1.0   Min.   : 526301100   Min.   : 334   Min.   : 12789   Min.   : 20.00   Length:2930        Min.   : 21.00   Min.   :  1300   Length:2930
 1st Qu.: 733.2   1st Qu.: 528477022   1st Qu.:1126   1st Qu.:129500   1st Qu.: 20.00   Class :character   1st Qu.: 58.00   1st Qu.:  7440   Class :character
 Median :1465.5   Median : 535453620   Median :1442   Median :160000   Median : 50.00   Mode  :character   Median : 68.00   Median :  9436   Mode  :character
 Mean   :1465.5   Mean   : 714464497   Mean   :1500   Mean   :180796   Mean   : 57.39                      Mean   : 69.22   Mean   : 10148
 3rd Qu.:2197.8   3rd Qu.: 907181098   3rd Qu.:1743   3rd Qu.:213500   3rd Qu.: 70.00                      3rd Qu.: 80.00   3rd Qu.: 11555
 Max.   :2930.0   Max.   :1007100110   Max.   :5642   Max.   :755000   Max.   :190.00                      Max.   :313.00   Max.   :215245
                                                                                                           NA's   :490
    Alley           Lot.Shape         Land.Contour       Utilities         Lot.Config        Land.Slope        Neighborhood       Condition.1        Condition.2
 Length:2930        Length:2930        Length:2930        Length:2930        Length:2930        Length:2930        Length:2930        Length:2930        Length:2930
 Class :character   Class :character   Class :character   Class :character   Class :character   Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character


  Bldg.Type         House.Style         Overall.Qual     Overall.Cond     Year.Built    Year.Remod.Add  Roof.Style         Roof.Matl          Exterior.1st       Exterior.2nd
 Length:2930        Length:2930        Min.   : 1.000   Min.   :1.000   Min.   :1872   Min.   :1950   Length:2930        Length:2930        Length:2930        Length:2930
 Class :character   Class :character   1st Qu.: 5.000   1st Qu.:5.000   1st Qu.:1954   1st Qu.:1965   Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Median : 6.000   Median :5.000   Median :1973   Median :1993   Mode  :character   Mode  :character   Mode  :character   Mode  :character
                                       Mean   : 6.095   Mean   :5.563   Mean   :1971   Mean   :1984
                                       3rd Qu.: 7.000   3rd Qu.:6.000   3rd Qu.:2001   3rd Qu.:2004
                                       Max.   :10.000   Max.   :9.000   Max.   :2010   Max.   :2010

 Mas.Vnr.Type       Mas.Vnr.Area     Exter.Qual         Exter.Cond         Foundation         Bsmt.Qual          Bsmt.Cond          Bsmt.Exposure      BsmtFin.Type.1
 Length:2930        Min.   :   0.0   Length:2930        Length:2930        Length:2930        Length:2930        Length:2930        Length:2930        Length:2930
 Class :character   1st Qu.:   0.0   Class :character   Class :character   Class :character   Class :character   Class :character   Class :character   Class :character
 Mode  :character   Median :   0.0   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character
                    Mean   : 101.9
                    3rd Qu.: 162.8
                    Max.   :1600.0

 BsmtFin.SF.1    BsmtFin.Type.2     BsmtFin.SF.2     Bsmt.Unf.SF     Total.Bsmt.SF     Heating            Heating.QC         Central.Air        Electrical
 Min.   :   0.0   Length:2930        Min.   :   0.00   Min.   :   0.0   Min.   :   0   Length:2930        Length:2930        Length:2930        Length:2930
```

#Exploring some of the numeric variables with the summary() function

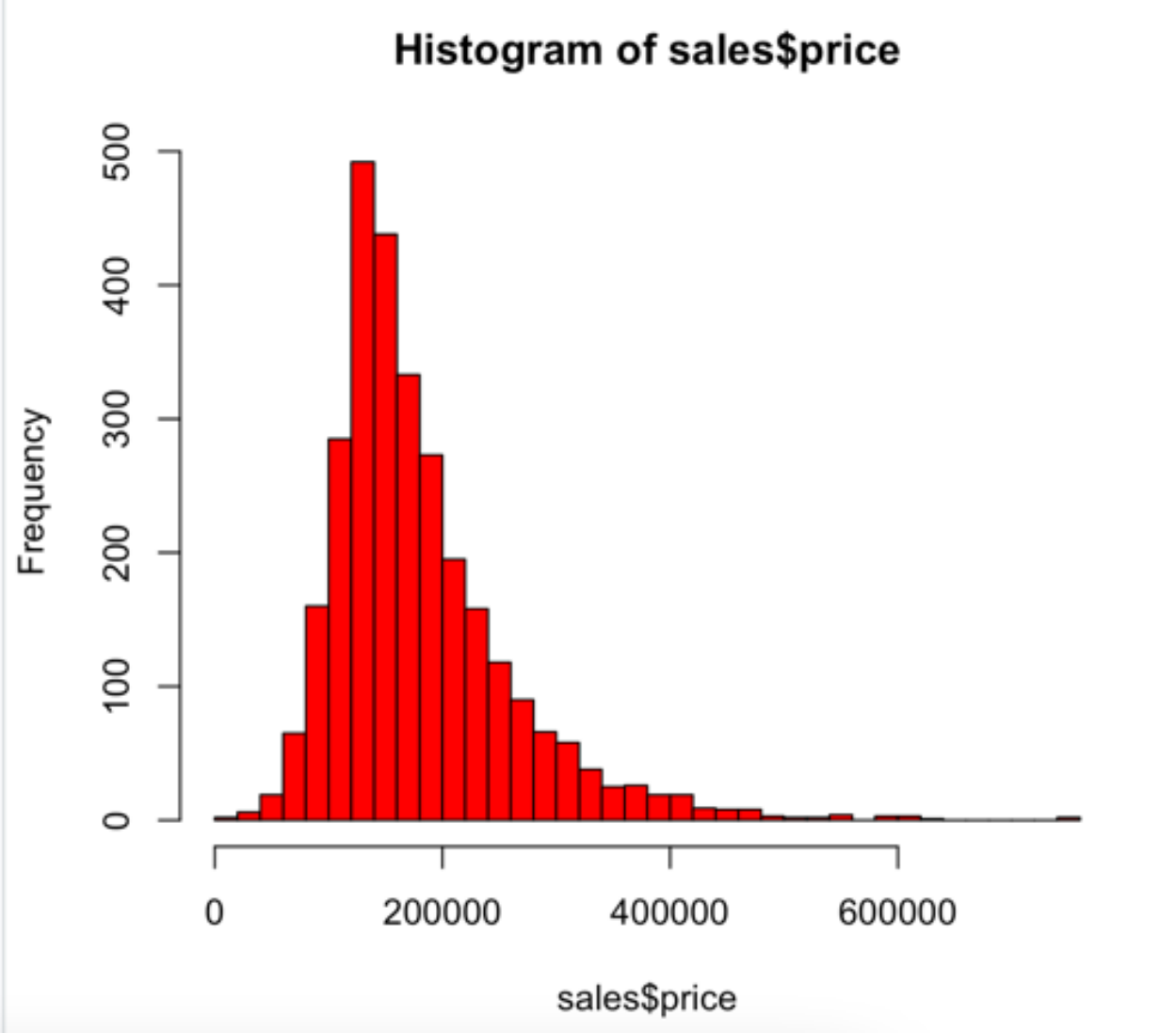
```
> summary(sales[c("price", "area","Mas.Vnr.Area","Garage.Area", "X1st.Flr.SF")])
     price             area        Mas.Vnr.Area     Garage.Area      X1st.Flr.SF
 Min.   : 12789   Min.   : 334   Min.   :   0.0   Min.   :   0.0   Min.   : 334.0
 1st Qu.:129500   1st Qu.:1126   1st Qu.:   0.0   1st Qu.: 320.0   1st Qu.: 876.2
 Median :160000   Median :1442   Median :   0.0   Median : 480.0   Median :1084.0
 Mean   :180796   Mean   :1500   Mean   : 101.9   Mean   : 472.8   Mean   :1159.6
 3rd Qu.:213500   3rd Qu.:1743   3rd Qu.: 162.8   3rd Qu.: 576.0   3rd Qu.:1384.0
 Max.   :755000   Max.   :5642   Max.   :1600.0   Max.   :1488.0   Max.   :5095.0
```

#Exploring some categorical variables with table() function

```
> table(sales$Overall.Qual)

  1   2   3   4   5   6   7   8   9  10
  4  13  40 226 825 732 602 350 107  31
> table(sales$Year.Built)

1872 1875 1879 1880 1882 1885 1890 1892 1893 1895 1896 1898 1900 1901 1902 1904 1905 1906 1907 1908 1910 1911 1912 1913 1914 1915 1916 1917 1918 1919 1920 1921 1922 1923 1924
   1    1    1    5    1    2    7    2    1    3    1    1   29    2    1    1    3    1    1    2   43    1    5    1    8   24   10    3   10    5   57   11   16   17   16
1925 1926 1927 1928 1929 1930 1931 1932 1934 1935 1936 1937 1938 1939 1940 1941 1942 1945 1946 1947 1948 1949 1950 1951 1952 1953 1954 1955 1956 1957 1958 1959 1960 1961 1962
  34   19    9    9    8   26    7    5    5   13   11    9   13   20   36   23    6   15   15   11   27   18   38   18   18   24   43   34   39   35   48   43   37   34   35
1963 1964 1965 1966 1967 1968 1969 1970 1971 1972 1973 1974 1975 1976 1977 1978 1979 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997
  35   33   34   35   41   45   28   42   39   40   21   23   25   54   57   42   21   27   10    7    8   19    7   11    8   15    8   19   12   27   40   37   31   34   35
1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010
  47   52   48   35   47   88   99  142  138  109   49   25    3
> table(sales$Exter.Qual)

  Ex   Fa   Gd   TA
 107   35  989 1799
> table(sales$Neighborhood)

Blmngtn Blueste  BrDale BrkSide ClearCr CollgCr Crawfor Edwards Gilbert  Greens GrnHill  IDOTRR Landmrk MeadowV Mitchel   NAmes NoRidge NPkVill NridgHt  NWAmes OldTown  Sawyer
      28      10      30     108      44     267     103     194     165       8       2      93       1      37     114     443      71      23     166     131     239     151
SawyerW Somerst StoneBr   SWISU  Timber Veenker
    125     182      51      48      72      24
```

Histogram of prices of the houses

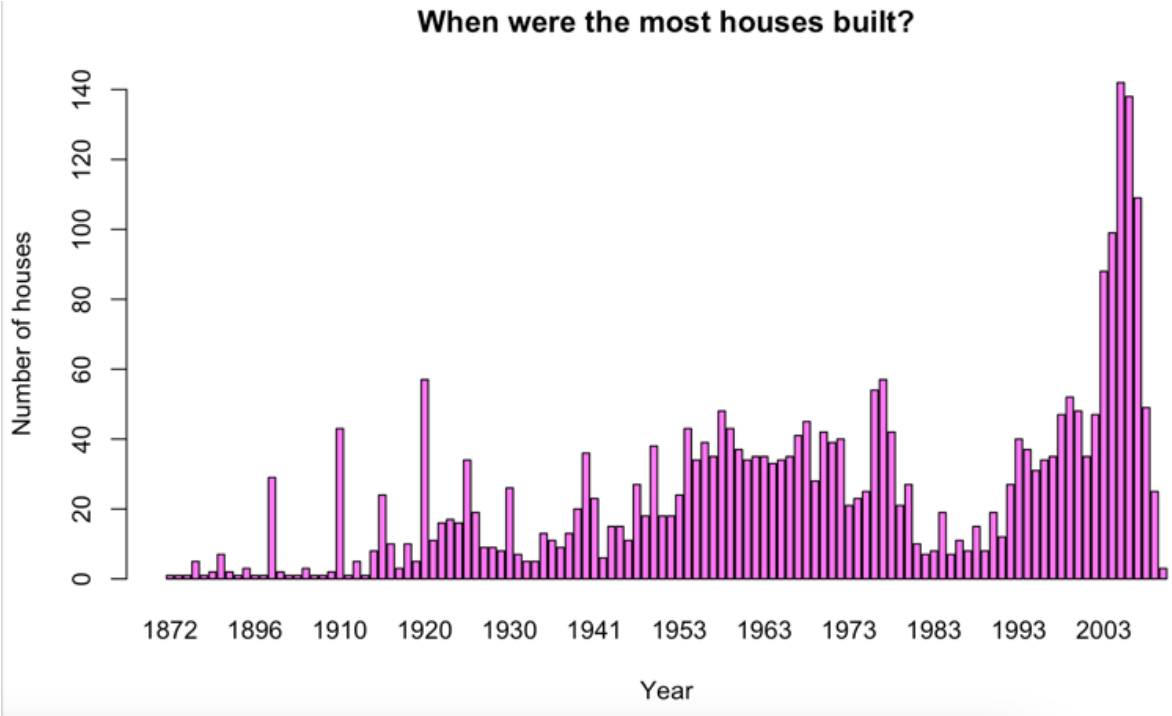
```
> hist(sales$price,
+      col = "red", breaks   = 50)
```



The average price of the house was $180,796, whereas the median was 160, 000. The median price was a little more than the average which tells that some of the values were extreme which is clearly visualized in the histogram plot.

Now let us see the houses-built year in a bar plot.

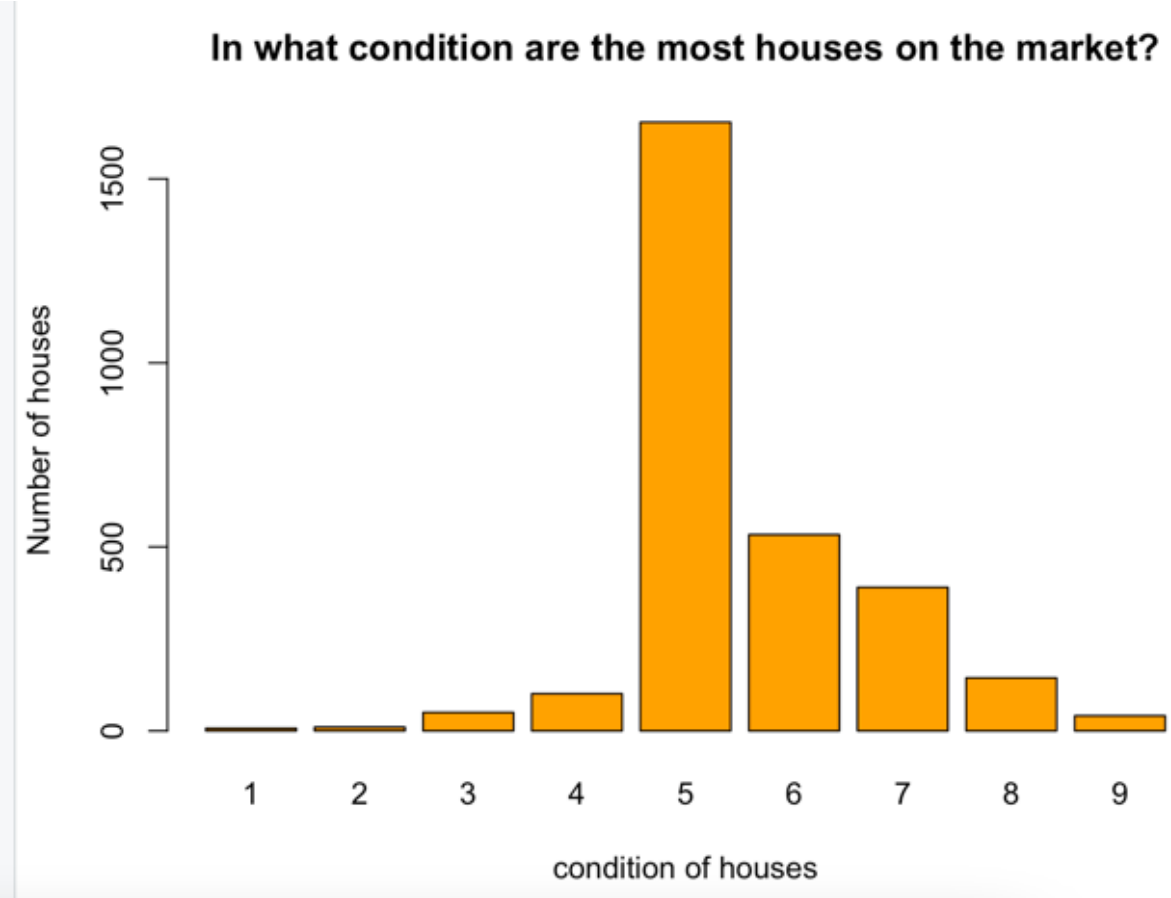
```
> counts <- table(sales$Year.Built)
> barplot(counts, main = "When were the most houses built?",
+         xlab = "Year",
+         ylab = "Number of houses",
+         col ="violet")
```



From the bar plot, it is seen that more houses are built after 2002.

Now let us look at the conditions of the houses,

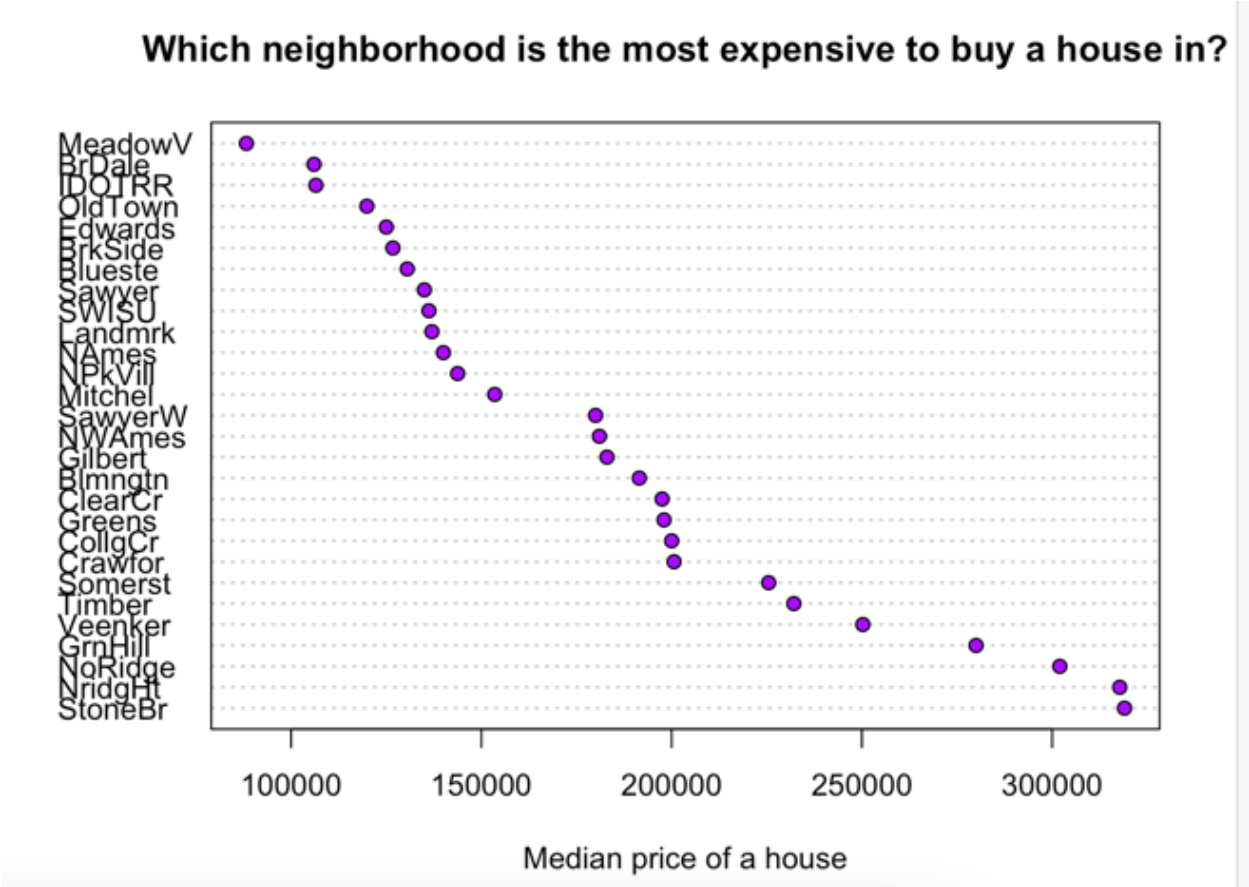
```
> barplot(table(sales$Overall.Cond),
+         main = "In what condition are the most houses on the market?",
+         xlab = "condition of houses",
+         ylab = "Number of houses",
+         col = "Orange")
```



From the bar plot, it is seen that many houses are in normal and good condition.

Now, let us see the weather prices of the houses are determined by the neighborhood. We will create a dot plot to see this relationship.

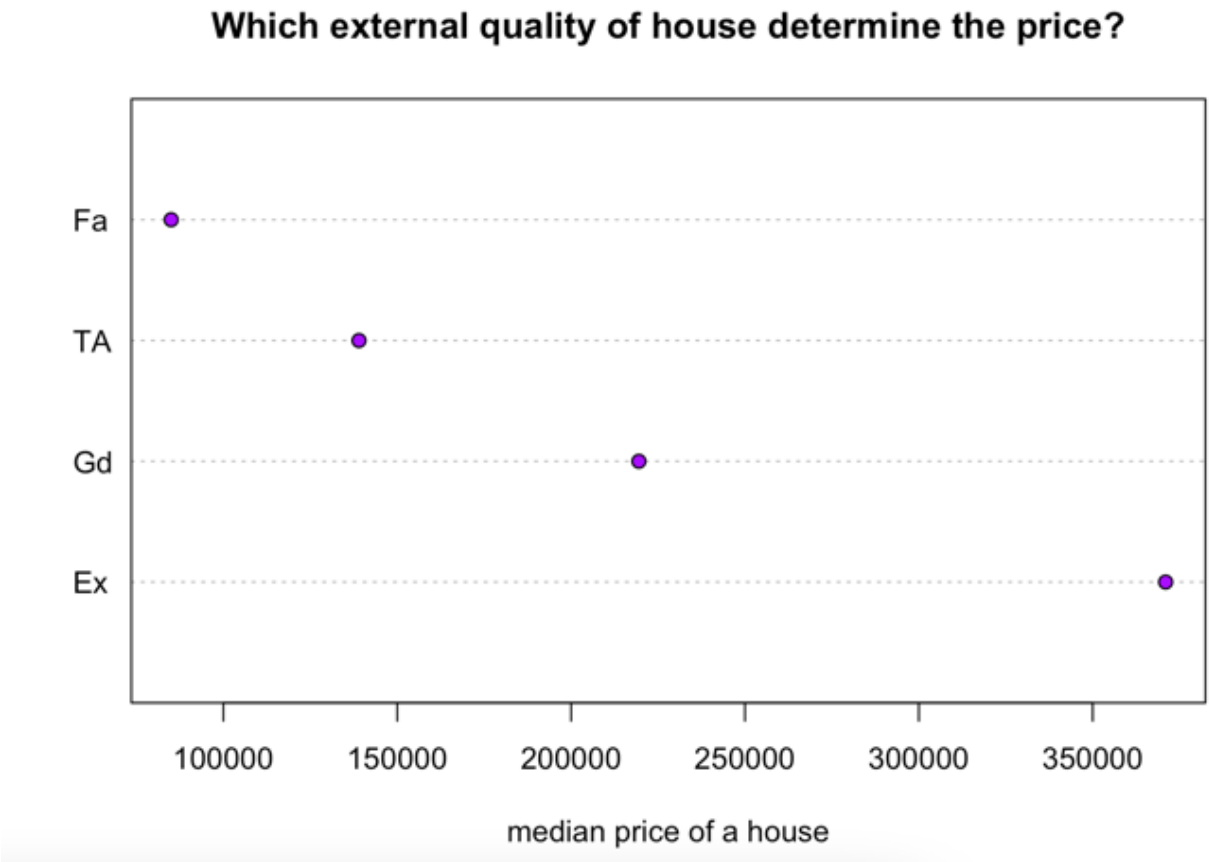
```
> neighbourhoods = tapply(sales$price, sales$Neighborhood, median)
  neighbourhoods = sort(neighbourhoods, decreasing = TRUE)
  dotchart(neighbourhoods,
           bg = "purple1",
           xlab="Median price of a house",
           main = "Which neighborhood is the most expensive to buy a house in?")
```



From the visualizations, it is clear that some of the neighborhood has high prices for houses. While plotting the chart, I am using the median value of house prices as median is less affected by outliers than the mean value. (In the code, tapply () function is used for comparison, the first argument of the tapply() function should be the variable of interest, the second argument should be for the group.

Next, let us see the relationship between external quality of the houses with the price of the houses.

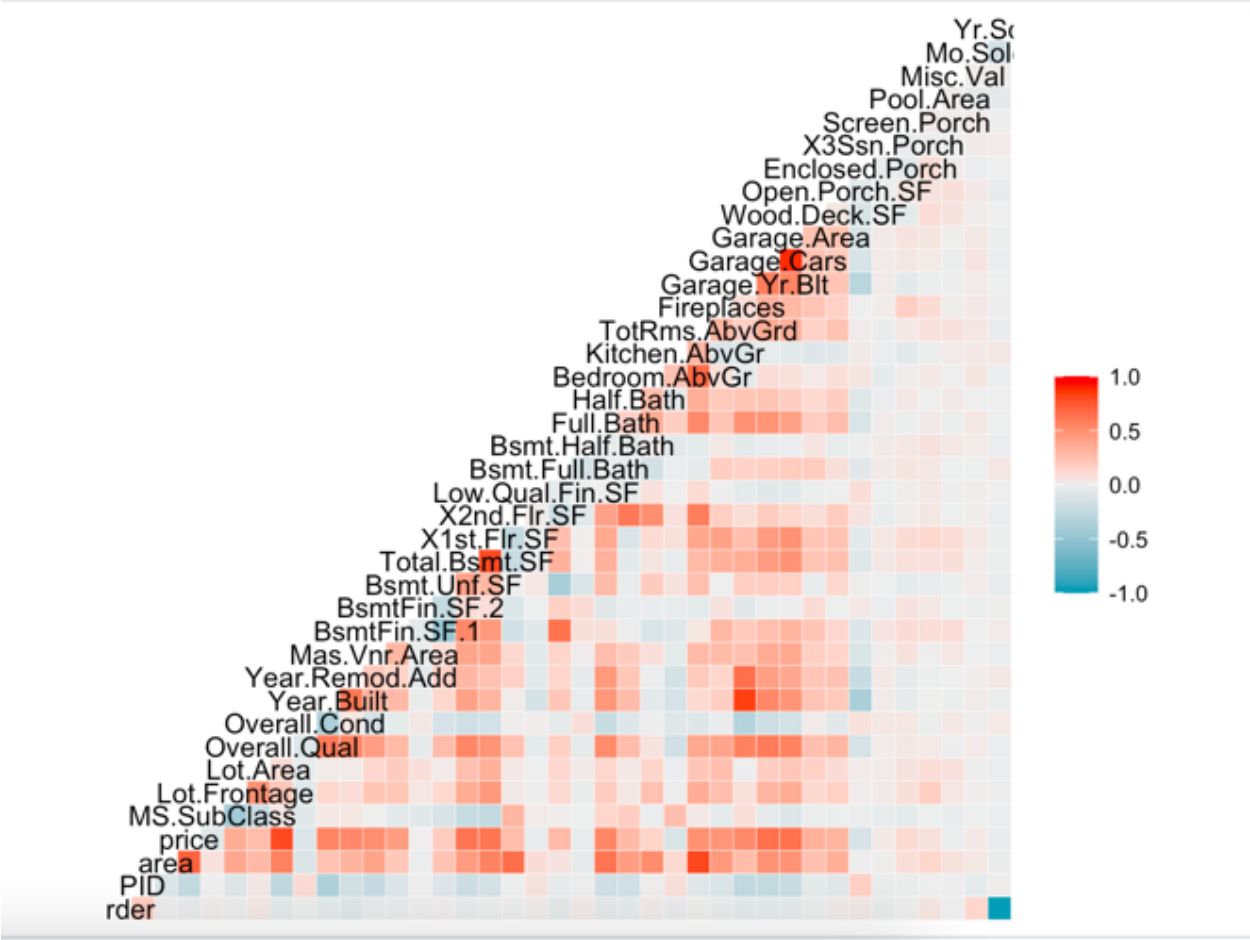
```
> externalQual = tapply(sales$price, sales$Exter.Qual, median)
> externalQual = sort(externalQual, decreasing = TRUE)
> dotchart(externalQual,
+          bg = "purple1",
+          xlab="median price of a house",
+          main = "Which external quality of house determine the price?")
```



From the visualization, it is clear that the median value for prices of the houses is influenced by the external quality of the houses.

Now, selecting numeric variables from all the variables present in the data frame to see the correlations using the heat map as the heat map only takes numeric variables I have used the ggcorr() function to generate the heat map.

```
> numeric = sales %>% select(where(is.numeric))
> ggcorr(numeric)
>
```



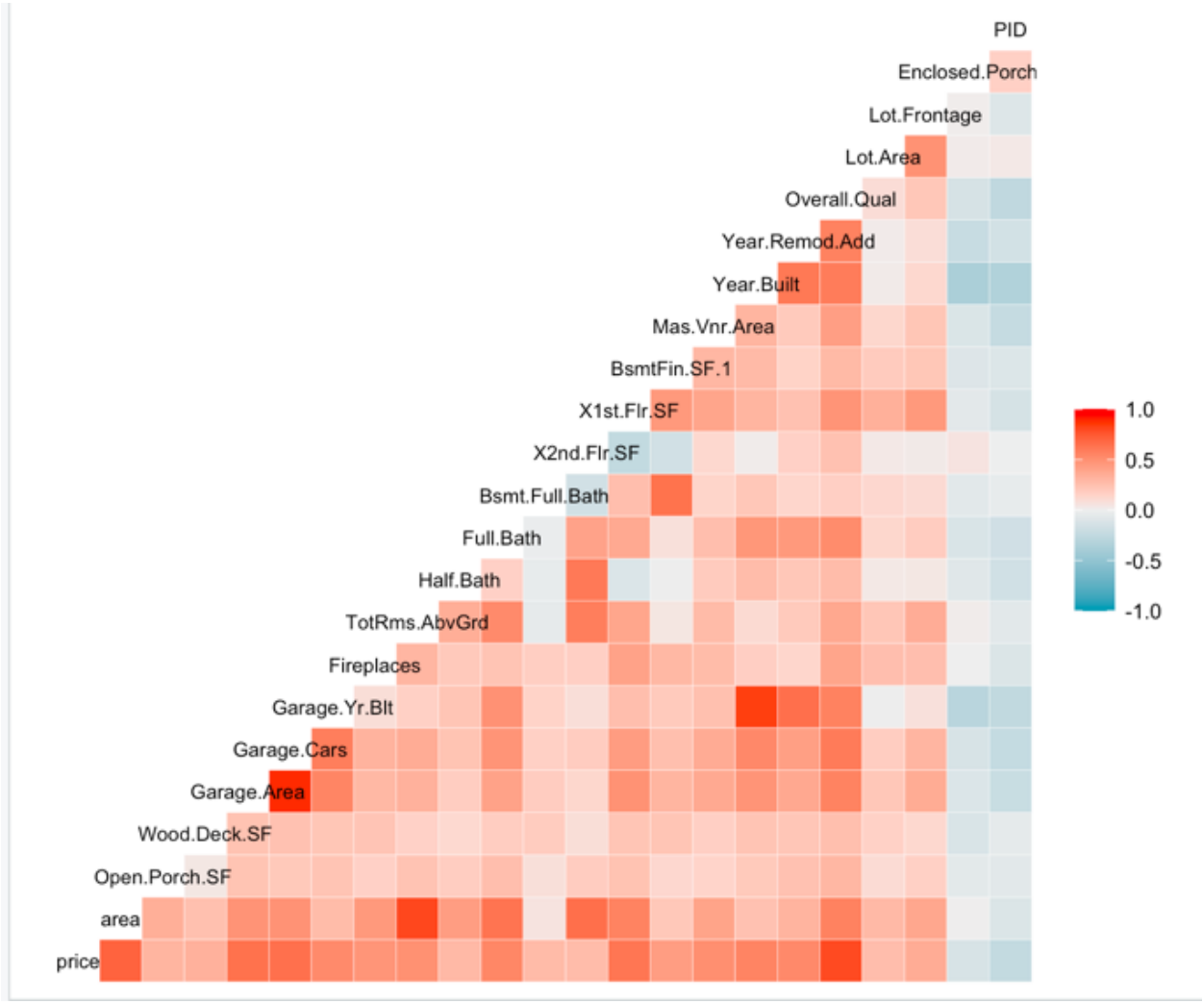
Multicollinearity happens when the two independent variables in the regression model are highly correlated to each other, “year sold” variables and an “order” variables have a strong negative correlation as seen in the above heat map. Using strong negatively correlated independent variables in the model makes it hard to interpret the model and also creates an overfitting problem.

Now for determining the correlation coefficient of the independent variables with the price variable, I am using the cor() function, checking the correlation coefficient of different variables with the target variables.

```
> cor(sales$Enclosed.Porch, sales$price)
[1] -0.1287874
> cor(sales$price, sales$PID)
[1] -0.2465212
> cor(sales$price, sales$Overall.Qual)
[1] 0.7992618
> cor(sales$price, sales$area)
[1] 0.7067799
> cor(sales$price, sales$X1st.Flr.SF) #first floor sq.ft
[1] 0.6216761
> cor(sales$price, sales$Full.Bath)
[1] 0.5456039
> cor(sales$price, sales$TotRms.AbvGrd) #Total rooms above ground
[1] 0.4954744
> cor(sales$price, sales$Year.Built)
[1] 0.5584261
> cor(sales$price, sales$Mas.Vnr.Area)
[1] 0.5057841
```

Now drawing a heat map of prices with the only variables that have a strong and weak relationship, omitting intermediate variables with the intermediate relationship. It will further narrow our variable selection process, let see how it looks in the heat map.

```
> new_sales = numeric %>% select(price, area, Open.Porch.SF, Wood.Deck.SF, Garage.Area,
+                                Garage.Cars, Garage.Yr.Blt, Fireplaces, TotRms.AbvGrd, Half.Bath,
 Full.Bath,
+                                Bsmt.Full.Bath, Garage.Area, X2nd.Flr.SF, X1st.Flr.SF, BsmtFin.SF.
1,
+                                Mas.Vnr.Area, Year.Built, Year.Remod.Add, Overall.Qual, Lot.Area,
+                                Lot.Frontage, Enclosed.Porch, PID)
> ggcorr(new_sales, size = 3)
```



From the above heat map, it is seen that PID and Enclosed Porch do not have a strong positive correlation with the price of the house. Now we use only use those variables which have a strong relationship with the price of the houses while building the regression model. First, let us split our data into training and test data.

- Step 5: Train - Test Split.

```
> set.seed(2)
> sample <- sample.split(sales, SplitRatio = 0.7)
> train  <- subset(sales, sample == TRUE)
> test   <- subset(sales, sample == FALSE)
> dim(train)
[1] 2035   82
> dim(test)
[1] 895  82
>
```

#The set. seed() function in R is used to create reproducible results when writing code that involves creating variables that take on random values. By using the set. seed() function, it guarantees that the same random values are produced each time you run the code., whereas the sample.split() function will split the data into random training and test datasets, I have trained the data using 70% of the data and used the dim() function to observe the dimensions of the training and test dataset.

- Step 6: Separate the test labels from the test data.

```
> test_label <- test[ , 4]
> test_label
  [1] 189900 195500 191500 180400 538000 164000 216000 184000 127500 149900
 [11] 376162 220000 611657 224000 500000 205000 192000 216500 185088 180000
 [21] 355000 260400 221000 143000  99500 122000 133000 169000 190000 149500
 [31] 152000 267916 218500 196500 197500 143000 136300 142125 197600 172500
 [41] 128000 154300 190000 135000 214000 145000 148000 108538  97500 162000
 [51] 155000  80400 109500 119000 129000 100000  76500 209500 132000 139900
 [61] 109500 122000 244400 173000 107500  93369 136500 121500 125000 154000
 [71] 137250 160250 328000 128000 308030 206000 198900 320000 200500 128200
 [81] 127000 143750 155891 125200 107000 113000 128000 160000 100000 169000
 [91] 266500 162500 125500  82000 110000  55993  50138 190000 378500 173500
[101]  85500 130000 149900 231000 345000 189500 278000 178000 178000 174000
[111] 180500 260000  82500 215000 154000 187500 152000 240900 263435 220000
[121] 167900 158000 136000 125000  97000 147000 148500 128500 485000 256300
[131] 253293 610000 335000 350000 280000 233170 255900 212500 230000 552000
[141] 248500 173000 167800 174000 174000 192500 181000 188500 244000 179000
[151] 327000 340000 265000 402000 275000 257500 252678 250000 270000 252000
[161] 291000 209000 193000 203000 184900 153000 189000 120000 145000 184000
[171]  82000  76000 141000 173733 195000 150000 154000 185750 197900 230000
[181] 167900 168500 140000 153000 145100 158000  82500 167000 128900 140000
[191] 187500 193500 104900 150000 156500 139000 155000 144000 102900 152500
[201] 142900 156500 105000 163000 135000 153000  90000 132500 145000 127000
[211] 110000 130000  94550 124500 135000 129500  37900  99500 113000  87500
[221] 265979 160000  58500 137000 128000  60000 105000 150000 155000  62500
[231] 149000 116000 103600 172500 113500 134900 148000 143000 135000  82500
[241] 122000 154400 125000  84000 139500 108000 162000 271900 148325 269500
[251] 272500 239000 200000 275000 152000 143000 197900 230000 124000 140000
[261] 136500 133900 236500 261500 313000 220000 219500 178000 213000 144000
[271] 190000 190000 108000 120000  99900 159434  60000 197000 155000 137000
[281] 234000 154900 158500 121000 124000 222000  85000 251000 239686 240000
[291] 173000 137500 315500 224500 410000 175000 204000 170000  80000  88000
[301] 173000 176500 206900 173000 286000 247900 194500 387000 215000 226500
[311] 235000 175000 180000 159900 205000 133000 111250 103400 100000 100500
[321]  89500 111750 148500 460000 250000 367294 192000 266000 154000 219990
[331] 191000 176000 184000 284500 315000 350000 341000 235128 214000 232000
[341] 245000 247000 250580 182000 226700 339750 205950 207500 193500 190500
[351] 115000 119916 192000 171900 146000 146000 305000 200000 324000 162500
[361] 205000 145000 141000 147000 135000 135000 173000 176000  98000 109008
[371] 180500 174900 179900 141500 141500 135000 124000 136500 166000 116000
[381] 137500 118500 139000 186000 157000 116000 159000 112500 105500 127500
[391] 109500 143000 157500 122600 111000 139500 118000 122250  65000 139500
[401] 163000 128000 116900 131500 138000 108000 164900 115000 162900 150000
[411] 128000 184000 160000 161000 127500 141000  89500  79900  85000  82375
```

#test_label variable was created using all the rows and price variables of the dataset.

- Step 7: Train the model

```
> model = lm(price ~  Overall.Qual  + area + X1st.Flr.SF + Garage.Area + Mas.Vnr.Area + Year.Built + Exter.Qual+ Full.Bath , data = train)
> summary(model)

Call:
lm(formula = price ~ Overall.Qual + area + X1st.Flr.SF + Garage.Area +
    Mas.Vnr.Area + Year.Built + Exter.Qual + Full.Bath, data = train)

Residuals:
    Min      1Q  Median      3Q     Max
-492562  -17865   -1146   15018  287928

Coefficients:
               Estimate  Std. Error t value             Pr(>|t|)
(Intercept)  -706499.549   69379.146 -10.183 < 0.0000000000000002 ***
Overall.Qual   17097.329     902.104  18.953 < 0.0000000000000002 ***
area              54.773       2.367  23.139 < 0.0000000000000002 ***
X1st.Flr.SF       25.143       2.446  10.280 < 0.0000000000000002 ***
Garage.Area       44.411       4.535   9.793 < 0.0000000000000002 ***
Mas.Vnr.Area      25.161       5.056   4.976   0.0000007040321412 ***
Year.Built       369.250      35.761  10.326 < 0.0000000000000002 ***
Exter.QualFa  -75704.145    9664.253  -7.833   0.0000000000000076 ***
Exter.QualGd  -58215.849    4574.304 -12.727 < 0.0000000000000002 ***
Exter.QualTA  -71643.274    5120.488 -13.991 < 0.0000000000000002 ***
Full.Bath      -9365.965    1909.217  -4.906   0.0000010054741250 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 33000 on 2024 degrees of freedom
Multiple R-squared:  0.8267,    Adjusted R-squared:  0.8258
F-statistic: 965.4 on 10 and 2024 DF,  p-value: < 0.00000000000000022

>
```

The regression model was created , lm() is a function used to create the regression model using the variables overall quality, first-floor sqft, Masonry veneer area in square feet, garage area, year build, and full bath as shown in the screenshot.

Here, the R-squared value for the model is 83%, which states that our model explains 83% of the variations in the dependent variable(price). The adjusted R-square value corrects R-squared by penalizing models with a large number of independent variables.

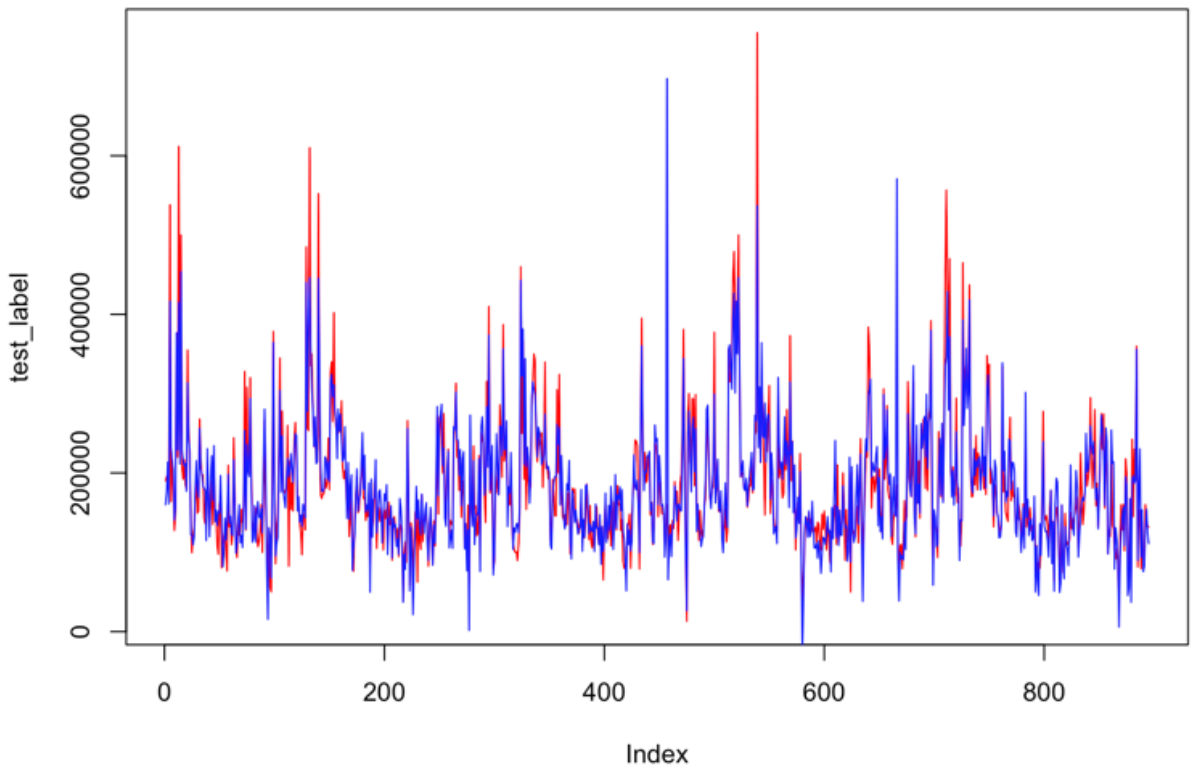
- Step 8: Make predictions.

```
> pred <- predict(model, test)
> pred
        5         6         8        13        16        17        23        29        33        34
159967.744 175984.998 213641.198 161004.170 416401.948 232366.748 216334.513 203905.854 140687.076 155765.189
       37        41        45        46        47        50        54        56        57        58
376643.481 229043.065 414927.203 211298.056 453801.205 229504.657 214220.460 209457.423 211658.273 176856.687
       61        62        65        76        78        87        88        90        95        98
313607.117 242620.626 227106.910 122310.825 121541.998 109431.715 119606.403 180766.206 215463.026 178341.429
       99       105       111       115       116       119       123       127       128       129
169019.895 256048.224 204968.649 198480.767 183621.419 180837.309 177891.777 114822.986 230543.538 165652.339
      132       136       138       139       140       143       144       147       158       160
119645.380 195078.452 221632.221 165368.660 234382.906 138665.977 153246.717 133862.637 151818.855 149770.681
      169       170       172       177       180       181       187       193       197       198
196634.107 142604.396  81738.836  96630.915 143201.180 115770.824 145807.973 198296.337 136864.518 145400.180
      201       205       209       210       211       214       218       220       221       222
123283.140 121192.991 216617.472 164305.638  99094.601 111916.827 139489.189 159820.505 112097.737 121363.341
      225       226       229       240       242       251       252       254       259       262
106144.026 164325.326 253551.122 129407.832 235754.367 202159.111 195470.451 293823.858 181959.114 112739.000
      263       269       275       279       280       283       287       291       292       293
117639.870 158808.762 112935.733 147564.084 163823.100 144548.054 143137.701 155284.545 124728.822 205968.681
      296       300       302       303       304       307       308       311       322       324
280393.722 206531.250  84459.202  15436.800 131073.710 120508.012  67652.933 159414.376 364498.626 170580.608
      333       334       336       341       344       345       351       357       361       362
 95283.062 111248.528 119123.028 209800.356 303776.592 175809.333 246572.271 178065.354 166099.666 155995.583
      365       369       373       374       375       378       382       384       385       386
195057.778 218350.310 194992.832 203832.268 224416.728 203332.083 187957.114 230381.657 249341.440 247190.371
      389       390       393       404       406       415       416       418       423       426
152175.032 169886.550 138192.281 146790.566 132924.725 160629.778 156051.346 143941.772 440205.111 277909.616
      427       433       439       443       444       447       451       455       456       457
385619.183 446237.681 357705.936 307189.181 279372.273 233434.154 270709.737 211767.272 214305.757 445381.663
      460       464       466       467       468       471       472       475       486       488
322973.432 207829.384 177234.986 176742.098 179890.683 220348.196 217471.607 191833.545 214162.743 189084.012
      497       498       500       505       508       509       515       521       525       526
315612.265 324091.211 296406.300 310951.463 279528.157 266924.928 218102.924 280503.917 252992.772 274174.137
      529       533       537       538       539       542       546       548       549       550
241345.335 214687.360 218587.251 258310.369 198952.326 142826.349 213106.729 141378.559 184462.441 197809.832
      553       554       557       568       570       579       580       582       587       590
 77123.025 105631.003 123633.078 168750.558 205138.538 152377.127 167414.400 171905.353 209514.303 250802.125
      591       597       603       607       608       611       615       619       620       621
179360.639 222341.411 150481.154 159556.841 153152.637 187298.749  49744.917 192153.630 119153.139 166467.919
```

Prediction of the test sample was done by using predict() function, where the model was assigned as the first argument of the predict function while test data as the second argument.

- Step 9: Compare the predicted and actual values.

```
> plot(test_label, type = "l", lty=1.8,  col="red") #red color for test price
> lines(pred, type="l", col="blue")#blue color for predicted price
```



By visualizing the line graphs of the prices of both the test data(red) and the predicted price(blue), the lines of predicted value fairly overlap the test data price, which concludes that our model is performing well in the test dataset.

Now let us see the Root Mean Square Error (RMSE) to see the accuracy of the model we have created.

```
> #Accuracy of the model(root mean square error)
> sqrt(mean((pred-test_label)^2))
[1] 38982.75
>
```

The Root mean square error for the model was found to be 38982.75.

VI. Conclusion

The weighted average error between the predicted price and the actual price in the dataset was 38982.75 which was likely a good value given that the average actual house price of the dataset was 180,796. RMSE value was quite small, which means the model has enough good to predict the future dataset. Thus, while forecasting the numeric dataset, the regression method was the best option.

Some of the struggles and difficulties I faced while running R-studio on multiple linear regression models were -understanding the syntax and language of the code was difficult for me. For a large dataset, R-studio was more time-consuming, and understanding the statistical significance was hard for me as the output of regression was difficult to interpret without an appropriate statistical background. For the regression model, troubleshooting error or bugs were difficult without appropriate technical knowledge.

For the multiple linear regression model, other tools that work properly are Stata, MATLAB, SAS, and Python. Stata There is a Stata, a software package that analyzes and manage the data for a wide range of data visualization and statistical procedure. Likewise, SAS is another tool that offers extensive data management and management functions. Similarly, SPSS is another statistical software package that helps to model the data and visualize it. Whereas Python is a popular high-level language that can be used for machine learning purposes and have a high range of available libraries to work on statistical data. There is another powerful language that can be used for data analysis and numerical computation known as MATLAB which has a wide range of tools for data analysis and visualization and also it would be helpful in building regression models.

VII. References.

1. https://www.kaggle.com/datasets/marcopale/housing.
2. https://www.youtube.com/watch?v=6dEUTmoXz0w&ab_channel=Simplilearn
3. https://rpubs.com/Zetrosoft/lbb-rm