Predicting House sales pricing with linear regression

Vibhore Singh

2024-05-01

#Title: "Predicting House sales pricing with linear regression"

Abstract

Predicting real estate prices is one of the most challenging task in data science. House price fluctuations in the real estate market occur due to the effects of several reasons. Our study tried to develop a linear regression model using the comprehenside data set of Ames housing sales based on 81 features.

Our approach involves rigorous exploratory data analysis to identify significant predictors and manage missing values, followed by feature engineering to enhance model inputs. We employ linear regression techniques, acknowledging its advantages in interpretability and implementation for real estate valuation. The initial phase of the project involved a thorough exploratory data analysis (EDA) aimed at understanding the distribution of the data, handling missing values, and identifying potential outliers. Significant features impacting house prices were highlighted through correlation analysis and visual inspections. The linear regression model was selected for its simplicity and interpretability. Trained the model on several feature subsets and evaluated using r squared and rmse values with the final model showing a robust fit to the data. This study could be a great help to Homeowners , Realestate agents, Policy makers, etc by helping them understand how specific featues can impactt the valuation of the property.

Dataset

-The Ames Housing dataset was compiled by Dean De Cock for use in data science education. It contains 2930 observations and a large number of explanatory variables (23 nominal, 23 ordinal, 14 discrete, and 20 continuous) involved in assessing home prices in Ames, Iowa from 2006 to 2010.

- The dataset includes a wide range of features that describe almost every aspect of residential homes. It includes features such as living area, basement, garage, pool. neighborhood.
- The author has tried to give a dataset suitable for deeper analysis and advanced regression techniques.

Domain Expertise for Analysis

- Understanding of Seasons : We used the Season variable from the Months variable to analyse the buying and seeling habbits for different proprties. We found out summer is the best season for real estate agents as the prices are high and the count is high as well .

- Econonomic Factors like the financial crisis of 2008 affected the sales of houses but the market came back on time. This gives real estate agents an idea about keeping an eye on the global issues as they can can influence economy and furter their business as well. so they can decide which is the time to see and when the market is down they can prepare for the better time.

- We found out additional things like Basements, Fireplace are really appreciated bt the buyrs.

Loading Libraries

Data Loading

```r
train <- read_csv("train.csv")
```

```
## Rows: 1460 Columns: 81
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (43): MSZoning, Street, Alley, LotShape, LandContour, Utilities, LotConf...
## dbl (38): Id, MSSubClass, LotFrontage, LotArea, OverallQual, OverallCond, Ye...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
test <- read.csv("test.csv")
```

Data Structure Overview

```r
summary(train)
```

```
##        Id           MSSubClass      MSZoning          LotFrontage
##  Min.   :   1.0   Min.   : 20.0   Length:1460        Min.   : 21.00
##  1st Qu.: 365.8   1st Qu.: 20.0   Class :character   1st Qu.: 59.00
##  Median : 730.5   Median : 50.0   Mode  :character   Median : 69.00
##  Mean   : 730.5   Mean   : 56.9                      Mean   : 70.05
##  3rd Qu.:1095.2   3rd Qu.: 70.0                      3rd Qu.: 80.00
##  Max.   :1460.0   Max.   :190.0                      Max.   :313.00
##                                                      NA's   :259
##     LotArea          Street             Alley             LotShape
##  Min.   :  1300   Length:1460        Length:1460        Length:1460
##  1st Qu.:  7554   Class :character   Class :character   Class :character
##  Median :  9478   Mode  :character   Mode  :character   Mode  :character
##  Mean   : 10517
##  3rd Qu.: 11602
##  Max.   :215245
##
##  LandContour         Utilities          LotConfig          LandSlope
##  Length:1460        Length:1460        Length:1460        Length:1460
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  Neighborhood        Condition1         Condition2          BldgType
##  Length:1460        Length:1460        Length:1460        Length:1460
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
```

```
##   HouseStyle         OverallQual      OverallCond      YearBuilt
##  Length:1460        Min.   : 1.000   Min.   :1.000   Min.   :1872
##  Class :character   1st Qu.: 5.000   1st Qu.:5.000   1st Qu.:1954
##  Mode  :character   Median : 6.000   Median :5.000   Median :1973
##                     Mean   : 6.099   Mean   :5.575   Mean   :1971
##                     3rd Qu.: 7.000   3rd Qu.:6.000   3rd Qu.:2000
##                     Max.   :10.000   Max.   :9.000   Max.   :2010
##
##   YearRemodAdd   RoofStyle           RoofMatl          Exterior1st
##  Min.   :1950   Length:1460        Length:1460        Length:1460
##  1st Qu.:1967   Class :character   Class :character   Class :character
##  Median :1994   Mode  :character   Mode  :character   Mode  :character
##  Mean   :1985
##  3rd Qu.:2004
##  Max.   :2010
##
##  Exterior2nd         MasVnrType          MasVnrArea       ExterQual
##  Length:1460        Length:1460        Min.   :   0.0   Length:1460
##  Class :character   Class :character   1st Qu.:   0.0   Class :character
##  Mode  :character   Mode  :character   Median :   0.0   Mode  :character
##                                        Mean   : 103.7
##                                        3rd Qu.: 166.0
##                                        Max.   :1600.0
##                                        NA's   :8
##   ExterCond          Foundation          BsmtQual           BsmtCond
##  Length:1460        Length:1460        Length:1460        Length:1460
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  BsmtExposure       BsmtFinType1         BsmtFinSF1     BsmtFinType2
##  Length:1460        Length:1460        Min.   :   0.0   Length:1460
##  Class :character   Class :character   1st Qu.:   0.0   Class :character
##  Mode  :character   Mode  :character   Median : 383.5   Mode  :character
##                                        Mean   : 443.6
##                                        3rd Qu.: 712.2
##                                        Max.   :5644.0
##
##    BsmtFinSF2        BsmtUnfSF       TotalBsmtSF       Heating
##  Min.   :   0.00   Min.   :   0.0   Min.   :   0.0   Length:1460
##  1st Qu.:   0.00   1st Qu.: 223.0   1st Qu.: 795.8   Class :character
##  Median :   0.00   Median : 477.5   Median : 991.5   Mode  :character
##  Mean   :  46.55   Mean   : 567.2   Mean   :1057.4
##  3rd Qu.:   0.00   3rd Qu.: 808.0   3rd Qu.:1298.2
##  Max.   :1474.00   Max.   :2336.0   Max.   :6110.0
##
##   HeatingQC          CentralAir         Electrical           1stFlrSF
##  Length:1460        Length:1460        Length:1460        Min.   : 334
##  Class :character   Class :character   Class :character   1st Qu.: 882
##  Mode  :character   Mode  :character   Mode  :character   Median :1087
##                                                           Mean   :1163
##                                                           3rd Qu.:1391
```

```
##                                                          Max.   :4692
##
##     2ndFlrSF     LowQualFinSF       GrLivArea     BsmtFullBath
##  Min.   :   0   Min.   :  0.000   Min.   : 334   Min.   :0.0000
##  1st Qu.:   0   1st Qu.:  0.000   1st Qu.:1130   1st Qu.:0.0000
##  Median :   0   Median :  0.000   Median :1464   Median :0.0000
##  Mean   : 347   Mean   :  5.845   Mean   :1515   Mean   :0.4253
##  3rd Qu.: 728   3rd Qu.:  0.000   3rd Qu.:1777   3rd Qu.:1.0000
##  Max.   :2065   Max.   :572.000   Max.   :5642   Max.   :3.0000
##
##   BsmtHalfBath        FullBath        HalfBath       BedroomAbvGr
##  Min.   :0.00000   Min.   :0.000   Min.   :0.0000   Min.   :0.000
##  1st Qu.:0.00000   1st Qu.:1.000   1st Qu.:0.0000   1st Qu.:2.000
##  Median :0.00000   Median :2.000   Median :0.0000   Median :3.000
##  Mean   :0.05753   Mean   :1.565   Mean   :0.3829   Mean   :2.866
##  3rd Qu.:0.00000   3rd Qu.:2.000   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :2.00000   Max.   :3.000   Max.   :2.0000   Max.   :8.000
##
##   KitchenAbvGr   KitchenQual         TotRmsAbvGrd     Functional
##  Min.   :0.000   Length:1460        Min.   : 2.000   Length:1460
##  1st Qu.:1.000   Class :character   1st Qu.: 5.000   Class :character
##  Median :1.000   Mode  :character   Median : 6.000   Mode  :character
##  Mean   :1.047                      Mean   : 6.518
##  3rd Qu.:1.000                      3rd Qu.: 7.000
##  Max.   :3.000                      Max.   :14.000
##
##    Fireplaces    FireplaceQu         GarageType         GarageYrBlt
##  Min.   :0.000   Length:1460        Length:1460        Min.   :1900
##  1st Qu.:0.000   Class :character   Class :character   1st Qu.:1961
##  Median :1.000   Mode  :character   Mode  :character   Median :1980
##  Mean   :0.613                                         Mean   :1979
##  3rd Qu.:1.000                                         3rd Qu.:2002
##  Max.   :3.000                                         Max.   :2010
##                                                        NA's   :81
##  GarageFinish         GarageCars      GarageArea      GarageQual
##  Length:1460        Min.   :0.000   Min.   :   0.0   Length:1460
##  Class :character   1st Qu.:1.000   1st Qu.: 334.5   Class :character
##  Mode  :character   Median :2.000   Median : 480.0   Mode  :character
##                     Mean   :1.767   Mean   : 473.0
##                     3rd Qu.:2.000   3rd Qu.: 576.0
##                     Max.   :4.000   Max.   :1418.0
##
##   GarageCond         PavedDrive          WoodDeckSF      OpenPorchSF
##  Length:1460        Length:1460        Min.   :  0.00   Min.   :  0.00
##  Class :character   Class :character   1st Qu.:  0.00   1st Qu.:  0.00
##  Mode  :character   Mode  :character   Median :  0.00   Median : 25.00
##                                        Mean   : 94.24   Mean   : 46.66
##                                        3rd Qu.:168.00   3rd Qu.: 68.00
##                                        Max.   :857.00   Max.   :547.00
##
##  EnclosedPorch      3SsnPorch       ScreenPorch        PoolArea
##  Min.   :  0.00   Min.   :  0.00   Min.   :  0.00   Min.   :  0.000
##  1st Qu.:  0.00   1st Qu.:  0.00   1st Qu.:  0.00   1st Qu.:  0.000
##  Median :  0.00   Median :  0.00   Median :  0.00   Median :  0.000
```

```
##  Mean   : 21.95   Mean   :  3.41   Mean   : 15.06   Mean   :  2.759
##  3rd Qu.:  0.00   3rd Qu.:  0.00   3rd Qu.:  0.00   3rd Qu.:  0.000
##  Max.   :552.00   Max.   :508.00   Max.   :480.00   Max.   :738.000
##
##     PoolQC             Fence           MiscFeature           MiscVal
##  Length:1460        Length:1460        Length:1460        Min.   :    0.00
##  Class :character   Class :character   Class :character   1st Qu.:    0.00
##  Mode  :character   Mode  :character   Mode  :character   Median :    0.00
##                                                           Mean   :   43.49
##                                                           3rd Qu.:    0.00
##                                                           Max.   :15500.00
##
##      MoSold           YrSold       SaleType         SaleCondition
##  Min.   : 1.000   Min.   :2006   Length:1460        Length:1460
##  1st Qu.: 5.000   1st Qu.:2007   Class :character   Class :character
##  Median : 6.000   Median :2008   Mode  :character   Mode  :character
##  Mean   : 6.322   Mean   :2008
##  3rd Qu.: 8.000   3rd Qu.:2009
##  Max.   :12.000   Max.   :2010
##
##    SalePrice
##  Min.   : 34900
##  1st Qu.:129975
##  Median :163000
##  Mean   :180921
##  3rd Qu.:214000
##  Max.   :755000
##
```

Some observations from the summary statistics

- 'LotFrontage' shows a mean greater than the median (70.05 vs. 69.00), suggesting a right skew in the distribution
- OverallQual With a mean and median close to 6 (6.099 vs. 6.000), most houses have above average quality. The range from 1 to 10 also suggests significant variability in house overall quality
- GrLivArea has the mean and median values 1515 vs. 1464 that are close, but the max value (5642) is very high, indicating potential outliers or luxury homes with large living areas.
- The target variable SalePrice has a wide range from $34,900 to $755,000, with a mean significantly higher than the median ($180,921 vs. $163,000), suggesting a right-skewed distribution.

```r
str(train)
```

```
## spc_tbl_ [1,460 x 81] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Id           : num [1:1460] 1 2 3 4 5 6 7 8 9 10 ...
##  $ MSSubClass   : num [1:1460] 60 20 60 70 60 50 20 60 50 190 ...
##  $ MSZoning     : chr [1:1460] "RL" "RL" "RL" "RL" ...
##  $ LotFrontage  : num [1:1460] 65 80 68 60 84 85 75 NA 51 50 ...
##  $ LotArea      : num [1:1460] 8450 9600 11250 9550 14260 ...
##  $ Street       : chr [1:1460] "Pave" "Pave" "Pave" "Pave" ...
##  $ Alley        : chr [1:1460] NA NA NA NA ...
##  $ LotShape     : chr [1:1460] "Reg" "Reg" "IR1" "IR1" ...
##  $ LandContour  : chr [1:1460] "Lvl" "Lvl" "Lvl" "Lvl" ...
##  $ Utilities    : chr [1:1460] "AllPub" "AllPub" "AllPub" "AllPub" ...
##  $ LotConfig    : chr [1:1460] "Inside" "FR2" "Inside" "Corner" ...
##  $ LandSlope    : chr [1:1460] "Gtl" "Gtl" "Gtl" "Gtl" ...
```

```
##  $ Neighborhood : chr [1:1460] "CollgCr" "Veenker" "CollgCr" "Crawfor" ...
##  $ Condition1   : chr [1:1460] "Norm" "Feedr" "Norm" "Norm" ...
##  $ Condition2   : chr [1:1460] "Norm" "Norm" "Norm" "Norm" ...
##  $ BldgType     : chr [1:1460] "1Fam" "1Fam" "1Fam" "1Fam" ...
##  $ HouseStyle   : chr [1:1460] "2Story" "1Story" "2Story" "2Story" ...
##  $ OverallQual  : num [1:1460] 7 6 7 7 8 5 8 7 7 5 ...
##  $ OverallCond  : num [1:1460] 5 8 5 5 5 5 5 6 5 6 ...
##  $ YearBuilt    : num [1:1460] 2003 1976 2001 1915 2000 ...
##  $ YearRemodAdd : num [1:1460] 2003 1976 2002 1970 2000 ...
##  $ RoofStyle    : chr [1:1460] "Gable" "Gable" "Gable" "Gable" ...
##  $ RoofMatl     : chr [1:1460] "CompShg" "CompShg" "CompShg" "CompShg" ...
##  $ Exterior1st  : chr [1:1460] "VinylSd" "MetalSd" "VinylSd" "Wd Sdng" ...
##  $ Exterior2nd  : chr [1:1460] "VinylSd" "MetalSd" "VinylSd" "Wd Shng" ...
##  $ MasVnrType   : chr [1:1460] "BrkFace" "None" "BrkFace" "None" ...
##  $ MasVnrArea   : num [1:1460] 196 0 162 0 350 0 186 240 0 0 ...
##  $ ExterQual    : chr [1:1460] "Gd" "TA" "Gd" "TA" ...
##  $ ExterCond    : chr [1:1460] "TA" "TA" "TA" "TA" ...
##  $ Foundation   : chr [1:1460] "PConc" "CBlock" "PConc" "BrkTil" ...
##  $ BsmtQual     : chr [1:1460] "Gd" "Gd" "Gd" "TA" ...
##  $ BsmtCond     : chr [1:1460] "TA" "TA" "TA" "Gd" ...
##  $ BsmtExposure : chr [1:1460] "No" "Gd" "Mn" "No" ...
##  $ BsmtFinType1 : chr [1:1460] "GLQ" "ALQ" "GLQ" "ALQ" ...
##  $ BsmtFinSF1   : num [1:1460] 706 978 486 216 655 ...
##  $ BsmtFinType2 : chr [1:1460] "Unf" "Unf" "Unf" "Unf" ...
##  $ BsmtFinSF2   : num [1:1460] 0 0 0 0 0 0 0 32 0 0 ...
##  $ BsmtUnfSF    : num [1:1460] 150 284 434 540 490 64 317 216 952 140 ...
##  $ TotalBsmtSF  : num [1:1460] 856 1262 920 756 1145 ...
##  $ Heating      : chr [1:1460] "GasA" "GasA" "GasA" "GasA" ...
##  $ HeatingQC    : chr [1:1460] "Ex" "Ex" "Ex" "Gd" ...
##  $ CentralAir   : chr [1:1460] "Y" "Y" "Y" "Y" ...
##  $ Electrical   : chr [1:1460] "SBrkr" "SBrkr" "SBrkr" "SBrkr" ...
##  $ 1stFlrSF     : num [1:1460] 856 1262 920 961 1145 ...
##  $ 2ndFlrSF     : num [1:1460] 854 0 866 756 1053 ...
##  $ LowQualFinSF : num [1:1460] 0 0 0 0 0 0 0 0 0 0 ...
##  $ GrLivArea    : num [1:1460] 1710 1262 1786 1717 2198 ...
##  $ BsmtFullBath : num [1:1460] 1 0 1 1 1 1 1 1 0 1 ...
##  $ BsmtHalfBath : num [1:1460] 0 1 0 0 0 0 0 0 0 0 ...
##  $ FullBath     : num [1:1460] 2 2 2 1 2 1 2 2 2 1 ...
##  $ HalfBath     : num [1:1460] 1 0 1 0 1 1 0 1 0 0 ...
##  $ BedroomAbvGr : num [1:1460] 3 3 3 3 4 1 3 3 2 2 ...
##  $ KitchenAbvGr : num [1:1460] 1 1 1 1 1 1 1 1 2 2 ...
##  $ KitchenQual  : chr [1:1460] "Gd" "TA" "Gd" "Gd" ...
##  $ TotRmsAbvGrd : num [1:1460] 8 6 6 7 9 5 7 7 8 5 ...
##  $ Functional   : chr [1:1460] "Typ" "Typ" "Typ" "Typ" ...
##  $ Fireplaces   : num [1:1460] 0 1 1 1 1 0 1 2 2 2 ...
##  $ FireplaceQu  : chr [1:1460] NA "TA" "TA" "Gd" ...
##  $ GarageType   : chr [1:1460] "Attchd" "Attchd" "Attchd" "Detchd" ...
##  $ GarageYrBlt  : num [1:1460] 2003 1976 2001 1998 2000 ...
##  $ GarageFinish : chr [1:1460] "RFn" "RFn" "RFn" "Unf" ...
##  $ GarageCars   : num [1:1460] 2 2 2 3 3 2 2 2 2 1 ...
##  $ GarageArea   : num [1:1460] 548 460 608 642 836 480 636 484 468 205 ...
##  $ GarageQual   : chr [1:1460] "TA" "TA" "TA" "TA" ...
##  $ GarageCond   : chr [1:1460] "TA" "TA" "TA" "TA" ...
##  $ PavedDrive   : chr [1:1460] "Y" "Y" "Y" "Y" ...
```

```
##  $ WoodDeckSF   : num [1:1460] 0 298 0 0 192 40 255 235 90 0 ...
##  $ OpenPorchSF  : num [1:1460] 61 0 42 35 84 30 57 204 0 4 ...
##  $ EnclosedPorch: num [1:1460] 0 0 0 272 0 0 0 228 205 0 ...
##  $ 3SsnPorch    : num [1:1460] 0 0 0 0 0 320 0 0 0 0 ...
##  $ ScreenPorch  : num [1:1460] 0 0 0 0 0 0 0 0 0 0 ...
##  $ PoolArea     : num [1:1460] 0 0 0 0 0 0 0 0 0 0 ...
##  $ PoolQC       : chr [1:1460] NA NA NA NA ...
##  $ Fence        : chr [1:1460] NA NA NA NA ...
##  $ MiscFeature  : chr [1:1460] NA NA NA NA ...
##  $ MiscVal      : num [1:1460] 0 0 0 0 0 700 0 350 0 0 ...
##  $ MoSold       : num [1:1460] 2 5 9 2 12 10 8 11 4 1 ...
##  $ YrSold       : num [1:1460] 2008 2007 2008 2006 2008 ...
##  $ SaleType     : chr [1:1460] "WD" "WD" "WD" "WD" ...
##  $ SaleCondition: chr [1:1460] "Normal" "Normal" "Normal" "Abnorml" ...
##  $ SalePrice    : num [1:1460] 208500 181500 223500 140000 250000 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Id = col_double(),
##   ..   MSSubClass = col_double(),
##   ..   MSZoning = col_character(),
##   ..   LotFrontage = col_double(),
##   ..   LotArea = col_double(),
##   ..   Street = col_character(),
##   ..   Alley = col_character(),
##   ..   LotShape = col_character(),
##   ..   LandContour = col_character(),
##   ..   Utilities = col_character(),
##   ..   LotConfig = col_character(),
##   ..   LandSlope = col_character(),
##   ..   Neighborhood = col_character(),
##   ..   Condition1 = col_character(),
##   ..   Condition2 = col_character(),
##   ..   BldgType = col_character(),
##   ..   HouseStyle = col_character(),
##   ..   OverallQual = col_double(),
##   ..   OverallCond = col_double(),
##   ..   YearBuilt = col_double(),
##   ..   YearRemodAdd = col_double(),
##   ..   RoofStyle = col_character(),
##   ..   RoofMatl = col_character(),
##   ..   Exterior1st = col_character(),
##   ..   Exterior2nd = col_character(),
##   ..   MasVnrType = col_character(),
##   ..   MasVnrArea = col_double(),
##   ..   ExterQual = col_character(),
##   ..   ExterCond = col_character(),
##   ..   Foundation = col_character(),
##   ..   BsmtQual = col_character(),
##   ..   BsmtCond = col_character(),
##   ..   BsmtExposure = col_character(),
##   ..   BsmtFinType1 = col_character(),
##   ..   BsmtFinSF1 = col_double(),
##   ..   BsmtFinType2 = col_character(),
##   ..   BsmtFinSF2 = col_double(),
```

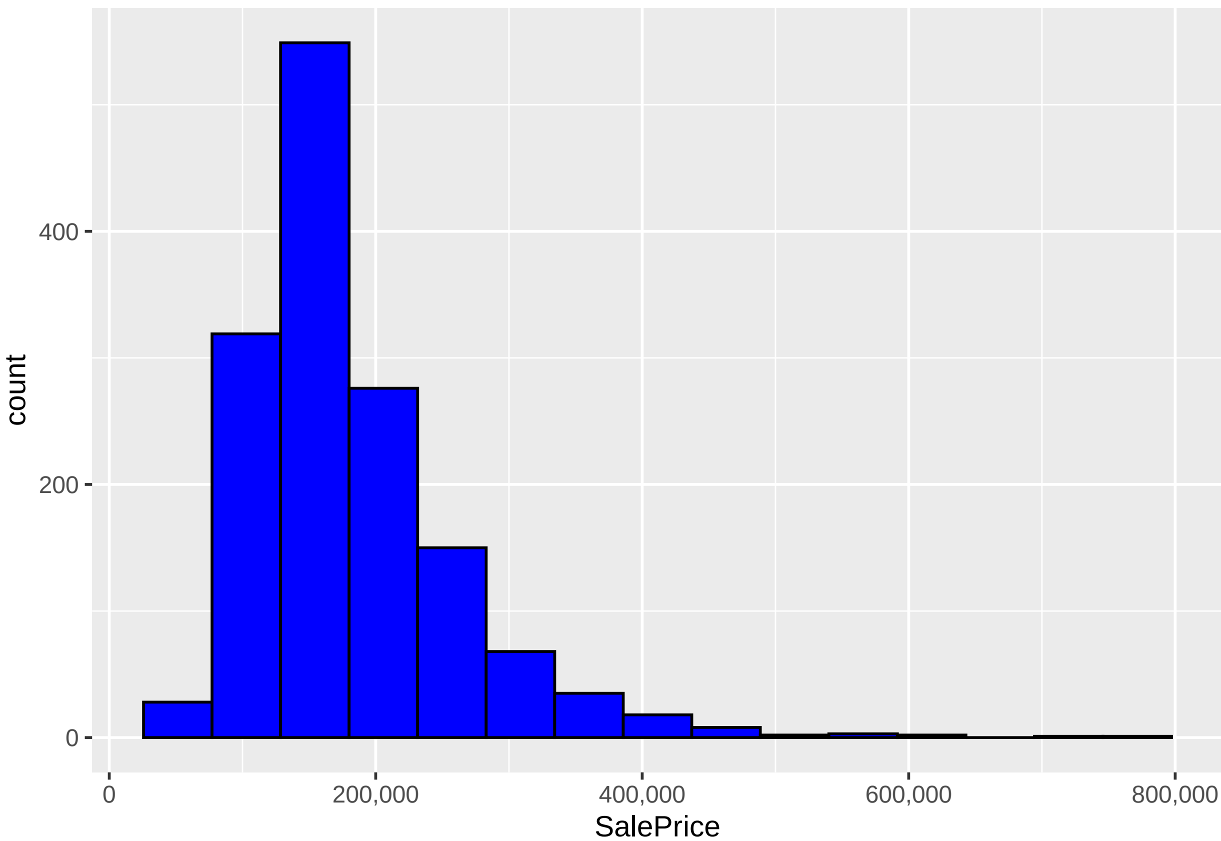
```
##   ..   BsmtUnfSF = col_double(),
##   ..   TotalBsmtSF = col_double(),
##   ..   Heating = col_character(),
##   ..   HeatingQC = col_character(),
##   ..   CentralAir = col_character(),
##   ..   Electrical = col_character(),
##   ..   `1stFlrSF` = col_double(),
##   ..   `2ndFlrSF` = col_double(),
##   ..   LowQualFinSF = col_double(),
##   ..   GrLivArea = col_double(),
##   ..   BsmtFullBath = col_double(),
##   ..   BsmtHalfBath = col_double(),
##   ..   FullBath = col_double(),
##   ..   HalfBath = col_double(),
##   ..   BedroomAbvGr = col_double(),
##   ..   KitchenAbvGr = col_double(),
##   ..   KitchenQual = col_character(),
##   ..   TotRmsAbvGrd = col_double(),
##   ..   Functional = col_character(),
##   ..   Fireplaces = col_double(),
##   ..   FireplaceQu = col_character(),
##   ..   GarageType = col_character(),
##   ..   GarageYrBlt = col_double(),
##   ..   GarageFinish = col_character(),
##   ..   GarageCars = col_double(),
##   ..   GarageArea = col_double(),
##   ..   GarageQual = col_character(),
##   ..   GarageCond = col_character(),
##   ..   PavedDrive = col_character(),
##   ..   WoodDeckSF = col_double(),
##   ..   OpenPorchSF = col_double(),
##   ..   EnclosedPorch = col_double(),
##   ..   `3SsnPorch` = col_double(),
##   ..   ScreenPorch = col_double(),
##   ..   PoolArea = col_double(),
##   ..   PoolQC = col_character(),
##   ..   Fence = col_character(),
##   ..   MiscFeature = col_character(),
##   ..   MiscVal = col_double(),
##   ..   MoSold = col_double(),
##   ..   YrSold = col_double(),
##   ..   SaleType = col_character(),
##   ..   SaleCondition = col_character(),
##   ..   SalePrice = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
head(train)
```

```
## # A tibble: 6 x 81
##      Id MSSubClass MSZoning LotFrontage LotArea Street Alley LotShape
##   <dbl>      <dbl> <chr>          <dbl>   <dbl> <chr>  <chr> <chr>
## 1     1         60 RL                65    8450 Pave   <NA>  Reg
## 2     2         20 RL                80    9600 Pave   <NA>  Reg
## 3     3         60 RL                68   11250 Pave   <NA>  IR1
```

```
## 4     4         70 RL                60    9550 Pave   <NA>  IR1
## 5     5         60 RL                84   14260 Pave   <NA>  IR1
## 6     6         50 RL                85   14115 Pave   <NA>  IR1
## # i 73 more variables: LandContour <chr>, Utilities <chr>, LotConfig <chr>,
## #   LandSlope <chr>, Neighborhood <chr>, Condition1 <chr>, Condition2 <chr>,
## #   BldgType <chr>, HouseStyle <chr>, OverallQual <dbl>, OverallCond <dbl>,
## #   YearBuilt <dbl>, YearRemodAdd <dbl>, RoofStyle <chr>, RoofMatl <chr>,
## #   Exterior1st <chr>, Exterior2nd <chr>, MasVnrType <chr>, MasVnrArea <dbl>,
## #   ExterQual <chr>, ExterCond <chr>, Foundation <chr>, BsmtQual <chr>,
## #   BsmtCond <chr>, BsmtExposure <chr>, BsmtFinType1 <chr>, ...
```

Distribution of the target variable

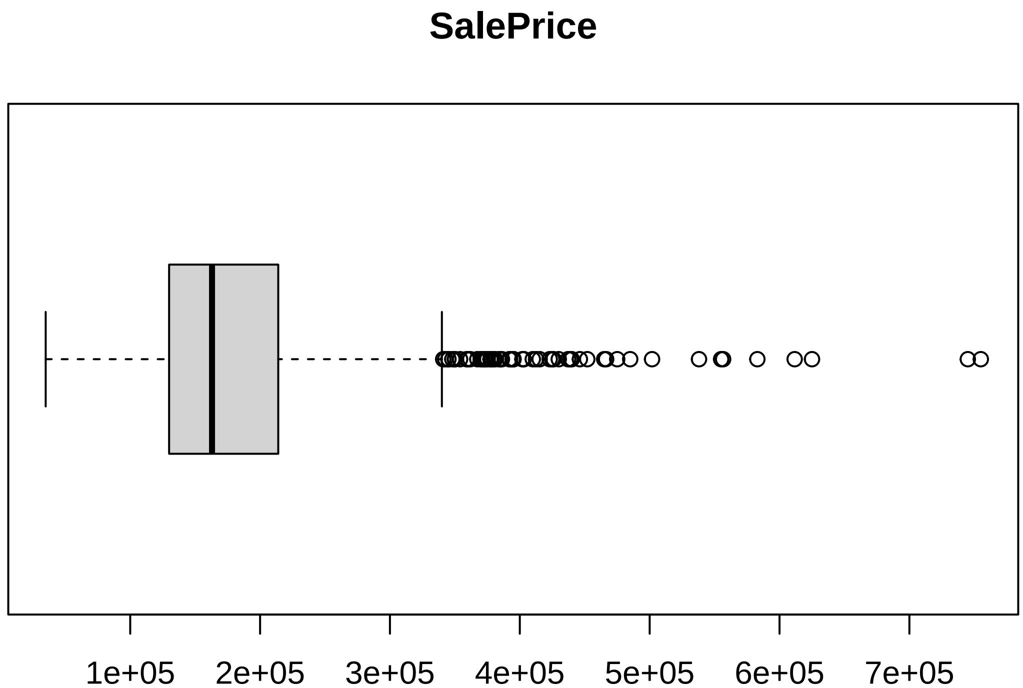


Check for Skewness and Kurtosis

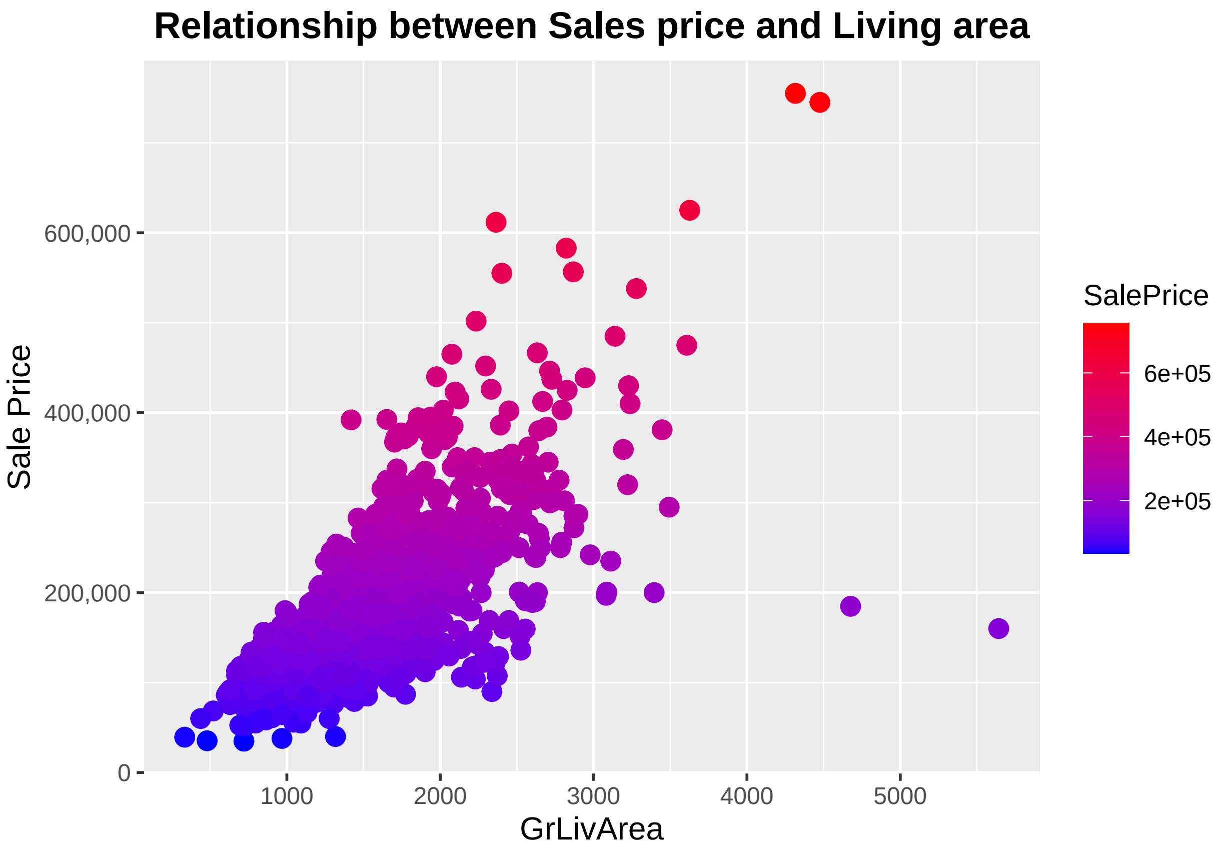
```
## Skewness: 1.880941
## Kurtosis: 9.509812
```

The skewness value of 1.563515 indicates that the distribution is moderately skewed to the right, while the kurtosis value of 6.862666 indicates the data has heavy tails, implying a higher chance of higher values.

Check for Outliers in the SalesPrice

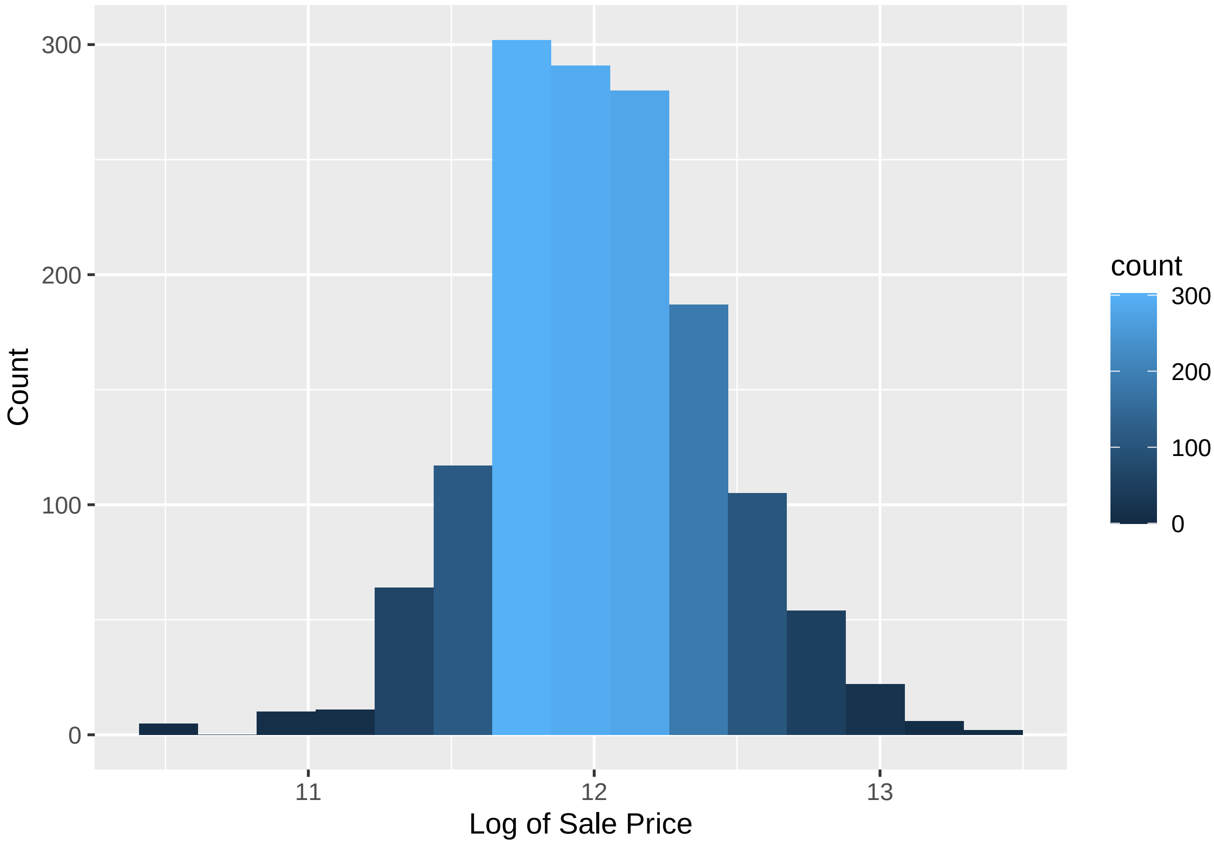



```
##  [1] 345000 385000 438780 383970 372402 412500 501837 475000 386250 403000
## [11] 415298 360000 375000 342643 354000 377426 437154 394432 426000 555000
## [21] 440000 380000 374000 430000 402861 446261 369900 451950 359100 345000
## [31] 370878 350000 402000 423000 372500 392000 755000 361919 341000 538000
## [41] 395000 485000 582933 385000 350000 611657 395192 348000 556581 424870
## [51] 625000 392500 745000 367294 465000 378500 381000 410000 466500 377500
## [61] 394617
```



We've identified outliers in the sale price column, notably those showcasing unusually large houses sold at remarkably cheap prices. As the dataset author's recommended, we'll exclude any houses with a living area exceeding 4000 square feet from our analysis.

Log Transformation



After the log transformation the distribution looks more normally distributed.

Check for the missing values in every column

```
##            Id    MSSubClass      MSZoning   LotFrontage       LotArea
##             0             0             0           259             0
##        Street         Alley      LotShape   LandContour     Utilities
##             0          1365             0             0             0
##     LotConfig     LandSlope  Neighborhood    Condition1    Condition2
##             0             0             0             0             0
##      BldgType    HouseStyle   OverallQual   OverallCond     YearBuilt
##             0             0             0             0             0
##  YearRemodAdd     RoofStyle      RoofMatl   Exterior1st   Exterior2nd
##             0             0             0             0             0
##    MasVnrType    MasVnrArea     ExterQual     ExterCond    Foundation
##             8             8             0             0             0
##      BsmtQual      BsmtCond  BsmtExposure  BsmtFinType1    BsmtFinSF1
##            37            37            38            37             0
##  BsmtFinType2    BsmtFinSF2     BsmtUnfSF   TotalBsmtSF       Heating
##            38             0             0             0             0
##     HeatingQC    CentralAir    Electrical      1stFlrSF      2ndFlrSF
##             0             0             1             0             0
##  LowQualFinSF     GrLivArea  BsmtFullBath  BsmtHalfBath      FullBath
##             0             0             0             0             0
##      HalfBath  BedroomAbvGr  KitchenAbvGr   KitchenQual  TotRmsAbvGrd
```

```
##            0            0            0            0            0
##    Functional   Fireplaces  FireplaceQu   GarageType  GarageYrBlt
##            0            0          690           81           81
##  GarageFinish   GarageCars   GarageArea   GarageQual   GarageCond
##           81            0            0           81           81
##    PavedDrive   WoodDeckSF  OpenPorchSF EnclosedPorch    3SsnPorch
##            0            0            0            0            0
##   ScreenPorch     PoolArea       PoolQC        Fence  MiscFeature
##            0            0         1451         1176         1402
##       MiscVal       MoSold       YrSold     SaleType SaleCondition
##            0            0            0            0            0
##     SalePrice
##            0
```

Check for Duplicate samples.

```
## [1] "There are 0 duplicate rows "
```

Now lets start to impute the NA values .

First create two subsets containing numerical and categorical data respectively.

```
## [1] "Number of categorical features are: 43"
```

```
## [1] "Number of numerical features are: 38"
```

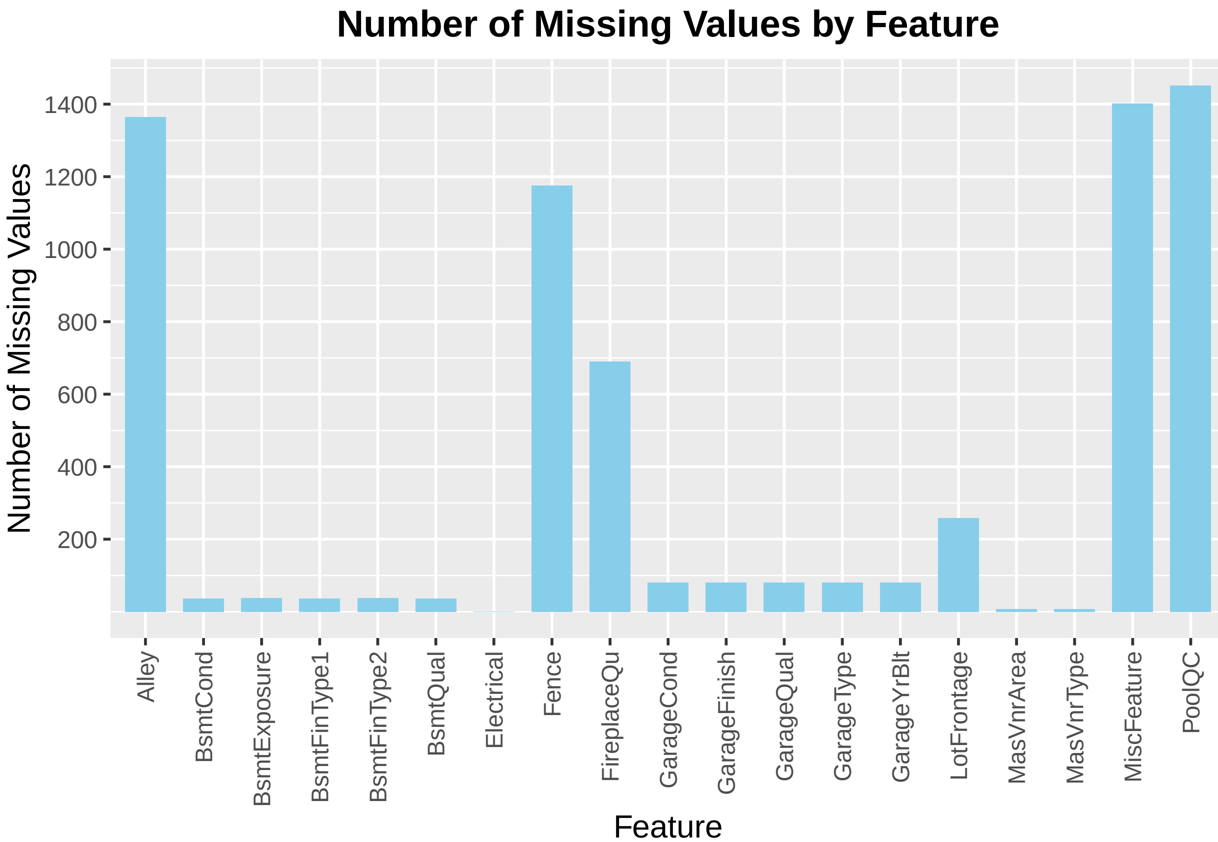
checking the number of missing values in each column

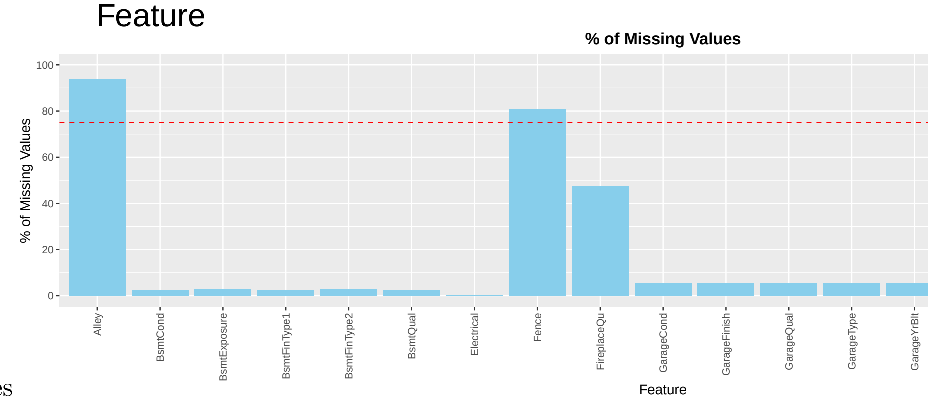
```
# Make a list of features with missing values
features_with_na <- names(train)[apply(train, 2, function(x) any(is.na(x)))]

# Print the feature name and the number of missing values
for (feature in features_with_na) {
  num_missing <- sum(is.na(train[[feature]]))
  cat(feature, ":", num_missing, "missing values\n")
}
```

```
## LotFrontage : 259 missing values
## Alley : 1365 missing values
## MasVnrType : 8 missing values
## MasVnrArea : 8 missing values
## BsmtQual : 37 missing values
## BsmtCond : 37 missing values
## BsmtExposure : 38 missing values
## BsmtFinType1 : 37 missing values
## BsmtFinType2 : 38 missing values
## Electrical : 1 missing values
## FireplaceQu : 690 missing values
## GarageType : 81 missing values
## GarageYrBlt : 81 missing values
## GarageFinish : 81 missing values
## GarageQual : 81 missing values
## GarageCond : 81 missing values
## PoolQC : 1451 missing values
## Fence : 1176 missing values
## MiscFeature : 1402 missing values
```

Identifying features with missing values and creating a dataframe to store information about these missing values.

plotting the number of missing value

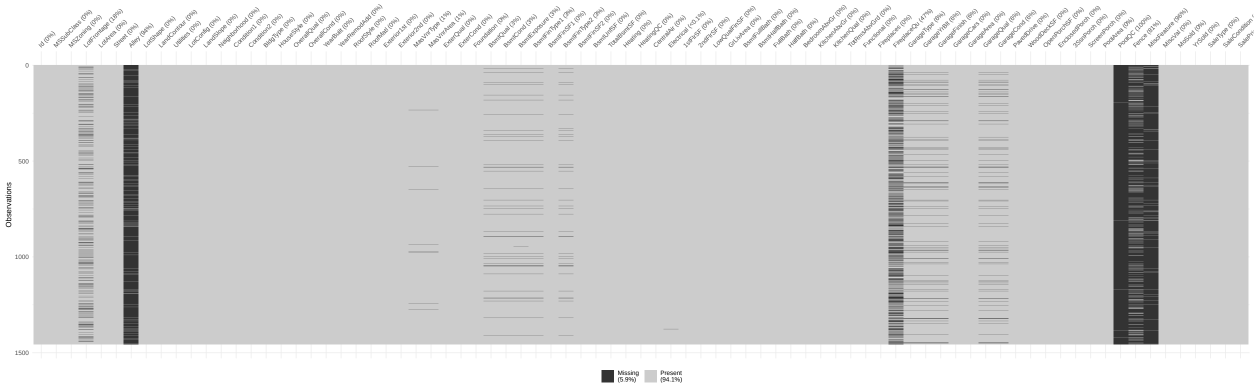




Plotting the percentage of missing values
Observation : We can clearly see from the above graph there are four columns that are missing over 75% of their values. These columns are Alley, Fence, MiscFeature,PoolQC. From the graph, it's evident that four columns stand out—they're missing more than 75% of their values. These columns—Alley, Fence, MiscFeature, and PoolQC—clearly show that over three-quarters of their information is missing.

But this doesn't mean we can remove them directly. As having Na value in there means the houses don't have these resources anf that might impact the target variable.

Visualise the missing values



Extracting feature names with NA values and preparing data.

```
## [1] "Number of categorical features with NA are: 16"
```

```
## [1] "Number of numerical features with NA are: 3"
```

Here we see that out of 19 columns that are missing only 3 are numeric . So First we will try to impute the numerical variables that contain NAs.

Analyzing correlation between numerical variables and target variable.

```
## [1] "Correlation between LotFrontage and SalePrice: 0.35677281588612"
## [1] "Correlation between MasVnrArea and SalePrice: 0.478862290442391"
## [1] "Correlation between GarageYrBlt and SalePrice: 0.499229793243099"
```

Imputing missing values in LotFrontage and MasVnrArea by grouping and median

The missing values in the LotFrontage variable were addressed through imputation to maintain the integrity of our dataset . For LotFrontage, we grouped data by Neighborhood and BldgType and replaced missing values with the median of each subgroup. Similarly, for MasVnrArea, which denotes masonry veneer area, we grouped the data by MSSubClass and Exterior1st.

```r
sum(is.na(train$LotFrontage))
```

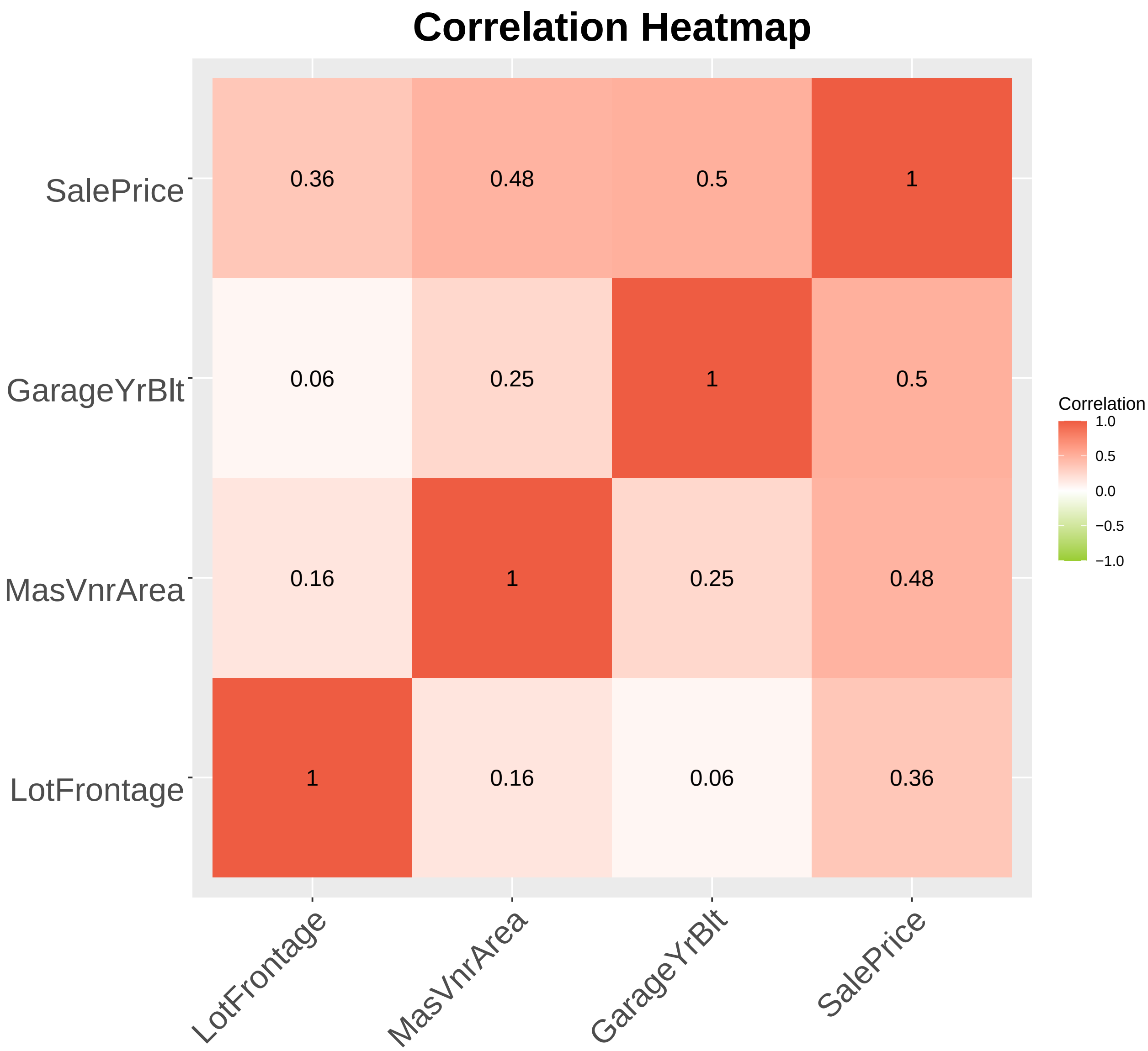
```
## [1] 2
```

Since there are still 2values left we will apply the same process again

```r
sum(is.na(train$LotFrontage))
```

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

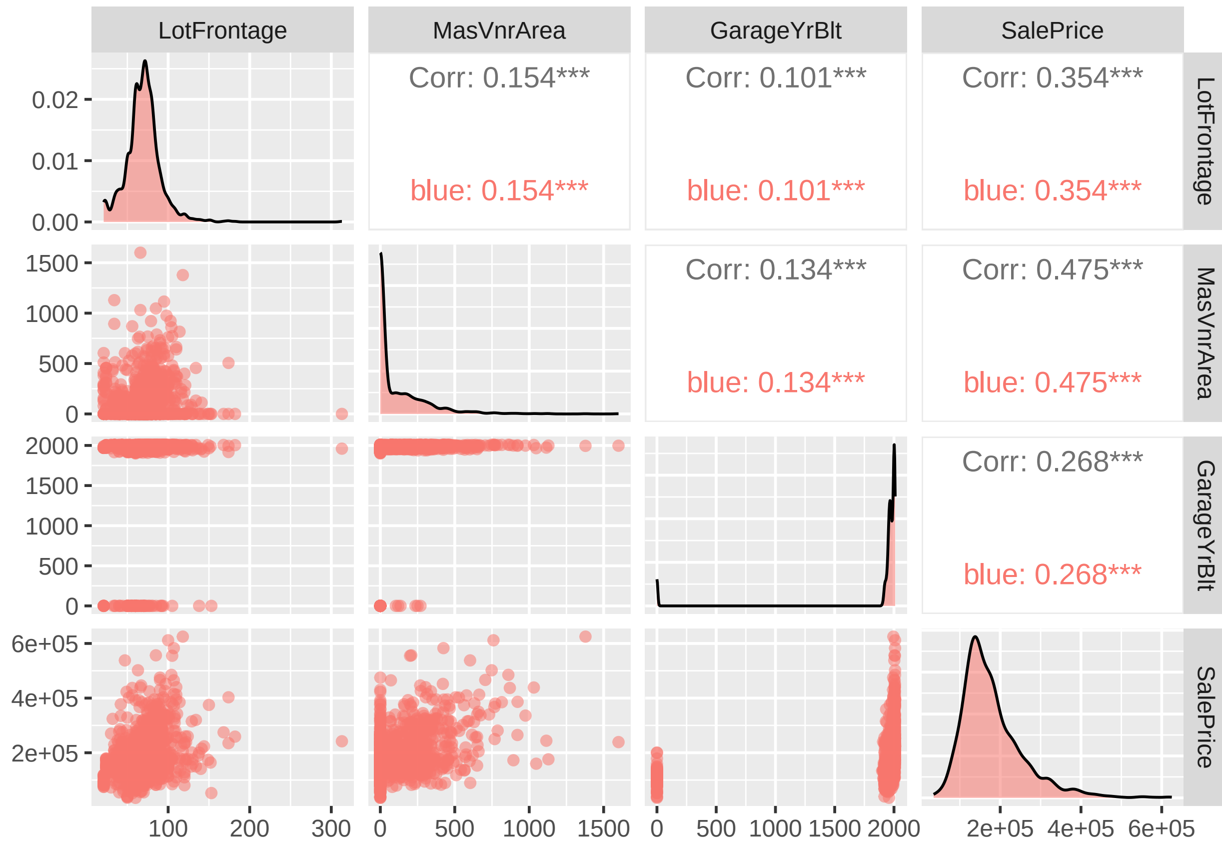
Check the correlation between the numerical variables which contained NA values with the sales price

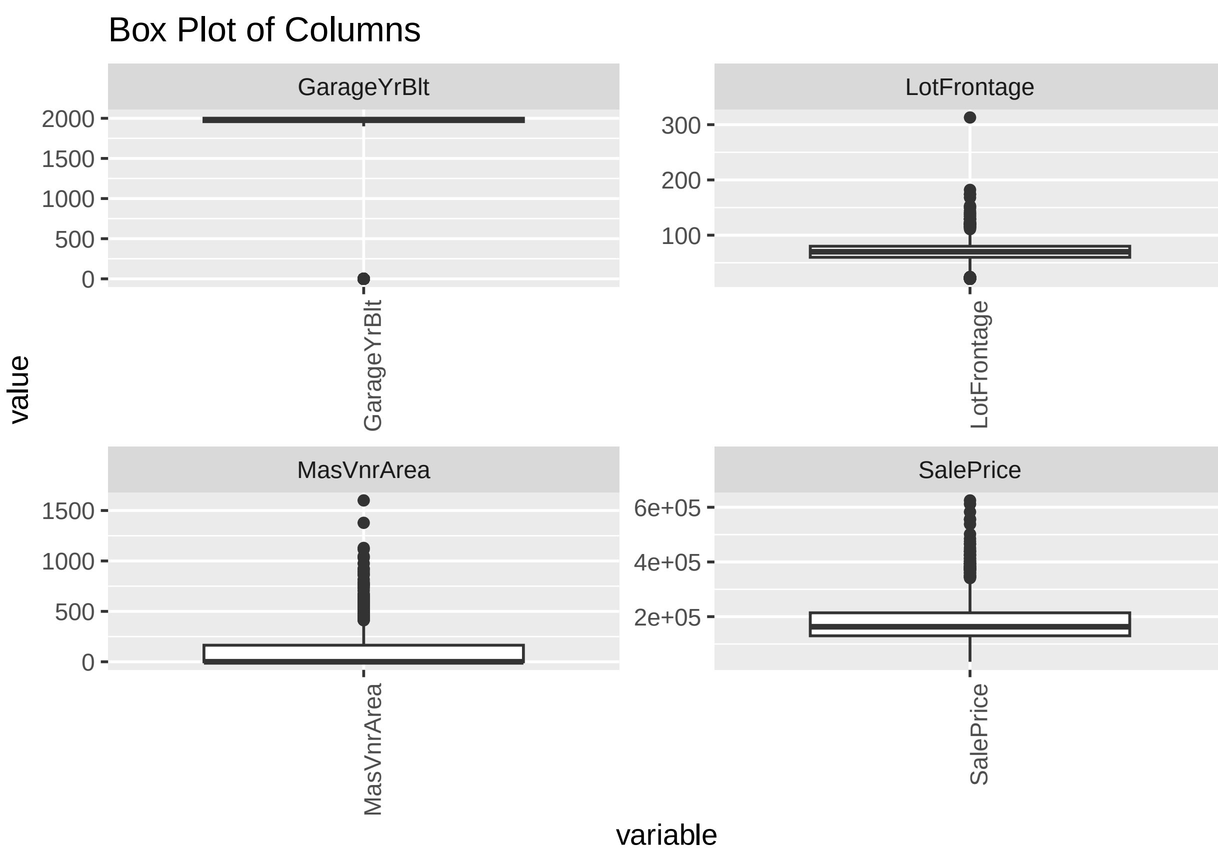



```
subset_with_na <- train[is.na(train$GarageYrBlt), ]

sum(is.na(subset_with_na$GarageType))
```

```
## [1] 81
```

Now we see that the rows with NA value for the column GarageYrBlt also has the NA value for GarageType. The dataset description says that Na value in GarageType means there is no garage and thus making GarageYrBlt obsolete for the data sample. Hence we are replacing the NA with 0

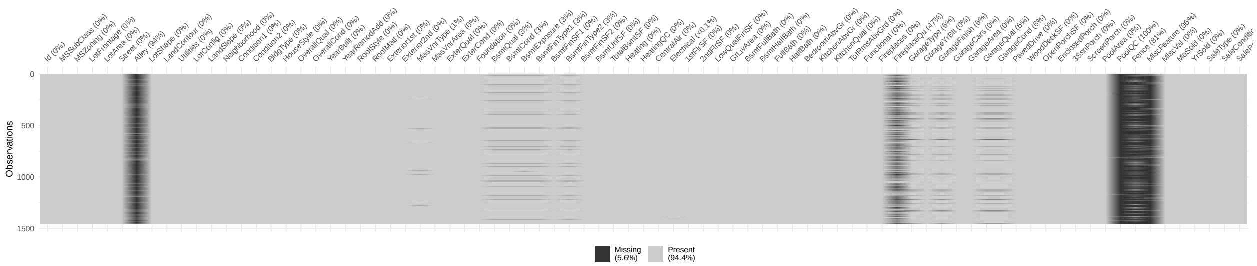

LotFrontage
MasVnrArea
GarageYrBlt
SalePrice
Corr: 0.154***
blue: 0.154***
Corr: 0.101***
blue: 0.101***
Corr: 0.354***
blue: 0.354***
Corr: 0.134***
blue: 0.134***
Corr: 0.475***
blue: 0.475***
Corr: 0.268***
blue: 0.268***
LotFrontage
MasVnrArea
GarageYrBlt
SalePrice


Box Plot of Columns
GarageYrBlt
LotFrontage
MasVnrArea
SalePrice
value
variable

So till now we have imputed the NA values in the numerical column. Next we try to remove the NA values in Categorical columns. These columns have na values in them

```
##  [1] "Alley"        "MasVnrType"   "BsmtQual"     "BsmtCond"     "BsmtExposure"
##  [6] "BsmtFinType1" "BsmtFinType2" "Electrical"   "FireplaceQu"  "GarageType"
## [11] "GarageFinish" "GarageQual"   "GarageCond"   "PoolQC"       "Fence"
## [16] "MiscFeature"
```

```
##        feature na_count percent_missing
## 17      PoolQC     1451        99.65659
## 19 MiscFeature     1402        96.29121
## 2        Alley     1365        93.75000
## 18       Fence     1176        80.76923
## 11 FireplaceQu      690        47.39011
```



Now we can see that Alley, Fence, MiscFeature,PoolQC, FireplaceQu have approximately 50 % and above missing values. So we will try to impute those values and check if they are significant to our variable. According to the dataset description,Na values in Alley, Fence, MiscFeature,PoolQC, FireplaceQu, Bsmt, Garage etc means the houses don't have these things. So we can not just drop them .

Hence we decided to Replace missing values in the columns with "None"

Replace missing values in the columns with "None", Iterate over each column that contains Bsmt and Garage and replace missing values with "None"

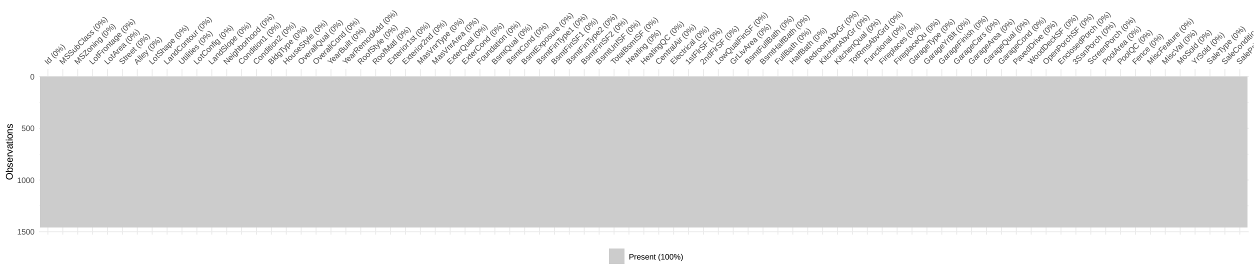
Check again for the columns containing missing values

```
##           Id   MSSubClass     MSZoning  LotFrontage      LotArea
##            0            0            0            0            0
##       Street        Alley     LotShape  LandContour    Utilities
##            0            0            0            0            0
##    LotConfig    LandSlope Neighborhood   Condition1   Condition2
##            0            0            0            0            0
##     BldgType   HouseStyle  OverallQual  OverallCond    YearBuilt
##            0            0            0            0            0
## YearRemodAdd    RoofStyle     RoofMatl  Exterior1st  Exterior2nd
##            0            0            0            0            0
##   MasVnrType   MasVnrArea    ExterQual    ExterCond   Foundation
##            0            0            0            0            0
##     BsmtQual     BsmtCond BsmtExposure BsmtFinType1   BsmtFinSF1
##            0            0            0            0            0
## BsmtFinType2   BsmtFinSF2    BsmtUnfSF  TotalBsmtSF      Heating
##            0            0            0            0            0
##    HeatingQC   CentralAir   Electrical     1stFlrSF     2ndFlrSF
##            0            0            1            0            0
## LowQualFinSF    GrLivArea BsmtFullBath BsmtHalfBath     FullBath
##            0            0            0            0            0
##     HalfBath BedroomAbvGr KitchenAbvGr  KitchenQual TotRmsAbvGrd
```

```
##             0             0             0             0             0
##    Functional    Fireplaces   FireplaceQu    GarageType   GarageYrBlt
##             0             0             0             0             0
##  GarageFinish    GarageCars    GarageArea    GarageQual    GarageCond
##             0             0             0             0             0
##    PavedDrive    WoodDeckSF   OpenPorchSF EnclosedPorch     3SsnPorch
##             0             0             0             0             0
##   ScreenPorch      PoolArea        PoolQC         Fence   MiscFeature
##             0             0             0             0             0
##       MiscVal        MoSold        YrSold      SaleType SaleCondition
##             0             0             0             0             0
##     SalePrice
##             0
```

Since Electrical has just one row which is missing , we decide to drop that row.

Plot gain and see if any missing values are left



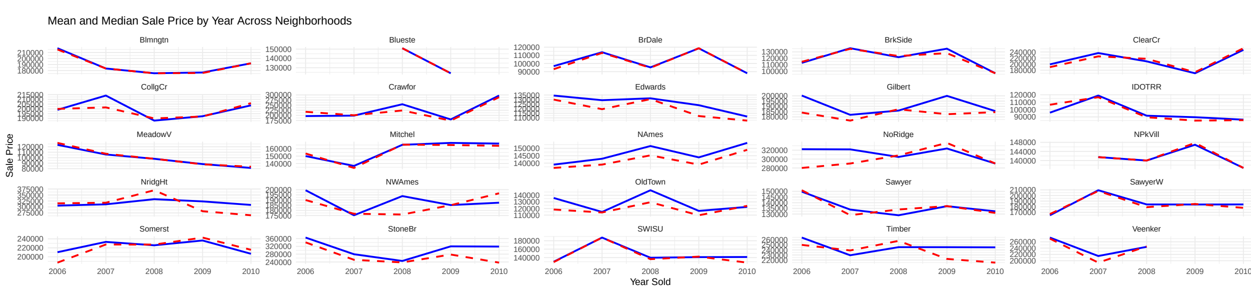

```r
sum(is.na(train))
```

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

Finally all the NA values have been imputed.

Problem1 : How the mean and meadian Sales Prices for each neaghborhood vary from 2006 to 2010 and compare with each other.

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

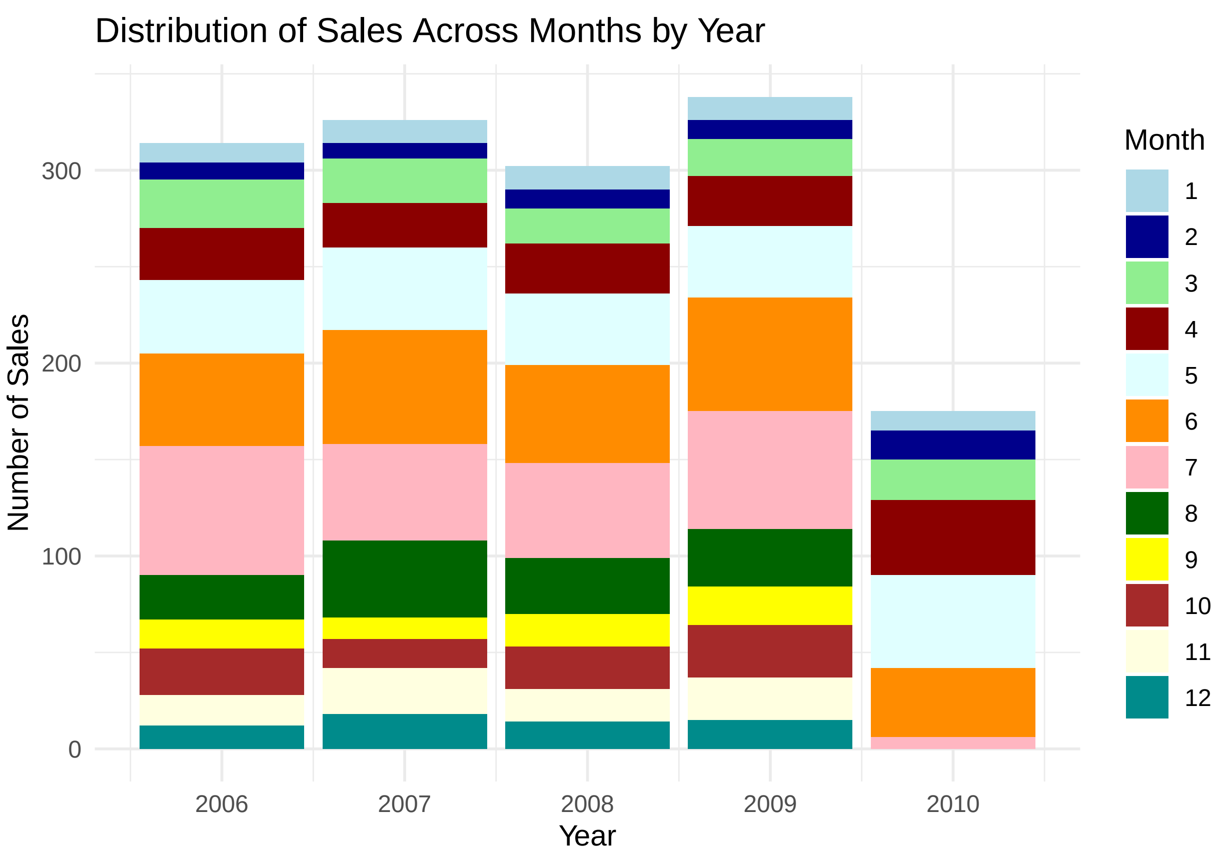


The graph represents a series of subplots each corresponding to a different neighborhood. Each subplot shows trends in both mean and median sale prices from 2006 to 2010. While the mean is sensitive to outliers, the median can give a better sense of the central tendency when distributions are skewed by very high or very low values.

Key Observations from the graph

- Some neighborhoods show stable prices over the years, while others show sharp increases or decreases.Example: "NridgHt" showed less volatility and maintain higher price levels, suggesting a stable and potentially high-value market.
- Almost all of the neighborhoods show rapid decrease in sales price in 2007/2008 due to the global economic fluctuations.
- Post 2008 most neighborhoods seem to recver from the crisis showing the econmy had recovered a bit . for example Gilbert in 2009 had almost the same mean and median price as it had in 2006
- The difference between mean and median prices in some neighborhoods can suggest the presence of outliers—highly priced sales that move the mean upwards.
- Some neighborhoods like Mitchell after the 2007/08 dip, show consistent upward or stable trends, potentially indicating steady market demand and growth. This could be due the the building of some good entities. in the neighborhood like park, school etc
- NridgHt" and NoRidge consistently show higher sale prices representing affluency of the neighborhood.
- Meadow seem to have the lowest sale prices and they keep on dropping every year.

Problem2 : How sales numbers vary for every month over different years.



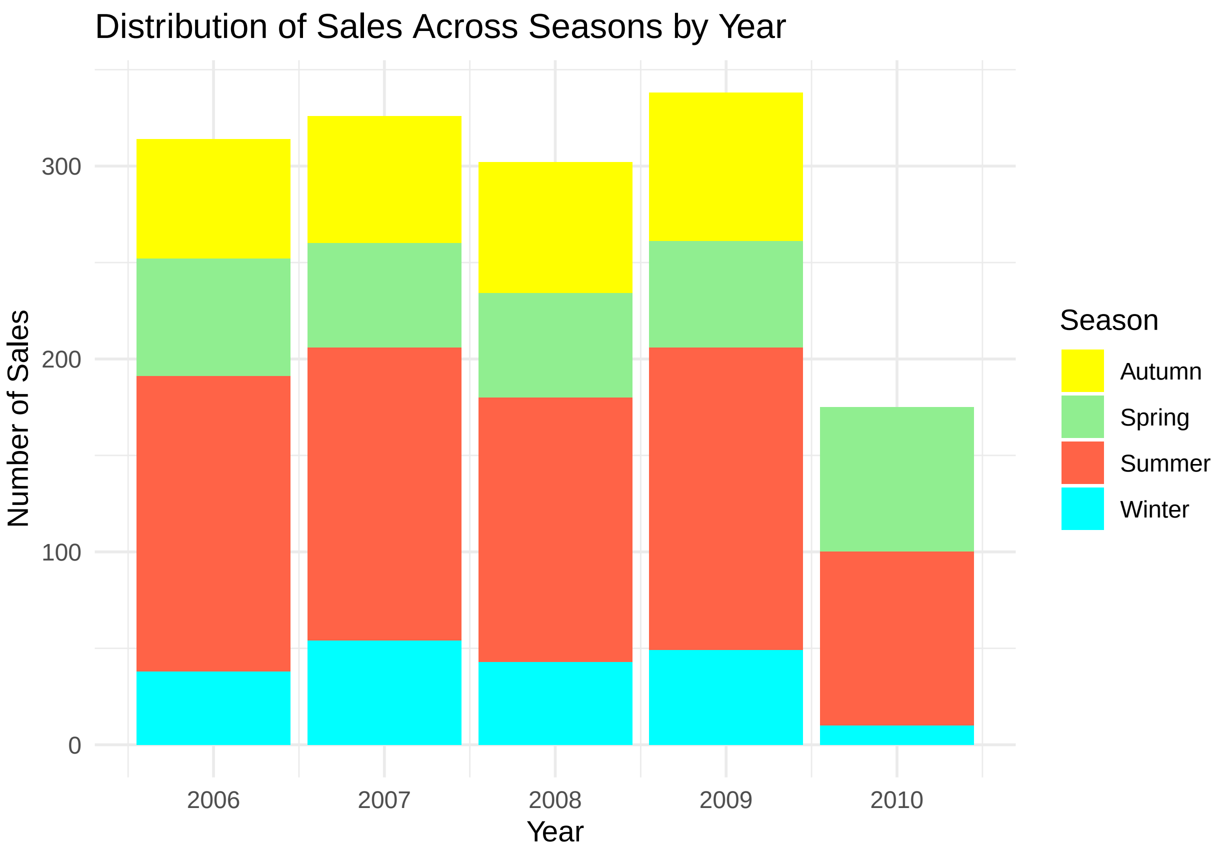
The graph shows a stacked bar chart representation of number of sales for each month from 2006 to 2010. Each bar represents a year, and each segment of the bar corresponds to a month, color-coded to differentiate between the months. The graph helps in a visual comparison of sales activity across different months of the year and across the five-year span.

Key observations from the graph

- Sales peak during the summer months (especially May, June, and July), showing these are the most popular months for buying homes.

- No of sales in December, January, and February suggests a seasonal slowdown for the sales in winter months.
- Sales numbers fluctuate from year to year reflecting changes in the housing market.for example the july has more sales in 2006 , less in 2007 even thought the volume of sales in 2007 were more than in 200-
- Despite the volume of sales fluctuates yearly, the pattern remains consistent that peaks in summer and lowest in winter.
- Real Estate Workers can use this information to push their advertising and open house events more in May and June when more people are looking to buy.

Problem 3 :How did the number of home sales in Ames, Iowa fluctuate seasonally and annually

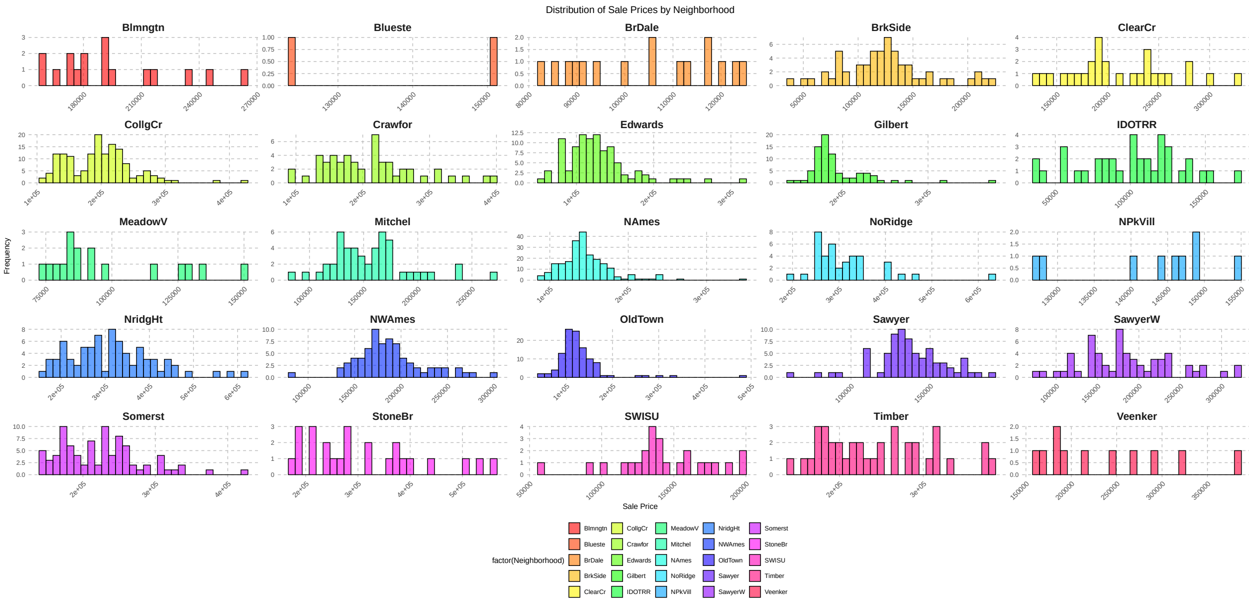


The graph illustrates the number of property sales broken down by season for each year from 2006 through 2010.

Key Observations

- Summer is the most dominant season followed by spring in regards of house sales.with over 100 properties sold every year.
- Sales in winter are the lowest.
- Sales in 2010 are visibly lower than in any previous year for every season, indicating a possible slowdown

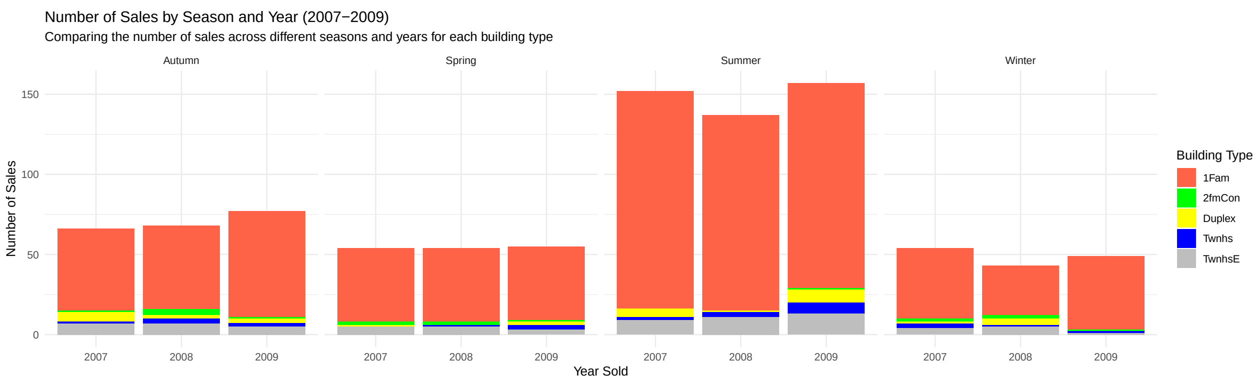
Problem 4 : Investigating the economic diversity and real estate market dynamics in each neighborhood?



The graph presents a series of histograms that depict the frequency distribution of sale prices within specific neighborhoods revealing how different neighborhoods cater to various economic segments.

###Observations - Neighborhoods display varied price distributions, indicating economic diversity across Ames - NridgHt and StoneBr stand out with a significant number of transactions in the higher price brackets i.e over $ 200,000 - MeadowV show a concentration of sales in the lower price ranges (under $100k), indicating it may have more affordable housing options. - Sawyer, BrkSide, and NAmes exhibit a distribution of sales around ($100k-$200k), suggesting a real estate market appealing to a middle-class.

Problem 5 : Investigating the Variation in Building Type Sales Across Seasons and Years during the financial crisis of 2008"

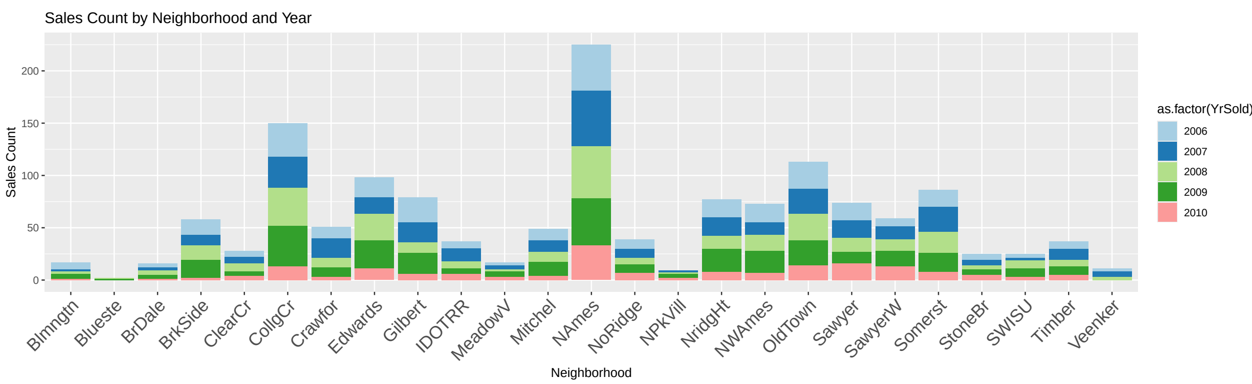


The stacked bar chart is displaying the total number of property sales divided by five building types: 1-Family Detached (1Fam), 2-Family Conversion (2fmCon), Duplex, Townhouse (Twnhs), and Townhouse End Unit (TwnhsE) for each season thoroughout the economic crisis. This graph effectively highlights how different types of buildings fared in terms of sales across different seasons during the critical period.

Key observation

- This graph also shows for every year Summer is the highest selling season even during the time of economic crisis.
- 1 Fam building sales was the highest in every season in all the three years of the period. 1Fam showed remarkable resilience during the financial crisis, maintaining an upward trajectory in sale prices despite the economic crisis.
- Summer sees the highest sales volumes and winter sees the least
- Spring sales didn't suffer because of the crisis.
- Sales of TwnhsE are low but stayed stable in every year with summer taking the majority of sales.
- showed high vulnerability to financial change as it had a steep drop after 2008
- Summer sales showed massive recovery in 2009 after the crisis for every building type.

Problem 6 : How have home sales trends varied across different neighborhoods throughout the years.



The stacked bar chart representing the number of home sales per year in each neighborhood from 2006 to 2010.

Observations

- There is significant variability in sales counts across neighborhoods, suggesting diverse housing market dynamics.
- The year 2008 does not show a uniform decline across all neighborhoods, which might indicate that some areas were more resilient or even unaffected by the economic
- Some neighborhoods, such as 'OldTown' and 'Edwards', display a consistent number of sales each year, indicating stability
- Certain neighborhoods consistently showed great sales numbers across the years, such as 'NAmes', 'CollgCr'.

Problem 7 : # How do average sale prices vary by seasons across different neighborhoods

The above aims to show the seasonal influences on house prices telling us when might be an optimal time to buy or sell properties in specific neighborhoods. This can help real estate investors, homeowners, and market analysts to make informed decisions.

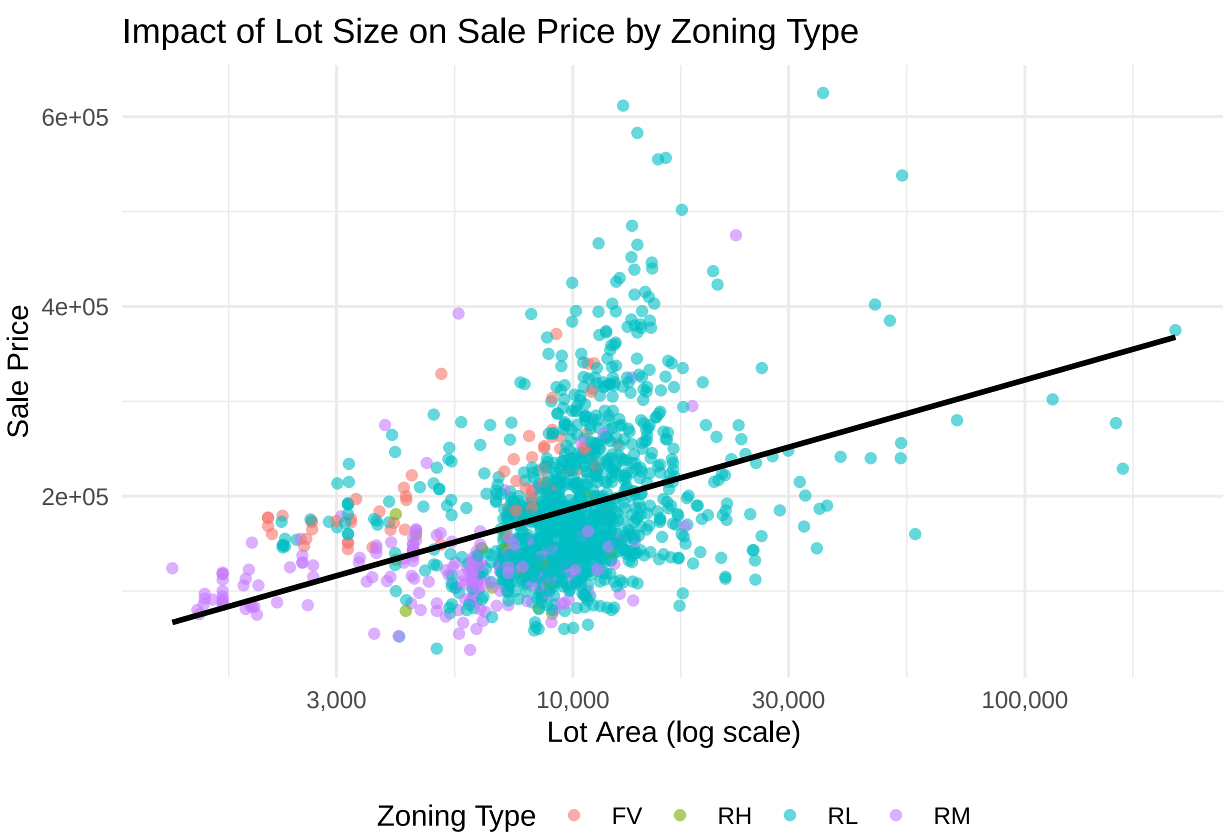
Key observations from the graph:

- .The graph shows considerable variability in average sale prices for different seasons for different neighborhoods. Some neighborhoods show significant price changes between seasons, while others are more stable throughout.

- Each neighborhood exhibits unique seasonal pricing trends for example the neighborhood Veenker has a very high average values for the winter season whereas winter has the lease average sales price for most neighborhoods.
- Summer/Spring seems to have better average prices in most of the neighborhoods whereas the in autumn we see decline in prices and winter has the lease prices . This infers that spring and summer are good season to sell and Winter is the the best time according to SalePrice.
- IDOTRR and MeadowW have the constantly lowest price among all the neighborhoods
- NridgHt and NoRidgHt are the most affluent neighborhood where the prices stayed almost constant throughout the year.

Problem 8 : How does lot size impact the sale price of properties across different zoning types

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

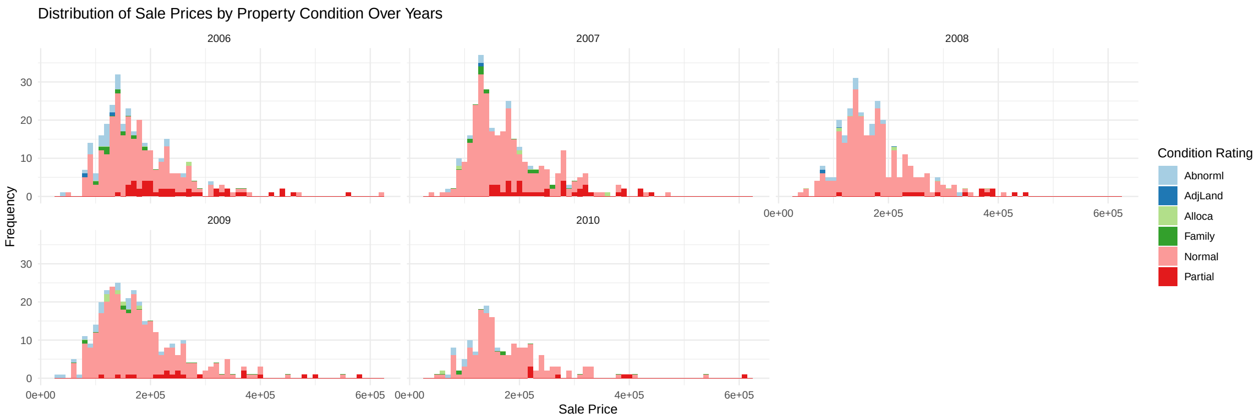


The graph visualizes the relationship between lot area and sale price across different zoning types. The x-axis represents the lot area on a logarithmic scaleto represent the wide range of lot sizes while the y-axis represents the sale price. Different colors represent different zoning types. A black line indicates a linear regression fit through the data.

Key Observations

- . The regression line shows a general positive correlation between lot size and sale price. This makes sense as normally larger lots tend to have higher sale price. 2.FV (Floating Village Residential) shows medium lot sizes with a high variation in sale prices.
- RL show a wide spread in sale prices at similar lot sizes indicating there are other factors impacting the sale prices beyond just lot size.
- There are noticeable outliers, particularly in zones like RL and RM

Problem9: How do home sale prices distribute across various property conditions within different years

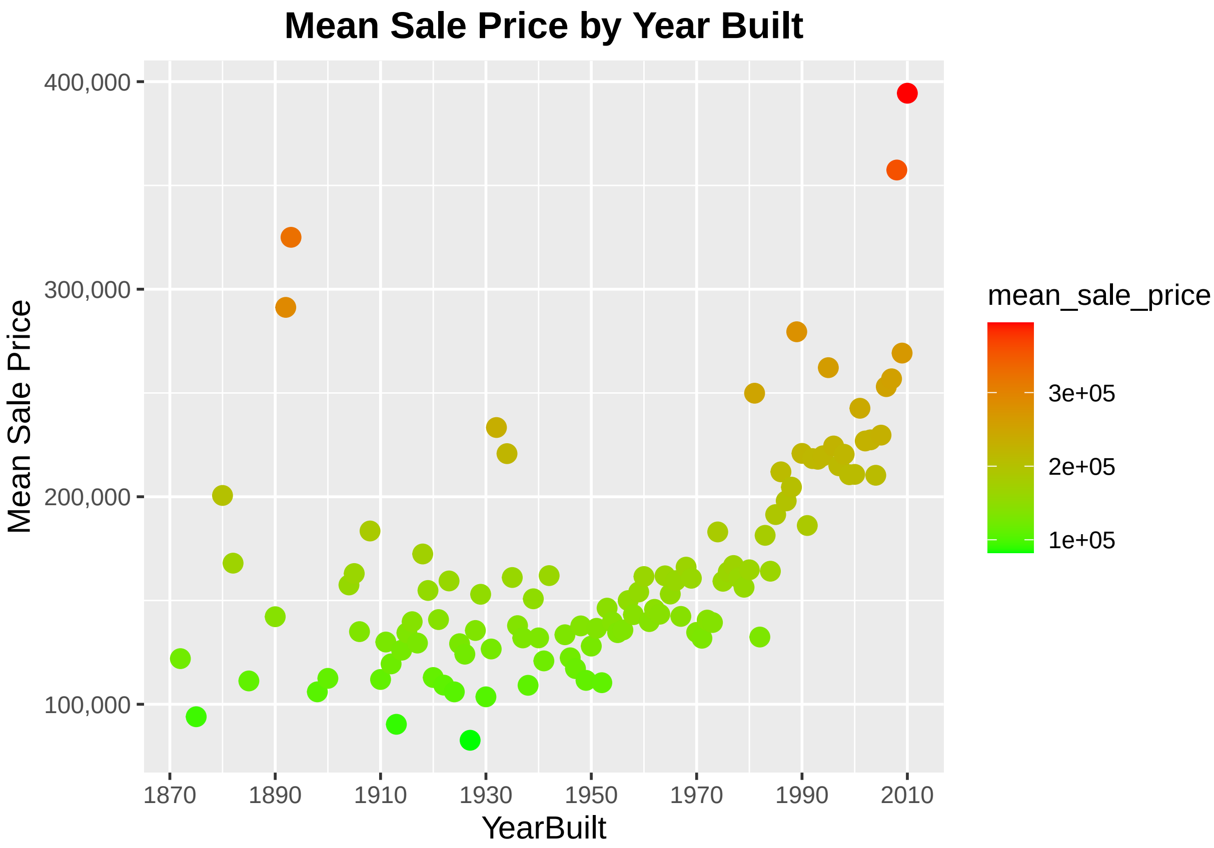


The graph shows a series of histograms for each year from 2006 to 2010. These histograms show the frequency of sale prices categorized by the condition of the property at the time of sale

Key observtions

- Normal Sales condition dominates in all years representing the most properties are sold under typical market conditions 2.There is a noticeable increase in 'Abnormal' sales in 2008, which may correspond with the financial crisis
- The histograms from year to year show fluctuations in both the number of sales and the price distribution reflecting changes in market conditions
- Sales under conditions like AdjLand and Alloca are rare across all years

Some other Visualisations that I did to explore the data. We used ggplot and plot_ly as well. SInce the plot_ly graphs were not rendered on pdf , I request you to please go through the code or the html version of the file



Feature Engineering

Since most of the variables are discrete, we will encode them into numeric.

```
# Selecting columns with object (string) data type as categorical variables
category_var <- train[, sapply(train, is.character)]



# Printing the number of categorical features in the dataset
cat_var_count <- ncol(category_var)
print(paste("Number of categorical features are:", cat_var_count))
```

```
## [1] "Number of categorical features are: 47"
```

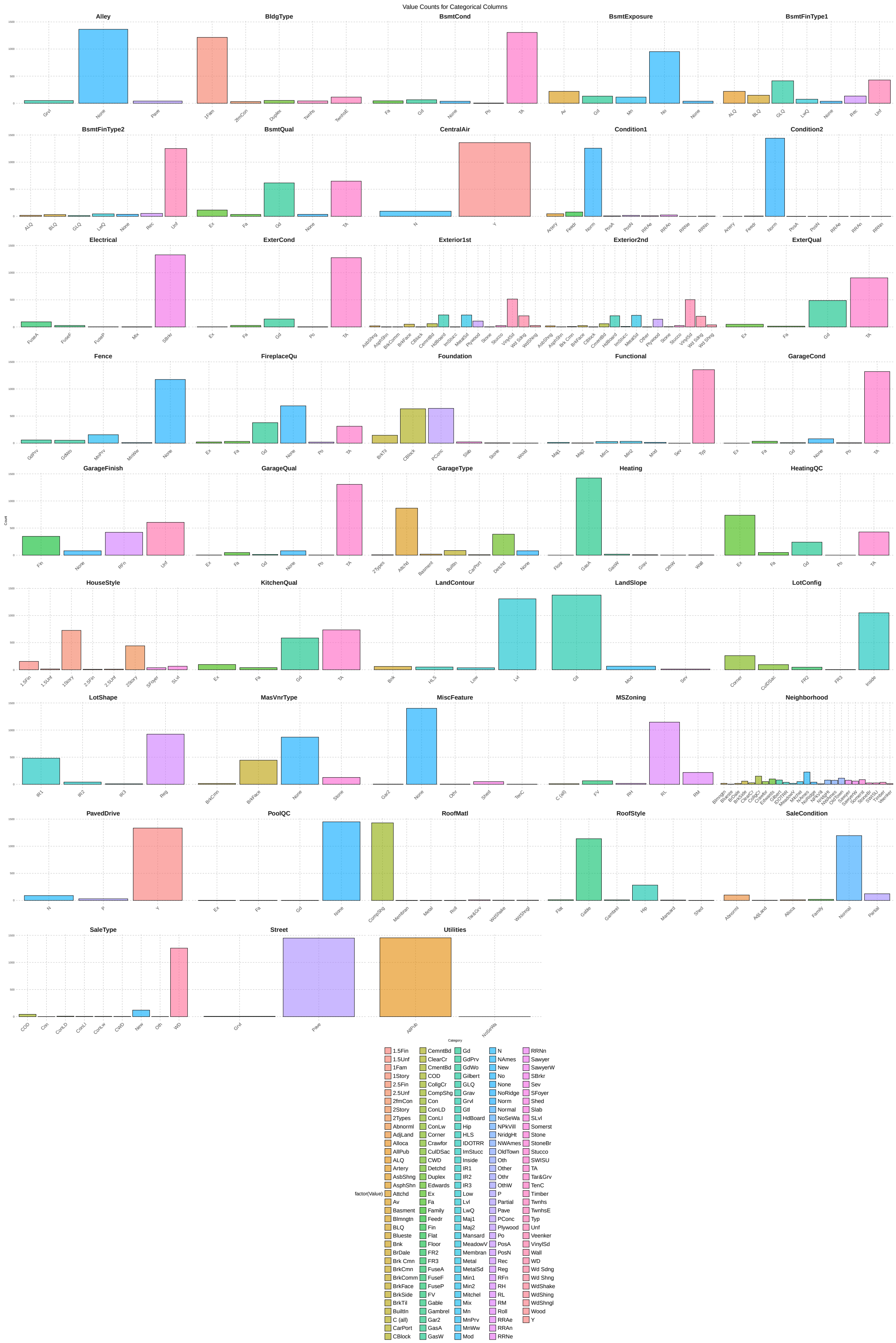
```
train$GarageCars <- as.numeric(train$GarageCars)
train$GarageArea <- as.numeric(train$GarageArea)
train$GarageYrBlt <- as.numeric(train$GarageYrBlt)
```

Make a list of discrete columns

```
# List of columns to convert
columns_to_convert <- c(
  "Alley", "BsmtCond", "BsmtExposure", "BsmtFinType1", "BsmtFinType2", "BsmtQual", "ExterCond", "ExterQu
)
```

Check for the distinct value counts foe each discrete variables

```
## Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.
## i Please use `all_of()` or `any_of()` instead.
##   # Was:
##   data %>% select(columns_to_convert)
##
##   # Now:
##   data %>% select(all_of(columns_to_convert))
##
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```


Value Counts for Categorical Columns
Alley
BldgType
BsmtCond
BsmtExposure
BsmtFinType1
BsmtFinType2
BsmtQual
CentralAir
Condition1
Condition2
Electrical
ExterCond
Exterior1st
Exterior2nd
ExterQual
Fence
FireplaceQu
Foundation
Functional
GarageCond
GarageFinish
GarageQual
GarageType
Heating
HeatingQC
HouseStyle
KitchenQual
LandContour
LandSlope
LotConfig
LotShape
MasVnrType
MiscFeature
MSZoning
Neighborhood
PavedDrive
PoolQC
RoofMatl
RoofStyle
SaleCondition
SaleType
Street
Utilities
Count
Category
factor(Value)

Encode them into numeric

```
## Convert the columns to numeric
for (col in columns_to_convert) {

  factors <- factor(train[[col]])
    nn <- as.numeric(factors)

  # Replace the original column with the numeric values in the dataframe
  train[[col]] <- nn
}
```

Make new Features using the older ones

```
train$Total_living_area <-  train$GrLivArea + train$TotalBsmtSF
train$Total_Bath <- train$FullBath + (0.5*train$HalfBath) + train$BsmtFullBath + (0.5 * train$BsmtHalfBa
train$Pool <- ifelse(train$PoolArea > 0, 1, 0)
train$Garage <- ifelse(train$GarageArea > 0, 1, 0)
train$No_Of_Floors <- ifelse(train$`2ndFlrSF` > 0, 2, 1)
train$Overall_Score <- train$OverallCond * train$OverallQual
train$Exter_Score <- train$ExterCond * train$ExterQual
train$Expensive_Misc <- ifelse(train$MiscVal > 600, 1, 0)
train$Luxury_Score <- train$Pool +train$Garage + train$Fireplaces + train$Expensive_Misc + train$Total_
```

```
 # Calculate the differences
train$House_age <- train$YrSold - train$YearBuilt
train$Remodel_age <- train$YrSold - train$YearRemodAdd
```

We created a dataframe with the every columns correlation score with the target vriable and order them

Then we print the top ten features with highest correlation value to get a view of the correlation data.

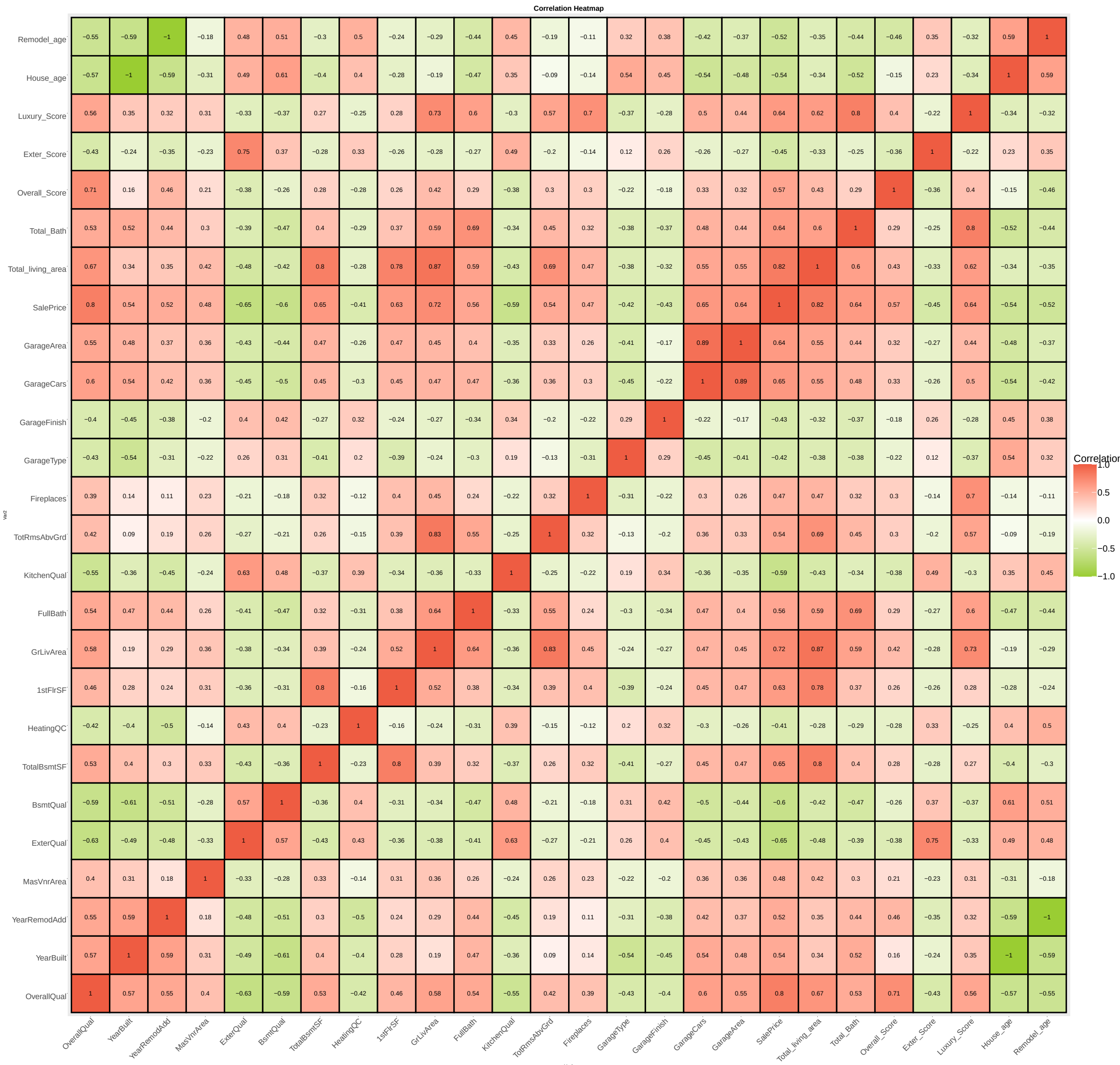
```
##                              variable correlation
## SalePrice                   SalePrice   1.0000000
## Total_living_area Total_living_area   0.8212789
## OverallQual               OverallQual   0.8009497
## GrLivArea                   GrLivArea   0.7205101
## GarageCars                 GarageCars   0.6493205
## ExterQual                   ExterQual  -0.6475079
## TotalBsmtSF               TotalBsmtSF   0.6469894
## Luxury_Score             Luxury_Score   0.6394817
## GarageArea                 GarageArea   0.6369562
## Total_Bath                 Total_Bath   0.6360197
```

Deleting the columns with less that 0,40 correlation score

```
low_correlation <- correlation_df[!abs(correlation_df$correlation) >= .40, ]
```

```
# Get the column names to be removed
cols_to_remove <- low_correlation$variable

# Remove columns from train
train1 <- train[, !names(train) %in% cols_to_remove]
```

Make Correlation Matrix for the remaining numrical columns

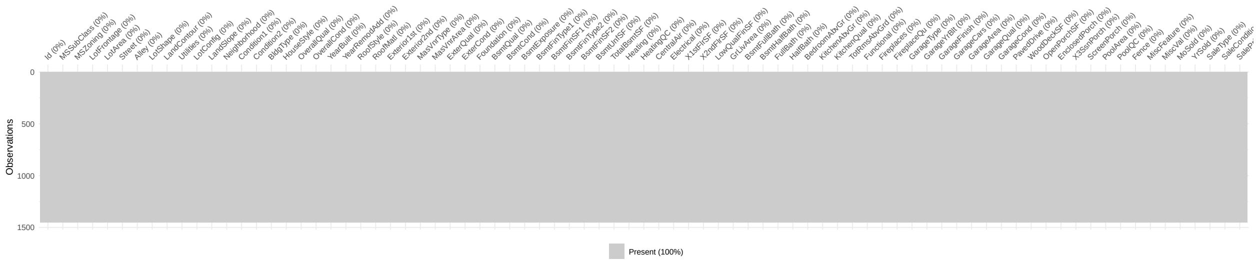


Now that the training data is cleaned and fit for modelling we will do the same for test data as well. We will use the same process for test data as well.

Check for Duplicate samples.

```
## [1] "There are 0 duplicate rows "
## [1] "Number of categorical features are: 43"
## [1] "Number of numerical features are: 38"
## [1] "Number of categorical features with NA are: 22"
## [1] "Number of numerical features with NA are: 11"
```

```
test <- test[complete.cases(test), ]
```



Finally all the NA values have been imputed.

Feature Engineering

```
for (col in columns_to_convert) {

  factors <- factor(test[[col]])
    nn <- as.numeric(factors)

  test[[col]] <- nn
}
```

```
test$Total_living_area <-  test$GrLivArea + test$TotalBsmtSF
test$Total_Bath <- test$FullBath + (0.5*test$HalfBath) + test$BsmtFullBath + (0.5 * test$BsmtHalfBath)
test$Pool <- ifelse(test$PoolArea > 0, 1, 0)
test$Garage <- ifelse(test$GarageArea > 0, 1, 0)
test$No_Of_Floors <- ifelse(test$X2ndFlrSF > 0, 2, 1)
test$Overall_Score <- test$OverallCond * test$OverallQual
test$Exter_Score <- test$ExterCond * test$ExterQual
test$Expensive_Misc <- ifelse(test$MiscVal > 600, 1, 0)
test$Luxury_Score <- test$Pool +test$Garage + test$Fireplaces + test$Expensive_Misc + test$Total_Bath +
```

```
 # Calculate the differences
test$House_age <- test$YrSold - test$YearBuilt
test$Remodel_age <- test$YrSold - test$YearRemodAdd
```

```
##  [1] "OverallQual"       "YearBuilt"         "YearRemodAdd"
##  [4] "MasVnrArea"        "ExterQual"         "BsmtQual"
##  [7] "TotalBsmtSF"       "HeatingQC"         "GrLivArea"
## [10] "FullBath"          "KitchenQual"       "TotRmsAbvGrd"
## [13] "Fireplaces"        "GarageType"        "GarageFinish"
## [16] "GarageCars"        "GarageArea"        "SalePrice"
## [19] "Total_living_area" "Total_Bath"        "Overall_Score"
## [22] "Exter_Score"       "Luxury_Score"      "House_age"
## [25] "Remodel_age"
```

Modelling

```
##  [1] "OverallQual"       "YearBuilt"         "YearRemodAdd"
##  [4] "MasVnrArea"        "ExterQual"         "BsmtQual"
##  [7] "TotalBsmtSF"       "HeatingQC"         "1stFlrSF"
## [10] "GrLivArea"         "FullBath"          "KitchenQual"
## [13] "TotRmsAbvGrd"      "Fireplaces"        "GarageType"
## [16] "GarageFinish"      "GarageCars"        "GarageArea"
## [19] "SalePrice"         "Season"            "Total_living_area"
```

```
## [22] "Total_Bath"        "Overall_Score"     "Exter_Score"
## [25] "Luxury_Score"      "House_age"         "Remodel_age"
## [28] "LogSalesPrice"
```

```
##
## Call:
## lm(formula = SalePrice ~ ., data = df_train)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -61692  -9690  -2353   6099 162072
##
## Coefficients: (2 not defined because of singularities)
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -1.765e+06  7.482e+05  -2.359 0.018480 *
## OverallQual        1.428e+03  8.803e+02   1.623 0.104906
## YearBuilt          1.410e+02  3.727e+02   0.378 0.705300
## YearRemodAdd      -1.204e+02  3.604e+01  -3.340 0.000859 ***
## MasVnrArea         2.473e+01  3.154e+00   7.842 8.60e-15 ***
## ExterQual         -8.639e+03  1.428e+03  -6.051 1.84e-09 ***
## BsmtQual          -3.528e+03  5.656e+02  -6.236 5.89e-10 ***
## TotalBsmtSF        3.502e+00  2.268e+00   1.544 0.122861
## HeatingQC          7.214e+02  3.398e+02   2.123 0.033906 *
## `1stFlrSF`        -2.343e+00  3.222e+00  -0.727 0.467375
## GrLivArea          1.405e+01  2.948e+00   4.766 2.07e-06 ***
## FullBath          -5.327e+03  1.457e+03  -3.657 0.000264 ***
## KitchenQual       -5.401e+03  8.171e+02  -6.610 5.42e-11 ***
## TotRmsAbvGrd       2.741e+01  5.646e+02   0.049 0.961285
## Fireplaces         6.282e+03  1.990e+03   3.157 0.001628 **
## GarageType         1.828e+03  3.379e+02   5.410 7.36e-08 ***
## GarageFinish      -1.295e+03  5.110e+02  -2.534 0.011389 *
## GarageCars        -1.438e+03  1.555e+03  -0.925 0.355183
## GarageArea         1.022e+01  5.198e+00   1.966 0.049481 *
## SeasonSpring       2.678e+01  1.569e+03   0.017 0.986385
## SeasonSummer       1.332e+03  1.338e+03   0.996 0.319395
## SeasonWinter       2.772e+03  1.744e+03   1.590 0.112072
## Total_living_area         NA         NA      NA       NA
## Total_Bath         7.164e+03  1.955e+03   3.665 0.000256 ***
## Overall_Score     -2.559e+02  1.008e+02  -2.539 0.011207 *
## Exter_Score       -3.508e+02  1.892e+02  -1.854 0.063897 .
## Luxury_Score      -6.774e+03  1.671e+03  -4.055 5.28e-05 ***
## House_age          2.829e+02  3.708e+02   0.763 0.445635
## Remodel_age               NA         NA      NA       NA
## LogSalesPrice      1.626e+05  3.700e+03  43.944  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18430 on 1427 degrees of freedom
## Multiple R-squared:  0.9434, Adjusted R-squared:  0.9423
## F-statistic: 880.8 on 27 and 1427 DF,  p-value: < 2.2e-16
```

```
##      rstudent unadjusted p-value Bonferroni p
## 1168 9.359693         3.0008e-20   4.3662e-17
## 897  8.389364         1.1606e-16   1.6887e-13
## 1045 8.085955         1.3052e-15   1.8990e-12
```

```
## 802  7.632556         4.1898e-14   6.0962e-11
## 441  6.707385         2.8484e-11   4.1444e-08
## 915  6.474846         1.3025e-10   1.8952e-07
## 496  5.715281         1.3316e-08   1.9375e-05
## 768  5.505035         4.3711e-08   6.3600e-05
## 967  5.467971         5.3677e-08   7.8100e-05
## 31   5.034521         5.4012e-07   7.8587e-04
```

```
##
## Call:
## lm(formula = LogSalesPrice ~ OverallQual + Total_living_area +
##     GrLivArea + Luxury_Score + GarageCars + Total_Bath + TotalBsmtSF +
##     Overall_Score, data = df_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.85705 -0.07434  0.01620  0.09175  0.52170
##
## Coefficients: (1 not defined because of singularities)
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        1.045e+01  1.967e-02 531.315  < 2e-16 ***
## OverallQual        7.281e-02  5.602e-03  12.997  < 2e-16 ***
## Total_living_area  2.276e-04  1.242e-05  18.327  < 2e-16 ***
## GrLivArea         -7.840e-05  1.940e-05  -4.042 5.59e-05 ***
## Luxury_Score       2.615e-02  5.798e-03   4.511 6.98e-06 ***
## GarageCars         9.265e-02  7.091e-03  13.066  < 2e-16 ***
## Total_Bath         6.788e-02  9.071e-03   7.483 1.26e-13 ***
## TotalBsmtSF               NA         NA      NA       NA
## Overall_Score      6.299e-03  6.424e-04   9.806  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1516 on 1447 degrees of freedom
## Multiple R-squared:  0.8543, Adjusted R-squared:  0.8536
## F-statistic:  1212 on 7 and 1447 DF,  p-value: < 2.2e-16
```

```
##      rstudent unadjusted p-value Bonferroni p
## 31  -5.730094         1.2195e-08   1.7744e-05
## 632 -5.601817         2.5355e-08   3.6892e-05
## 496 -5.348931         1.0273e-07   1.4947e-04
## 967 -4.452374         9.1450e-06   1.3306e-02
## 811 -4.332742         1.5743e-05   2.2906e-02
## 411 -4.185461         3.0174e-05   4.3904e-02
```

```
##
## Call:
## lm(formula = LogSalesPrice ~ ., data = subset_quality)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.88435 -0.11513  0.01762  0.12504  0.62548
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   11.0587162  0.0621509 177.933  < 2e-16 ***
```

```
## OverallQual    0.1181273  0.0075126  15.724  < 2e-16 ***
## ExterQual     -0.0737027  0.0143018  -5.153 2.91e-07 ***
## BsmtQual      -0.0320758  0.0053761  -5.966 3.05e-09 ***
## KitchenQual   -0.0353147  0.0081787  -4.318 1.68e-05 ***
## Overall_Score  0.0045388  0.0008349   5.436 6.38e-08 ***
## Exter_Score    0.0068211  0.0019147   3.562 0.000379 ***
## Luxury_Score   0.0913089  0.0042310  21.581  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1904 on 1447 degrees of freedom
## Multiple R-squared:  0.7703, Adjusted R-squared:  0.7692
## F-statistic: 693.1 on 7 and 1447 DF,  p-value: < 2.2e-16
```

```
##      rstudent unadjusted p-value Bonferroni p
## 496 -4.690178         2.9875e-06    0.0043468
## 704 -4.506804         7.1116e-06    0.0103470
## 31  -4.426974         1.0274e-05    0.0149490
## 811 -4.223086         2.5602e-05    0.0372510
```

```
##
## Call:
## lm(formula = LogSalesPrice ~ ., data = subset_utility)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.93684 -0.10343  0.00063  0.11841  0.88598
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.136e+01  4.407e-02 257.844  < 2e-16 ***
## BsmtQual     -3.549e-02  5.369e-03  -6.610 5.40e-11 ***
## TotalBsmtSF   2.684e-04  1.564e-05  17.163  < 2e-16 ***
## FullBath      7.325e-02  1.396e-02   5.248 1.76e-07 ***
## GarageType   -1.683e-02  3.248e-03  -5.183 2.50e-07 ***
## GarageFinish -2.873e-02  5.169e-03  -5.558 3.24e-08 ***
## GarageCars    8.201e-02  1.644e-02   4.990 6.79e-07 ***
## GarageArea    2.670e-04  5.554e-05   4.807 1.69e-06 ***
## Total_Bath    1.208e-01  1.017e-02  11.884  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2025 on 1446 degrees of freedom
## Multiple R-squared:  0.7403, Adjusted R-squared:  0.7389
## F-statistic: 515.3 on 8 and 1446 DF,  p-value: < 2.2e-16
```

```
##      rstudent unadjusted p-value Bonferroni p
## 496 -4.677403         3.1769e-06    0.0046224
## 676 -4.676510         3.1906e-06    0.0046423
## 915 -4.547086         5.8938e-06    0.0085755
## 31  -4.427934         1.0230e-05    0.0148840
## 186  4.425006         1.0368e-05    0.0150850
## 632 -4.347378         1.4742e-05    0.0214490
```

```
##
## Call:
```

```
## lm(formula = LogSalesPrice ~ ., data = subset_age)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.08190 -0.14074 -0.00805  0.13260  0.86300
##
## Coefficients: (1 not defined because of singularities)
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.826e+01  9.590e+00   1.904   0.0571 .
## YearBuilt    -7.807e-03  4.793e-03  -1.629   0.1036
## YearRemodAdd  4.682e-03  3.914e-04  11.963  < 2e-16 ***
## GarageType   -3.077e-02  4.005e-03  -7.681 2.89e-14 ***
## GarageFinish -5.516e-02  6.094e-03  -9.052  < 2e-16 ***
## GarageCars    1.164e-01  1.929e-02   6.035 2.02e-09 ***
## GarageArea    4.808e-04  6.494e-05   7.404 2.24e-13 ***
## House_age    -8.434e-03  4.775e-03  -1.766   0.0775 .
## Remodel_age          NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2415 on 1447 degrees of freedom
## Multiple R-squared:  0.6304, Adjusted R-squared:  0.6286
## F-statistic: 352.6 on 7 and 1447 DF,  p-value: < 2.2e-16
```

```
##      rstudent unadjusted p-value Bonferroni p
## 915 -4.521816         6.6318e-06    0.0096493
```

```
##
## Call:
## lm(formula = LogSalesPrice ~ OverallQual + YearBuilt + YearRemodAdd +
##     MasVnrArea + ExterQual + BsmtQual + TotalBsmtSF + HeatingQC +
##     `1stFlrSF` + GrLivArea + FullBath + KitchenQual + TotRmsAbvGrd +
##     Fireplaces + GarageType + GarageFinish + GarageCars + GarageArea,
##     data = df_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.86019 -0.07194  0.00841  0.08804  0.51374
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.532e+00  5.957e-01   5.929 3.80e-09 ***
## OverallQual   7.080e-02  4.985e-03  14.203  < 2e-16 ***
## YearBuilt     1.619e-03  2.208e-04   7.332 3.77e-13 ***
## YearRemodAdd  2.170e-03  2.611e-04   8.311  < 2e-16 ***
## MasVnrArea    1.831e-05  2.473e-05   0.740  0.45933
## ExterQual     1.007e-02  8.468e-03   1.189  0.23461
## BsmtQual     -6.884e-03  4.368e-03  -1.576  0.11525
## TotalBsmtSF   1.593e-04  1.696e-05   9.392  < 2e-16 ***
## HeatingQC    -8.007e-03  2.651e-03  -3.021  0.00257 **
## `1stFlrSF`    1.726e-06  1.934e-05   0.089  0.92892
## GrLivArea     2.668e-04  1.794e-05  14.868  < 2e-16 ***
## FullBath     -2.948e-02  1.045e-02  -2.821  0.00485 **
## KitchenQual  -1.929e-02  6.384e-03  -3.022  0.00256 **
## TotRmsAbvGrd -3.167e-04  4.427e-03  -0.072  0.94298
```

```
## Fireplaces    6.155e-02  7.180e-03   8.572  < 2e-16 ***
## GarageType   -1.150e-02  2.561e-03  -4.492 7.62e-06 ***
## GarageFinish -1.500e-03  3.896e-03  -0.385  0.70018
## GarageCars    1.609e-02  1.203e-02   1.338  0.18124
## GarageArea    1.866e-04  4.051e-05   4.605 4.48e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.145 on 1436 degrees of freedom
## Multiple R-squared:  0.8677, Adjusted R-squared:  0.8661
## F-statistic: 523.4 on 18 and 1436 DF,  p-value: < 2.2e-16
```

```
##       rstudent unadjusted p-value Bonferroni p
## 31   -6.022318         2.1807e-09   3.1730e-06
## 632  -5.613959         2.3705e-08   3.4491e-05
## 496  -5.447371         6.0075e-08   8.7409e-05
## 411  -5.306693         1.2918e-07   1.8795e-04
## 1321 -5.067123         4.5638e-07   6.6404e-04
## 811  -4.778072         1.9513e-06   2.8392e-03
## 915  -4.763088         2.0996e-06   3.0549e-03
## 967  -4.746058         2.2812e-06   3.3192e-03
## 463  -4.205706         2.7638e-05   4.0213e-02
```

These are the R squared values of all the models

```
## [1] 0.9423189 0.8535778 0.7691537 0.7388875 0.6286154 0.8660818
```

```
## Adjusted R-squared values:
```

```
## Model All : 0.9423189
## Model  1  : 0.8535778
## Model  2  : 0.7691537
## Model  3  : 0.7388875
## Model  4  : 0.6286154
```

We can clearly se whenn the top correlated variables are used we get the highest R-squared values. But we also get 0.86 R squared value for Model 1 where we use the top correlated variables. We will use the same model on the test data.

Following are the significant predictors for all the models

```
## [[1]]
##  [1] "(Intercept)"   "YearRemodAdd"  "MasVnrArea"    "ExterQual"
##  [5] "BsmtQual"      "HeatingQC"     "GrLivArea"     "FullBath"
##  [9] "KitchenQual"   "Fireplaces"    "GarageType"    "GarageFinish"
## [13] "GarageArea"    "Total_Bath"    "Overall_Score" "Luxury_Score"
## [17] "LogSalesPrice"
##
## [[2]]
## [1] "(Intercept)"       "OverallQual"       "Total_living_area"
## [4] "GrLivArea"         "Luxury_Score"      "GarageCars"
## [7] "Total_Bath"        "Overall_Score"
##
## [[3]]
## [1] "(Intercept)"   "OverallQual"   "ExterQual"     "BsmtQual"
## [5] "KitchenQual"   "Overall_Score" "Exter_Score"   "Luxury_Score"
##
```

```
## [[4]]
## [1] "(Intercept)"  "BsmtQual"     "TotalBsmtSF"  "FullBath"     "GarageType"
## [6] "GarageFinish" "GarageCars"   "GarageArea"   "Total_Bath"
##
## [[5]]
## [1] "YearRemodAdd" "GarageType"   "GarageFinish" "GarageCars"   "GarageArea"
```

test data

```
##
## Call:
## lm(formula = LogSalesPrice ~ OverallQual + YearBuilt + YearRemodAdd +
##     MasVnrArea + ExterQual + BsmtQual + TotalBsmtSF + HeatingQC +
##     `1stFlrSF` + GrLivArea + FullBath + KitchenQual + TotRmsAbvGrd +
##     Fireplaces + GarageType + GarageFinish + GarageCars + GarageArea,
##     data = df_test)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.88288 -0.06995  0.00256  0.08444  0.45268
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.808e+00  5.749e-01   6.624 4.94e-11 ***
## OverallQual   8.643e-02  4.716e-03  18.327  < 2e-16 ***
## YearBuilt     1.601e-03  2.114e-04   7.573 6.53e-14 ***
## YearRemodAdd  2.028e-03  2.633e-04   7.704 2.45e-14 ***
## MasVnrArea    9.951e-06  2.536e-05   0.392 0.694885
## ExterQual     9.614e-03  8.339e-03   1.153 0.249151
## BsmtQual      2.730e-03  4.131e-03   0.661 0.508815
## TotalBsmtSF   1.457e-04  1.491e-05   9.773  < 2e-16 ***
## HeatingQC    -1.068e-02  2.621e-03  -4.076 4.84e-05 ***
## `1stFlrSF`    3.254e-05  1.745e-05   1.865 0.062440 .
## GrLivArea     2.699e-04  1.724e-05  15.653  < 2e-16 ***
## FullBath     -3.009e-02  9.921e-03  -3.033 0.002468 **
## KitchenQual  -2.297e-02  6.486e-03  -3.541 0.000411 ***
## TotRmsAbvGrd -8.917e-03  4.157e-03  -2.145 0.032113 *
## Fireplaces    5.513e-02  6.932e-03   7.953 3.67e-15 ***
## GarageType   -1.287e-02  2.483e-03  -5.182 2.50e-07 ***
## GarageFinish -1.968e-03  3.739e-03  -0.526 0.598713
## GarageCars    2.436e-02  1.187e-02   2.052 0.040324 *
## GarageArea    1.172e-04  4.123e-05   2.843 0.004532 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1414 on 1430 degrees of freedom
## Multiple R-squared:  0.8821, Adjusted R-squared:  0.8806
## F-statistic: 594.5 on 18 and 1430 DF,  p-value: < 2.2e-16
```

We can see the Rsquared value is 0.88 which is very close to the R squared value to the model5 that we are using. We trained the model on the OverallQual, YearBuilt , YearRemodAdd , MasVnrArea , ExterQual , BsmtQual , TotalBsmtSF , HeatingQC , `1stFlrSF` , GrLivArea , FullBath , KitchenQual , TotRmsAbvGrd , Fireplaces , GarageType , GarageFinish ,GarageCars , GarageArea.

```
##        rstudent unadjusted p-value Bonferroni p
```

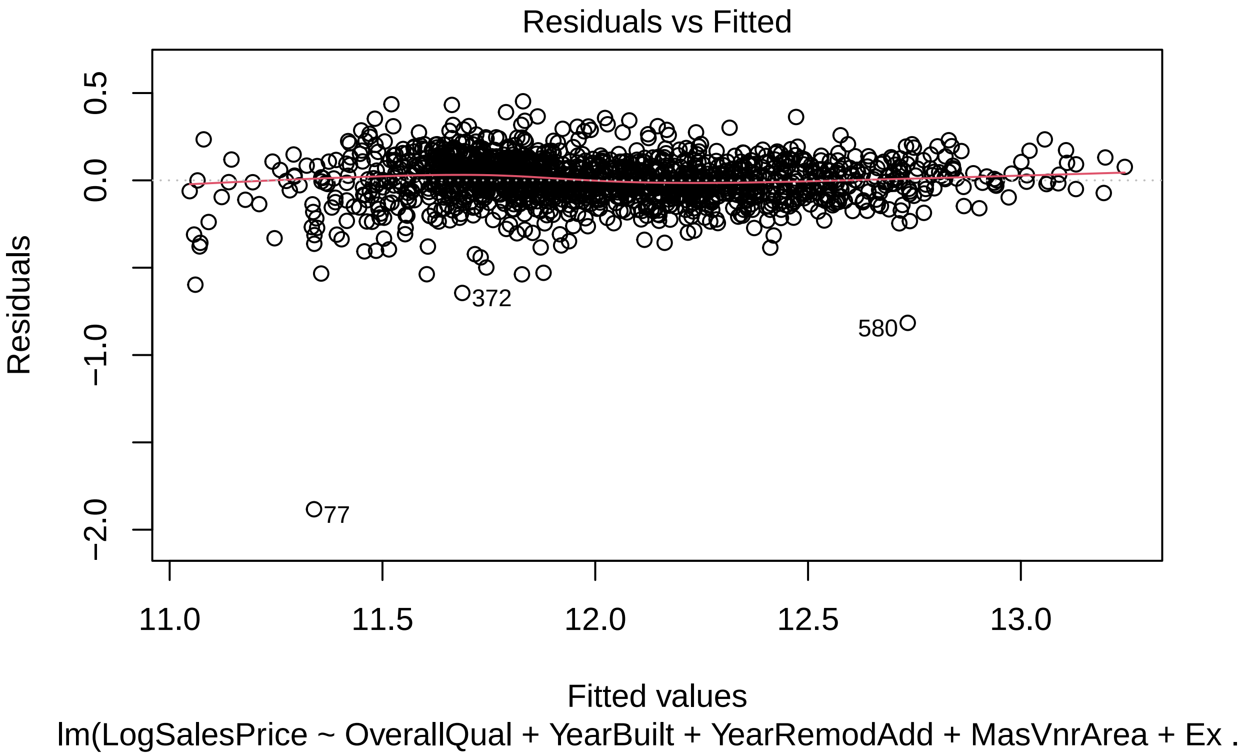
```
## 77   -14.492427         1.6465e-44   2.3858e-41
## 580   -5.952564         3.3164e-09   4.8054e-06
## 372   -4.641595         3.7744e-06   5.4691e-03
## 1411  -4.278893         2.0032e-05   2.9026e-02
```

Evaluation

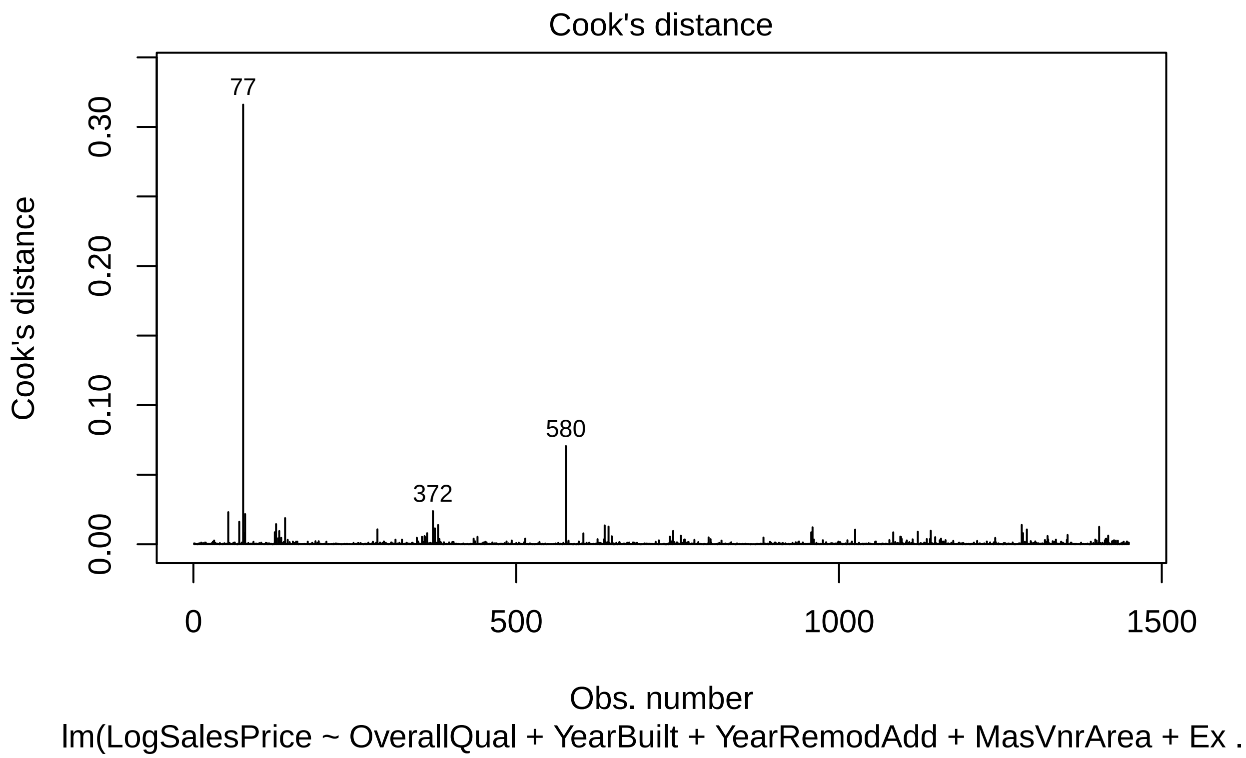
Predicting the model on new data.

Calculating RMSE

```
## RMSE: 0.1404242
```



- The above graph is the residuals vs fitted graph. The residuals mostly cluster around the zero line, which suggests that the model generally fits well across the range of predictions. -Notable outliers are labeled (e.g., 372, 77, 580).
- There's no clear non-linear pattern



Use predictions from your final model to compare suburbs which have shown varying growth.

Our Regression equation is LogSalePrice = LogSalePrice = 3.808 + 0.08643 × OverallQual + 0.001601 × YearBuilt + … + 0.0001172 × GarageArea .

Now we can put the data for these rows for each neighbourhood and we can find the best suburbs and compare all of them after converting them back from log value.

Conclusions and recomendations

- We loaded the data. _ we imputed the missing values.
- For numerical variables we grouped them and then replaced with median of the group
- For discrete variables we encode them into numeric.
- We did some more feature engineering and added some variables as well.
- We used linear regrssion models to predict the sales prices.
- we found the best model with 0.88 Rsquared and 0.14 RMSE.
- Overall Quality , year built and Living area were the most significant factors.
- further we can explorew the use of other regressors like random forests, or multiple linear regression.

Reference

References: Big vote of thanks to all the references mentioned below here. Without which I would have not successfully able to complete linear modelling for multivariable data set. https://www.youtube.com/playlist?list=PLZoTAELRMXVPQyArDHyQVjQxjj_YmEuO9 https://github.com/krishnaik06/Advanced-House-Price-Prediction- http://jse.amstat.org/v19n3/decock.pdf https://www.geeksforgeeks.org/label-encoding-in-r-programming/