STAT 420: Data Analysis Project - Predict the price of your dream home

Anupama Agrahari, Dhanendra Singh, Naveen Kumar Palani, Shanthakumar Subramanian

Team

We are team of four. Please find below our names and Net ID in alphabetical order.

FirstName	LastName	NetID
Anupama	Agrahari	anupama3
Dhanendra	Singh	disingh2
Naveen Kumar	Palani	npalani3
Shanthakumar	Subramanian	SS81

Introduction

About the Project

This data analysis project is inspired from the House Prices: Advanced Regression Techniques competition held in Kaggle.

In this project we are building a model to predict the SalePrice of each home, based on the 80 explanatory variables describing (almost) every aspect of residential homes in Ames, Iowa.

Data Set

Source

Below are the details about our dataset. We are using house prie data available at https://www.kaggle.com/c/house-prices-advanced-regression-techniques (https://www.kaggle.com/c/house-prices-advanced-regression-techniques) . We will use regression techniques to predict the SalePrice.

Background

The Ames Housing dataset was compiled by Dean De Cock for use in data science education. It's an incredible alternative for data scientists looking for a modernized and expanded version of the often cited Boston Housing dataset.

File descriptions

List of files used in the project

- train.csv - the training set
- test.csv - the test set
- data_description.txt - full description of each column, originally prepared by Dean De Cock but lightly edited to match the column names used here

Data fields

Here's a brief description of different fields of the house data set.

- SalePrice - The property's sale price in dollars. This is the target variable that we are trying to predict.
- MSSubClass: The building class
- MSZoning: The general zoning classification
- LotFrontage: Linear feet of street connected to property
- LotArea: Lot size in square feet
- Street: Type of road access
- Alley: Type of alley access
- LotShape: General shape of property
- LandContour: Flatness of the property
- Utilities: Type of utilities available
- LotConfig: Lot configuration
- LandSlope: Slope of property
- Neighborhood: Physical locations within Ames city limits
- Condition1: Proximity to main road or railroad
- Condition2: Proximity to main road or railroad (if a second is present)
- BldgType: Type of dwelling
- HouseStyle: Style of dwelling
- OverallQual: Overall material and finish quality
- OverallCond: Overall condition rating
- YearBuilt: Original construction date
- YearRemodAdd: Remodel date
- RoofStyle: Type of roof
- RoofMatl: Roof material
- Exterior1st: Exterior covering on house
- Exterior2nd: Exterior covering on house (if more than one material)
- MasVnrType: Masonry veneer type
- MasVnrArea: Masonry veneer area in square feet
- ExterQual: Exterior material quality
- ExterCond: Present condition of the material on the exterior
- Foundation: Type of foundation
- BsmtQual: Height of the basement
- BsmtCond: General condition of the basement
- BsmtExposure: Walkout or garden level basement walls
- BsmtFinType1: Quality of basement finished area
- BsmtFinSF1: Type 1 finished square feet
- BsmtFinType2: Quality of second finished area (if present)
- BsmtFinSF2: Type 2 finished square feet
- BsmtUnfSF: Unfinished square feet of basement area
- TotalBsmtSF: Total square feet of basement area

- Heating: Type of heating
- HeatingQC: Heating quality and condition
- CentralAir: Central air conditioning
- Electrical: Electrical system
- X1stFlrSF: First Floor square feet
- X2ndFlrSF: Second floor square feet
- LowQualFinSF: Low quality finished square feet (all floors)
- GrLivArea: Above grade (ground) living area square feet
- BsmtFullBath: Basement full bathrooms
- BsmtHalfBath: Basement half bathrooms
- FullBath: Full bathrooms above grade
- HalfBath: Half baths above grade
- Bedroom: Number of bedrooms above basement level
- Kitchen: Number of kitchens
- KitchenQual: Kitchen quality
- TotRmsAbvGrd: Total rooms above grade (does not include bathrooms)
- Functional: Home functionality rating
- Fireplaces: Number of fireplaces
- FireplaceQu: Fireplace quality
- GarageType: Garage location
- GarageYrBlt: Year garage was built
- GarageFinish: Interior finish of the garage
- GarageCars: Size of garage in car capacity
- GarageArea: Size of garage in square feet
- GarageQual: Garage quality
- GarageCond: Garage condition
- PavedDrive: Paved driveway
- WoodDeckSF: Wood deck area in square feet
- OpenPorchSF: Open porch area in square feet
- EnclosedPorch: Enclosed porch area in square feet
- 3SsnPorch: Three season porch area in square feet
- ScreenPorch: Screen porch area in square feet
- PoolArea: Pool area in square feet
- PoolQC: Pool quality
- Fence: Fence quality
- MiscFeature: Miscellaneous feature not covered in other categories
- MiscVal: $Value of miscellaneous feature
- MoSold: Month Sold
- YrSold: Year Sold
- SaleType: Type of sale
- SaleCondition: Condition of sale

Data Preparation

Data Import

```
#Libraries used

library(lmtest)
library(broom)
library(formattable)
library(faraway)
library(ggplot2)
library(leaps)
library(caret)
library(tinytex)
```

Importing test and train data from the files

```
#Import train and test dataset from the source files.

train = read.csv("train.csv", stringsAsFactors = FALSE)
test = read.csv("test.csv", stringsAsFactors = FALSE)
```

Checking the number of columns and observation in test and train data

```
dim(train)
```

```
## [1] 1460   81
```

```
dim(test)
```

```
## [1] 1459   80
```

From the dimension of test and train data, we could see that the test data has one less column than the train, as the **Sale Price** column would not be availble in test data and that's what we will be predicting as part of this project.

Data Cleansing

```
#Displaying the structure of train dataset

str(train)
```

```
## 'data.frame':    1460 obs. of  81 variables:
##  $ Id           : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ MSSubClass   : int  60 20 60 70 60 50 20 60 50 190 ...
##  $ MSZoning     : chr  "RL" "RL" "RL" "RL" ...
##  $ LotFrontage  : int  65 80 68 60 84 85 75 NA 51 50 ...
##  $ LotArea      : int  8450 9600 11250 9550 14260 14115 10084 10382 6120 7420 ...
##  $ Street       : chr  "Pave" "Pave" "Pave" "Pave" ...
##  $ Alley        : chr  NA NA NA NA ...
##  $ LotShape     : chr  "Reg" "Reg" "IR1" "IR1" ...
##  $ LandContour  : chr  "Lvl" "Lvl" "Lvl" "Lvl" ...
##  $ Utilities    : chr  "AllPub" "AllPub" "AllPub" "AllPub" ...
##  $ LotConfig    : chr  "Inside" "FR2" "Inside" "Corner" ...
##  $ LandSlope    : chr  "Gtl" "Gtl" "Gtl" "Gtl" ...
##  $ Neighborhood : chr  "CollgCr" "Veenker" "CollgCr" "Crawfor" ...
##  $ Condition1   : chr  "Norm" "Feedr" "Norm" "Norm" ...
##  $ Condition2   : chr  "Norm" "Norm" "Norm" "Norm" ...
##  $ BldgType     : chr  "1Fam" "1Fam" "1Fam" "1Fam" ...
##  $ HouseStyle   : chr  "2Story" "1Story" "2Story" "2Story" ...
##  $ OverallQual  : int  7 6 7 7 8 5 8 7 7 5 ...
##  $ OverallCond  : int  5 8 5 5 5 5 5 6 5 6 ...
##  $ YearBuilt    : int  2003 1976 2001 1915 2000 1993 2004 1973 1931 1939 ...
##  $ YearRemodAdd : int  2003 1976 2002 1970 2000 1995 2005 1973 1950 1950 ...
##  $ RoofStyle    : chr  "Gable" "Gable" "Gable" "Gable" ...
##  $ RoofMatl     : chr  "CompShg" "CompShg" "CompShg" "CompShg" ...
##  $ Exterior1st  : chr  "VinylSd" "MetalSd" "VinylSd" "Wd Sdng" ...
##  $ Exterior2nd  : chr  "VinylSd" "MetalSd" "VinylSd" "Wd Shng" ...
##  $ MasVnrType   : chr  "BrkFace" "None" "BrkFace" "None" ...
##  $ MasVnrArea   : int  196 0 162 0 350 0 186 240 0 0 ...
##  $ ExterQual    : chr  "Gd" "TA" "Gd" "TA" ...
##  $ ExterCond    : chr  "TA" "TA" "TA" "TA" ...
##  $ Foundation   : chr  "PConc" "CBlock" "PConc" "BrkTil" ...
##  $ BsmtQual     : chr  "Gd" "Gd" "Gd" "TA" ...
##  $ BsmtCond     : chr  "TA" "TA" "TA" "Gd" ...
##  $ BsmtExposure : chr  "No" "Gd" "Mn" "No" ...
##  $ BsmtFinType1 : chr  "GLQ" "ALQ" "GLQ" "ALQ" ...
##  $ BsmtFinSF1   : int  706 978 486 216 655 732 1369 859 0 851 ...
##  $ BsmtFinType2 : chr  "Unf" "Unf" "Unf" "Unf" ...
##  $ BsmtFinSF2   : int  0 0 0 0 0 0 0 32 0 0 ...
##  $ BsmtUnfSF    : int  150 284 434 540 490 64 317 216 952 140 ...
##  $ TotalBsmtSF  : int  856 1262 920 756 1145 796 1686 1107 952 991 ...
##  $ Heating      : chr  "GasA" "GasA" "GasA" "GasA" ...
##  $ HeatingQC    : chr  "Ex" "Ex" "Ex" "Gd" ...
##  $ CentralAir   : chr  "Y" "Y" "Y" "Y" ...
##  $ Electrical   : chr  "SBrkr" "SBrkr" "SBrkr" "SBrkr" ...
##  $ X1stFlrSF    : int  856 1262 920 961 1145 796 1694 1107 1022 1077 ...
##  $ X2ndFlrSF    : int  854 0 866 756 1053 566 0 983 752 0 ...
##  $ LowQualFinSF : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ GrLivArea    : int  1710 1262 1786 1717 2198 1362 1694 2090 1774 1077 ...
##  $ BsmtFullBath : int  1 0 1 1 1 1 1 1 0 1 ...
##  $ BsmtHalfBath : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ FullBath     : int  2 2 2 1 2 1 2 2 2 1 ...
##  $ HalfBath     : int  1 0 1 0 1 1 0 1 0 0 ...
##  $ BedroomAbvGr : int  3 3 3 3 4 1 3 3 2 2 ...
##  $ KitchenAbvGr : int  1 1 1 1 1 1 1 1 2 2 ...
##  $ KitchenQual  : chr  "Gd" "TA" "Gd" "Gd" ...
##  $ TotRmsAbvGrd : int  8 6 6 7 9 5 7 7 8 5 ...
##  $ Functional   : chr  "Typ" "Typ" "Typ" "Typ" ...
##  $ Fireplaces   : int  0 1 1 1 1 0 1 2 2 2 ...
##  $ FireplaceQu  : chr  NA "TA" "TA" "Gd" ...
##  $ GarageType   : chr  "Attchd" "Attchd" "Attchd" "Detchd" ...
##  $ GarageYrBlt  : int  2003 1976 2001 1998 2000 1993 2004 1973 1931 1939 ...
##  $ GarageFinish : chr  "RFn" "RFn" "RFn" "Unf" ...
##  $ GarageCars   : int  2 2 2 3 3 2 2 2 2 1 ...
##  $ GarageArea   : int  548 460 608 642 836 480 636 484 468 205 ...
##  $ GarageQual   : chr  "TA" "TA" "TA" "TA" ...
##  $ GarageCond   : chr  "TA" "TA" "TA" "TA" ...
##  $ PavedDrive   : chr  "Y" "Y" "Y" "Y" ...
##  $ WoodDeckSF   : int  0 298 0 0 192 40 255 235 90 0 ...
##  $ OpenPorchSF  : int  61 0 42 35 84 30 57 204 0 4 ...
##  $ EnclosedPorch: int  0 0 0 272 0 0 0 228 205 0 ...
##  $ X3SsnPorch   : int  0 0 0 0 0 320 0 0 0 0 ...
##  $ ScreenPorch  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PoolArea     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PoolQC       : chr  NA NA NA NA ...
##  $ Fence        : chr  NA NA NA NA ...
##  $ MiscFeature  : chr  NA NA NA NA ...
##  $ MiscVal      : int  0 0 0 0 0 700 0 350 0 0 ...
##  $ MoSold       : int  2 5 9 2 12 10 8 11 4 1 ...
##  $ YrSold       : int  2008 2007 2008 2006 2008 2009 2007 2009 2008 2008 ...
##  $ SaleType     : chr  "WD" "WD" "WD" "WD" ...
##  $ SaleCondition: chr  "Normal" "Normal" "Normal" "Abnorml" ...
##  $ SalePrice    : int  208500 181500 223500 140000 250000 143000 307000 200000 129900 118000 ...
```

From the above, `str` display of data, we could see the below issues:

- The data is not completely clean, certain variables contain `NA` values which could cause problems when we do any mathematical operation and while fitting the model.
- The levels of some of the factor variables are not the same across the training set and test set, which might also cause some problems while fitting the model.
- Some of the categoricals variables are listed as numeric, this might have some impact on fitting the model.

We will do the following to rectify the above issues:

- Remove NA and update the values with “None” for character variables.
- Impute/replce NA with either `0`, `Mean`, `Medain` or `Mode` for numeric variables appropriatley.

We should make sure that, the train and test sets have the same factor levels by loading each data set again without converting strings to factors, combining them into one large data set, converting strings to factors for the combined data set and then separating them.

We also can change the data types of the below features from numeric to string.

`MSSubClass`, `OverallCond`, `OverallQual`, `GarageCars`, `YrSold`, `MoSold`

The values for each feature above represent different categories and not different amounts of something. This is easiest to see in the case of MSSubClass, where the numbers encode different categories of houses, such as 2-Story 1946 and Newer (60) and 2-Story 1945 and Older (70). It is harder to discern in a feature like GarageCars, where each value seems to count something (cars) but in actuality represents the garage capacity, and therefore represents a category.

Step 1

Removing the target variable SalePrice, which is not found in test data set.

```
## Removing the target variable saleprice

SalePrice = train$SalePrice
train$SalePrice = NULL
```

Step 2

Combining test and training data set and changing numeric to factor variable.

```
# Combine data sets

house_data = rbind(train,test)

##Change Numeric to factor

house_data$MSSubClass = as.character(house_data$MSSubClass)
house_data$OverallCond = as.character(house_data$OverallCond)
house_data$OverallQual = as.character(house_data$OverallQual)
house_data$GarageCars = as.character(house_data$GarageCars)
house_data$YrSold = as.character(house_data$YrSold)
house_data$MoSold = as.character(house_data$MoSold)
```

Step 3

We will now convert `character` columns to factor and will replace NA values with “None”.

```
## Replacing the NA in character variable with None.

for (col in colnames(house_data)){
  if (typeof(house_data[,col]) == "character"){
    new_col = house_data[,col]
    new_col[is.na(new_col)] = "None"
    house_data[col] = as.factor(new_col)
  }
}
```

Now the `factor variable` should be common across test and train dataset.

Step 4

Lets try to find the missing values in `integer variable` and fix it.

```
## Checking for missing data values in all integer variables

Missing_indices = sapply(house_data,function(x) sum(is.na(x)))
Missing_Summary = data.frame(index = names(house_data),Missing_Values=Missing_indices)
Missing_Summary[Missing_Summary$Missing_Values > 0,]
```

```
##                     index Missing_Values
## LotFrontage   LotFrontage            486
## MasVnrArea     MasVnrArea             23
## BsmtFinSF1     BsmtFinSF1              1
## BsmtFinSF2     BsmtFinSF2              1
## BsmtUnfSF       BsmtUnfSF              1
## TotalBsmtSF   TotalBsmtSF              1
## BsmtFullBath BsmtFullBath              2
## BsmtHalfBath BsmtHalfBath              2
## GarageYrBlt   GarageYrBlt            159
## GarageArea     GarageArea              1
```

Above are the list of variable having NA values, in the below section, we will fix those values with relavant values

- Imputed missing values of `MasVnrArea` with its `mean` .
- Imputed missing values of `LotFrontage` with its `median`
- Imputed the missing values of below with `0` .

- `GarageYrBlt` , `BsmtFullBath` , `BsmtHalfBath` , `BsmtUnfSF` , `BsmtFinSF1` , `BsmtFinSF2` , `GarageArea` , `GarageCars` , `TotalBsmtSF`

```
## Impute Mean
house_data$MasVnrArea[which(is.na(house_data$MasVnrArea))] = mean(house_data$MasVnrArea,na.rm=T)

## Impute Median

house_data$LotFrontage[which(is.na(house_data$LotFrontage))] = median(house_data$LotFrontage,na.rm = T)

## Impute 0

house_data$GarageYrBlt[which(is.na(house_data$GarageYrBlt))] = 0

house_data$BsmtFullBath[which(is.na(house_data$BsmtFullBath ))] = 0

house_data$BsmtHalfBath[which(is.na(house_data$BsmtHalfBath ))] = 0

house_data$BsmtUnfSF[which(is.na(house_data$BsmtUnfSF))] = 0

house_data$BsmtFinSF1[which(is.na(house_data$BsmtFinSF1))] = 0

house_data$BsmtFinSF2[which(is.na(house_data$BsmtFinSF2))] = 0

house_data$GarageArea[which(is.na(house_data$GarageArea))] = 0

house_data$GarageCars[which(is.na(house_data$GarageCars))] = 0

house_data$TotalBsmtSF[which(is.na(house_data$TotalBsmtSF))] = 0
```

Step 5

We will split the train and test data now as the data issues are fixed.

```
# Separate out our train and test sets

train = house_data[1:nrow(train),]
train$SalePrice = SalePrice
test = house_data[(nrow(train)+1):nrow(house_data),]
```

Let’s now have a look at the train data set

```
#Displaying the train dataset structure after data cleansing.

str(train)
```

```
## 'data.frame':    1460 obs. of  81 variables:
##  $ Id           : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ MSSubClass   : Factor w/ 16 levels "120","150","160",..: 11 6 11 12 11 10 6 11 10 5 ...
##  $ MSZoning     : Factor w/ 6 levels "C (all)","FV",..: 5 5 5 5 5 5 5 5 6 5 ...
##  $ LotFrontage  : int  65 80 68 60 84 85 75 68 51 50 ...
##  $ LotArea      : int  8450 9600 11250 9550 14260 14115 10084 10382 6120 7420 ...
##  $ Street       : Factor w/ 2 levels "Grvl","Pave": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Alley        : Factor w/ 3 levels "Grvl","None",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ LotShape     : Factor w/ 4 levels "IR1","IR2","IR3",..: 4 4 1 1 1 1 4 1 4 4 ...
##  $ LandContour  : Factor w/ 4 levels "Bnk","HLS","Low",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ Utilities    : Factor w/ 3 levels "AllPub","None",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ LotConfig    : Factor w/ 5 levels "Corner","CulDSac",..: 5 3 5 1 3 5 5 1 5 1 ...
##  $ LandSlope    : Factor w/ 3 levels "Gtl","Mod","Sev": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Neighborhood : Factor w/ 25 levels "Blmngtn","Blueste",..: 6 25 6 7 14 12 21 17 18 4 ...
##  $ Condition1   : Factor w/ 9 levels "Artery","Feedr",..: 3 2 3 3 3 3 3 5 1 1 ...
##  $ Condition2   : Factor w/ 8 levels "Artery","Feedr",..: 3 3 3 3 3 3 3 3 3 1 ...
##  $ BldgType     : Factor w/ 5 levels "1Fam","2fmCon",..: 1 1 1 1 1 1 1 1 1 2 ...
##  $ HouseStyle   : Factor w/ 8 levels "1.5Fin","1.5Unf",..: 6 3 6 6 6 1 3 6 1 2 ...
##  $ OverallQual  : Factor w/ 10 levels "1","10","2","3",..: 8 7 8 8 9 6 9 8 8 6 ...
##  $ OverallCond  : Factor w/ 9 levels "1","2","3","4",..: 5 8 5 5 5 5 5 6 5 6 ...
##  $ YearBuilt    : int  2003 1976 2001 1915 2000 1993 2004 1973 1931 1939 ...
##  $ YearRemodAdd : int  2003 1976 2002 1970 2000 1995 2005 1973 1950 1950 ...
##  $ RoofStyle    : Factor w/ 6 levels "Flat","Gable",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ RoofMatl     : Factor w/ 8 levels "ClyTile","CompShg",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ Exterior1st  : Factor w/ 16 levels "AsbShng","AsphShn",..: 14 9 14 15 14 14 14 7 4 9 ...
##  $ Exterior2nd  : Factor w/ 17 levels "AsbShng","AsphShn",..: 15 9 15 17 15 15 15 7 17 9 ...
##  $ MasVnrType   : Factor w/ 4 levels "BrkCmn","BrkFace",..: 2 3 2 3 2 3 4 4 3 3 ...
##  $ MasVnrArea   : num  196 0 162 0 350 0 186 240 0 0 ...
##  $ ExterQual    : Factor w/ 4 levels "Ex","Fa","Gd",..: 3 4 3 4 3 4 3 4 4 4 ...
##  $ ExterCond    : Factor w/ 5 levels "Ex","Fa","Gd",..: 5 5 5 5 5 5 5 5 5 5 ...
##  $ Foundation   : Factor w/ 6 levels "BrkTil","CBlock",..: 3 2 3 1 3 6 3 2 1 1 ...
##  $ BsmtQual     : Factor w/ 5 levels "Ex","Fa","Gd",..: 3 3 3 5 3 3 1 3 5 5 ...
##  $ BsmtCond     : Factor w/ 5 levels "Fa","Gd","None",..: 5 5 5 2 5 5 5 5 5 5 ...
##  $ BsmtExposure : Factor w/ 5 levels "Av","Gd","Mn",..: 4 2 3 4 1 4 1 3 4 4 ...
##  $ BsmtFinType1 : Factor w/ 7 levels "ALQ","BLQ","GLQ",..: 3 1 3 1 3 3 3 1 7 3 ...
##  $ BsmtFinSF1   : num  706 978 486 216 655 ...
##  $ BsmtFinType2 : Factor w/ 7 levels "ALQ","BLQ","GLQ",..: 7 7 7 7 7 7 7 2 7 7 ...
##  $ BsmtFinSF2   : num  0 0 0 0 0 0 0 32 0 0 ...
##  $ BsmtUnfSF    : num  150 284 434 540 490 64 317 216 952 140 ...
##  $ TotalBsmtSF  : num  856 1262 920 756 1145 ...
##  $ Heating      : Factor w/ 6 levels "Floor","GasA",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ HeatingQC    : Factor w/ 5 levels "Ex","Fa","Gd",..: 1 1 1 3 1 1 1 1 3 1 ...
##  $ CentralAir   : Factor w/ 2 levels "N","Y": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Electrical   : Factor w/ 6 levels "FuseA","FuseF",..: 6 6 6 6 6 6 6 6 2 6 ...
##  $ X1stFlrSF    : int  856 1262 920 961 1145 796 1694 1107 1022 1077 ...
##  $ X2ndFlrSF    : int  854 0 866 756 1053 566 0 983 752 0 ...
##  $ LowQualFinSF : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ GrLivArea    : int  1710 1262 1786 1717 2198 1362 1694 2090 1774 1077 ...
##  $ BsmtFullBath : num  1 0 1 1 1 1 1 1 0 1 ...
##  $ BsmtHalfBath : num  0 1 0 0 0 0 0 0 0 0 ...
##  $ FullBath     : int  2 2 2 1 2 1 2 2 2 1 ...
##  $ HalfBath     : int  1 0 1 0 1 1 0 1 0 0 ...
##  $ BedroomAbvGr : int  3 3 3 3 4 1 3 3 2 2 ...
##  $ KitchenAbvGr : int  1 1 1 1 1 1 1 1 2 2 ...
##  $ KitchenQual  : Factor w/ 5 levels "Ex","Fa","Gd",..: 3 5 3 3 3 5 3 5 5 5 ...
##  $ TotRmsAbvGrd : int  8 6 6 7 9 5 7 7 8 5 ...
##  $ Functional   : Factor w/ 8 levels "Maj1","Maj2",..: 8 8 8 8 8 8 8 8 3 8 ...
##  $ Fireplaces   : int  0 1 1 1 1 0 1 2 2 2 ...
##  $ FireplaceQu  : Factor w/ 6 levels "Ex","Fa","Gd",..: 4 6 6 3 6 4 3 6 6 6 ...
##  $ GarageType   : Factor w/ 7 levels "2Types","Attchd",..: 2 2 2 6 2 2 2 2 6 2 ...
##  $ GarageYrBlt  : num  2003 1976 2001 1998 2000 ...
##  $ GarageFinish : Factor w/ 4 levels "Fin","None","RFn",..: 3 3 3 4 3 4 3 3 4 3 ...
##  $ GarageCars   : Factor w/ 7 levels "0","1","2","3",..: 3 3 3 4 4 3 3 3 3 2 ...
##  $ GarageArea   : num  548 460 608 642 836 480 636 484 468 205 ...
##  $ GarageQual   : Factor w/ 6 levels "Ex","Fa","Gd",..: 6 6 6 6 6 6 6 6 2 3 ...
##  $ GarageCond   : Factor w/ 6 levels "Ex","Fa","Gd",..: 6 6 6 6 6 6 6 6 6 6 ...
##  $ PavedDrive   : Factor w/ 3 levels "N","P","Y": 3 3 3 3 3 3 3 3 3 3 ...
##  $ WoodDeckSF   : int  0 298 0 0 192 40 255 235 90 0 ...
##  $ OpenPorchSF  : int  61 0 42 35 84 30 57 204 0 4 ...
##  $ EnclosedPorch: int  0 0 0 272 0 0 0 228 205 0 ...
##  $ X3SsnPorch   : int  0 0 0 0 0 320 0 0 0 0 ...
##  $ ScreenPorch  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PoolArea     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PoolQC       : Factor w/ 4 levels "Ex","Fa","Gd",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ Fence        : Factor w/ 5 levels "GdPrv","GdWo",..: 5 5 5 5 5 3 5 5 5 5 ...
##  $ MiscFeature  : Factor w/ 5 levels "Gar2","None",..: 2 2 2 2 2 4 2 4 2 2 ...
##  $ MiscVal      : int  0 0 0 0 0 700 0 350 0 0 ...
##  $ MoSold       : Factor w/ 12 levels "1","10","11",..: 5 8 12 5 4 2 11 3 7 1 ...
##  $ YrSold       : Factor w/ 5 levels "2006","2007",..: 3 2 3 1 3 4 2 4 3 3 ...
##  $ SaleType     : Factor w/ 10 levels "COD","Con","ConLD",..: 10 10 10 10 10 10 10 10 10 10 ...
##  $ SaleCondition: Factor w/ 6 levels "Abnorml","AdjLand",..: 5 5 5 1 5 5 5 5 1 5 ...
##  $ SalePrice    : int  208500 181500 223500 140000 250000 143000 307000 200000 129900 118000 ...
```

From the above `str` of train dataset, we could see that the `NA` in the dataset is completely removed now.

Train and Test data Split

We can split the data from the train dataset in 80 - 20 ratio, we can use this 20% data of train dataset to test our fitted model.

```
#Splitting the train dataset into train and test

set.seed(1)
house_trn_idx = sample(1:nrow(train), 1168)
house_trn = train[house_trn_idx,]
house_tst = train[-house_trn_idx,]
```

Exploratory Data Analysis

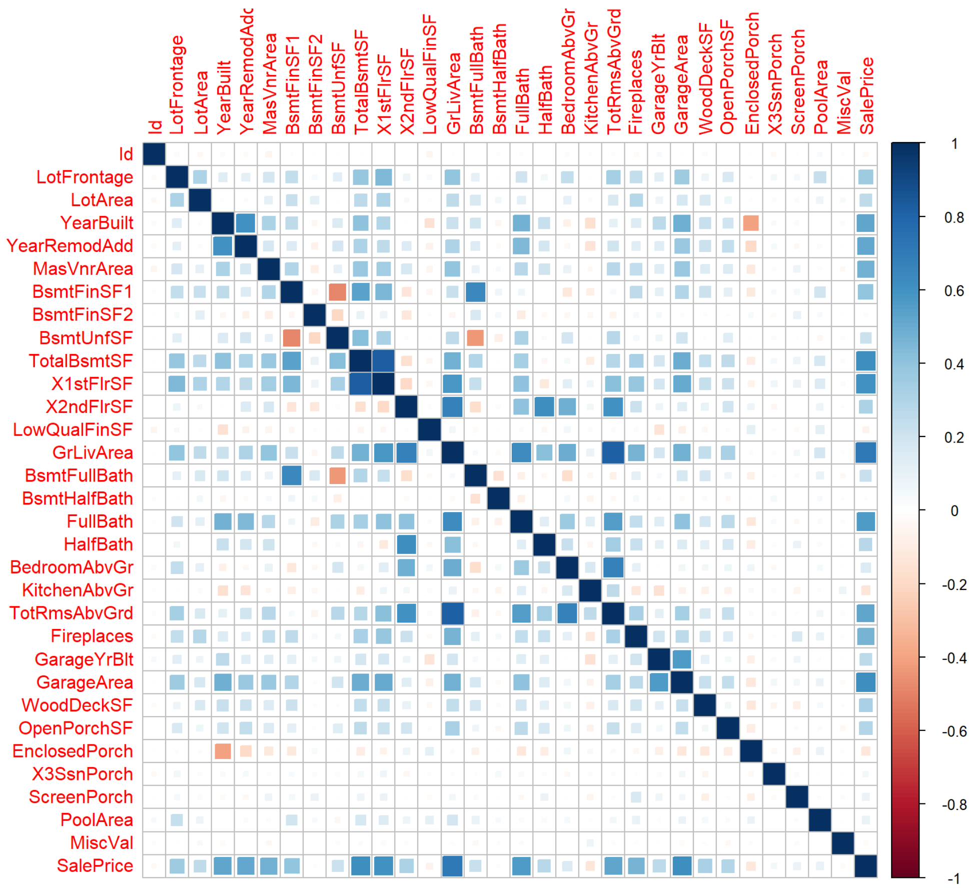
Correlation Matrix

Lets plot the correlation matrix to see which variables have multicollinearity.

```
# Correlation Plot

house_train_num<-house_trn[, sapply(house_trn, is.numeric)]
Correlation<-cor(na.omit(house_train_num))
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
corrplot(Correlation, method = "square")
```



From the above plot, we could see that some varaiables have high correlation between each other, below are the few examples:

- TotalBsmtSF and X1stFlrSF
- BsmtFinSF1 and BsmtFullBath
- X2ndFlrSF and GrLivArea

Correlation with SalePrice

We will explore the Variable which has correlation of more than `0.5` with SalePrice.

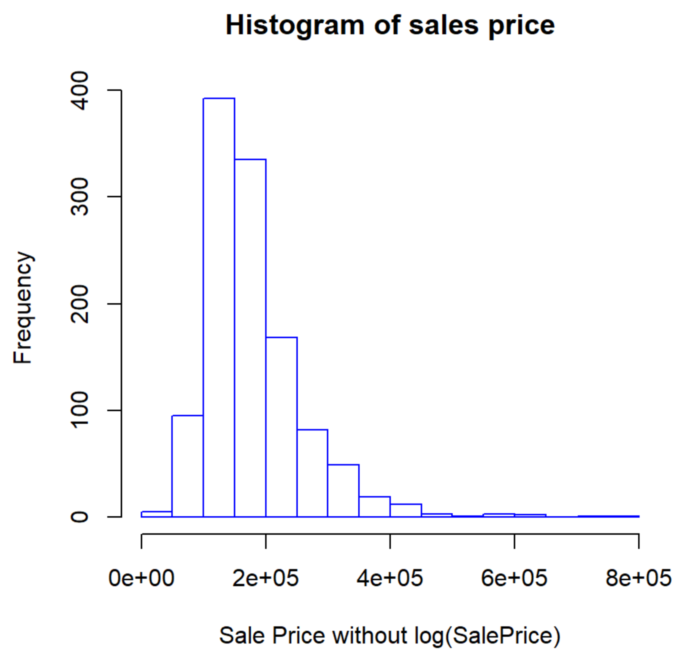
```
#variables having Correlation > 0.5 with SalePrice

for (col in colnames(house_trn)){
   if(is.numeric(house_trn[,col])){
       if( abs(cor(house_trn[,col],house_trn$SalePrice)) > 0.5){
           print(col)
           print( cor(house_trn[,col],house_trn$SalePrice))
         }
   }
}
```

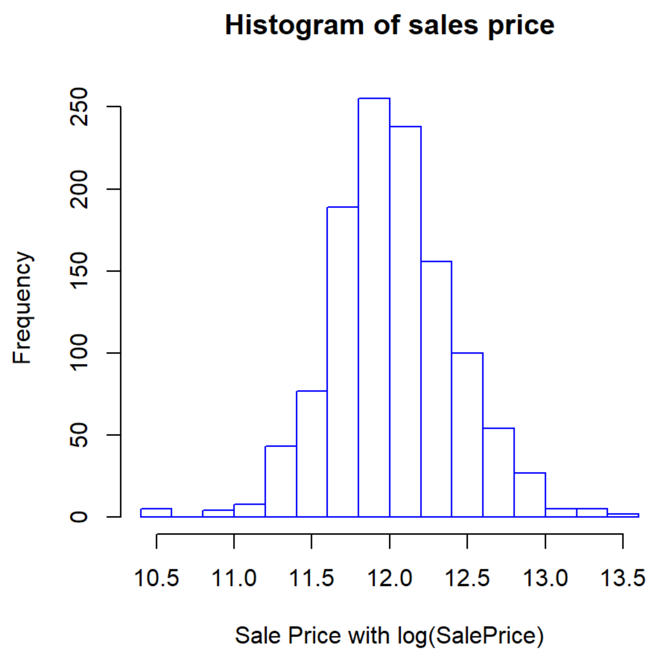
```
## [1] "YearBuilt"
## [1] 0.5250568
## [1] "YearRemodAdd"
## [1] 0.5130786
## [1] "TotalBsmtSF"
## [1] 0.6130396
## [1] "X1stFlrSF"
## [1] 0.6046266
## [1] "GrLivArea"
## [1] 0.7103528
## [1] "FullBath"
## [1] 0.5632221
## [1] "TotRmsAbvGrd"
## [1] 0.525106
## [1] "GarageArea"
## [1] 0.6172792
## [1] "SalePrice"
## [1] 1
```

We can use the variable listed above to fit our model as it has high corelation with sales price.

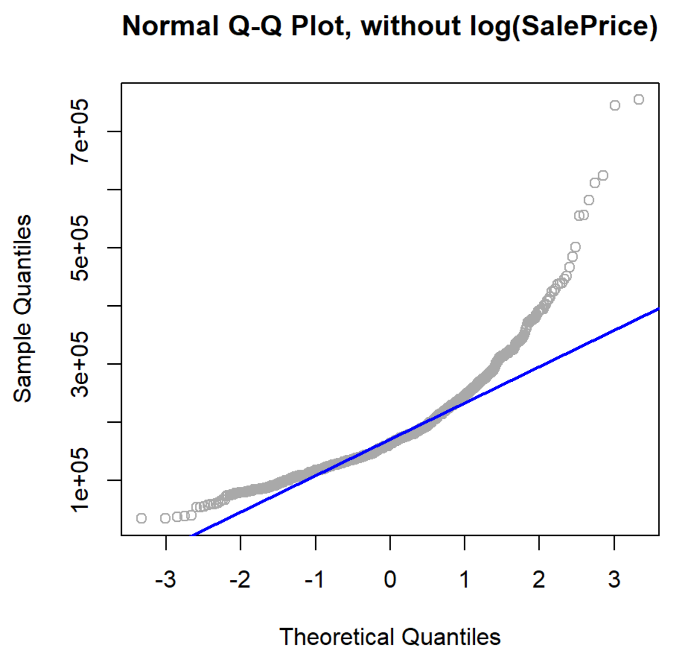
Distribtion of SalePrice

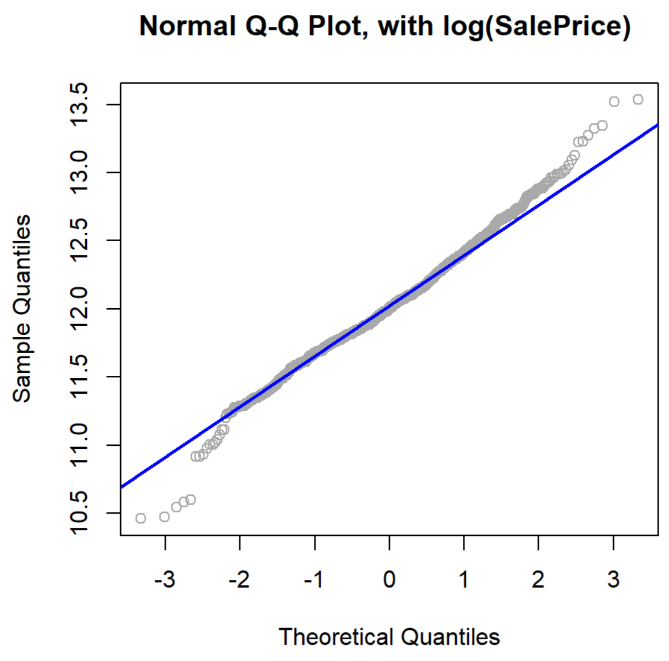
```
#Distribution of SalePrice

par(mfrow = c(1,2))

hist(house_trn$SalePrice,
    xlab   = "Sale Price without log(SalePrice)",
    main   = "Histogram of sales price",
    border = "blue",
    breaks = 20)

#Distribution of log(SalePrice)

hist(log(house_trn$SalePrice),
    xlab   = "Sale Price with log(SalePrice)",
    main   = "Histogram of sales price",
    border = "blue",
    breaks = 20)
```





```
#Q-Q Plot for SalePrice

qqnorm(house_trn$SalePrice, main = "Normal Q-Q Plot, without log(SalePrice)", col = "darkgrey")
qqline(house_trn$SalePrice, col = "blue", lwd = 2)

#Q-Q Plot for log(SalePrice)

qqnorm(log(house_trn$SalePrice), main = "Normal Q-Q Plot, with log(SalePrice) ", col = "darkgrey")
qqline(log(house_trn$SalePrice), col = "blue", lwd = 2)
```





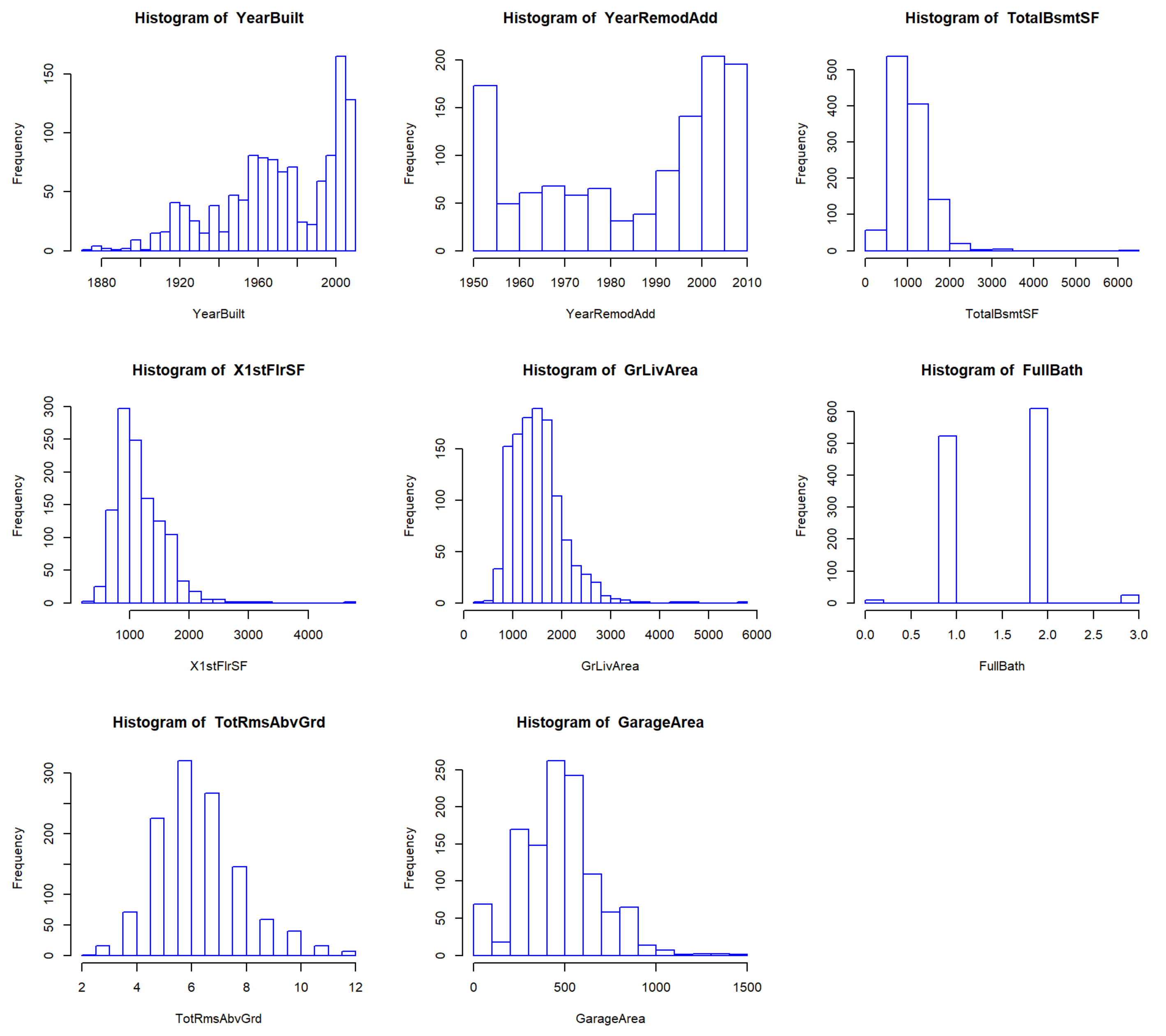
From the above plot, we could see that the SalePrice data is `right-skewed` so we can apply log transformation for SalePrice.

After applying `log transformation` , we could see that the SalePrice is `normally distributed` .

Distribution of numeric varibles

In this section, we will explore the distribution of numeric variables which have high correlation with SalePrice.

```r
#Histogram for checking the distribution of numeric variables

var_sel = c("YearBuilt", "YearRemodAdd","TotalBsmtSF", "X1stFlrSF","GrLivArea", "FullBath", "TotRmsAbvGrd", "Garag
eArea")

par(mfrow = c(1,3))
for(i in 1:length(var_sel))
hist(house_trn[,var_sel[i]],
    xlab   = paste(toString(var_sel[i])),
    main   = paste("Histogram of ",toString(var_sel[i])),
    border = "blue",
    breaks = 20)
```



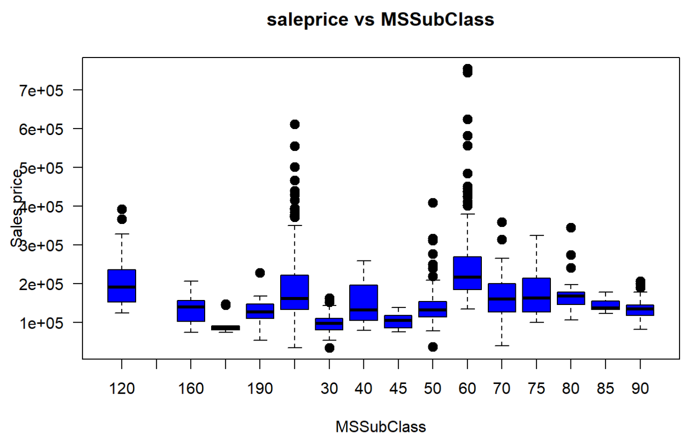
From the above plots, we can infer the following - TotalBsmtSF is `right-skewed` - GrLivArea is `right-skewed` - X1stFlrSF is `right-skewed` - GarageArea is `right-skewed` - YearBuilt is `left-skewed`

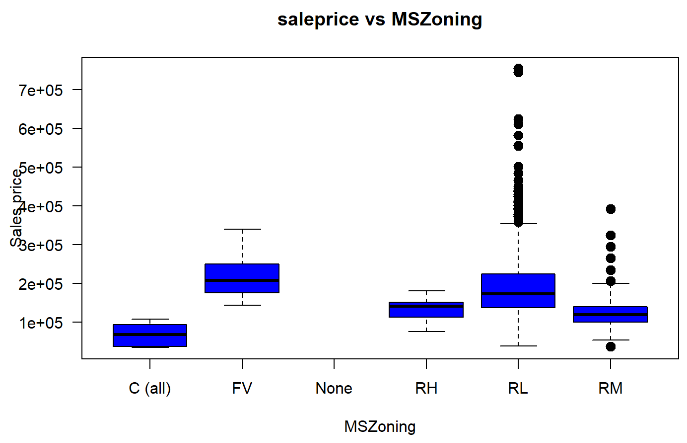
When we use the these variables which has right-skewedness, we can apply log transformation for these variables.

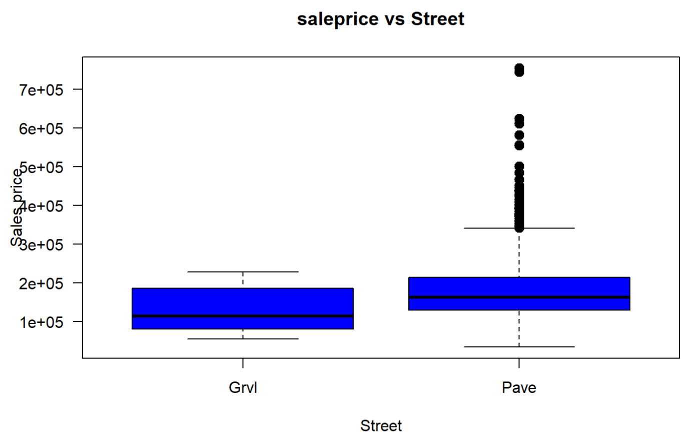
SalePrice Vs Factor Variables

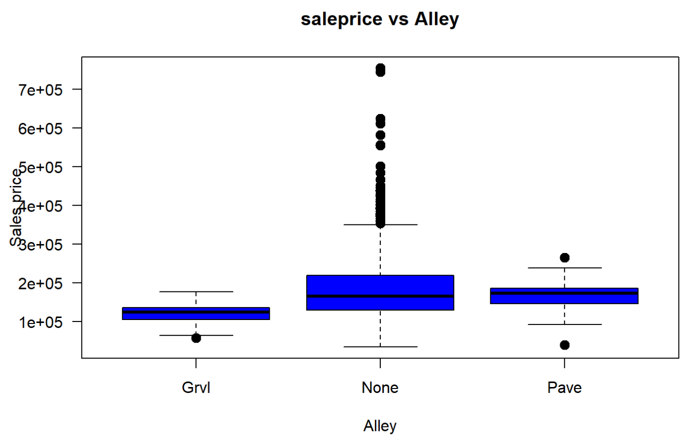
In this section, we can explore the effect of each factor variable on SalePrice.

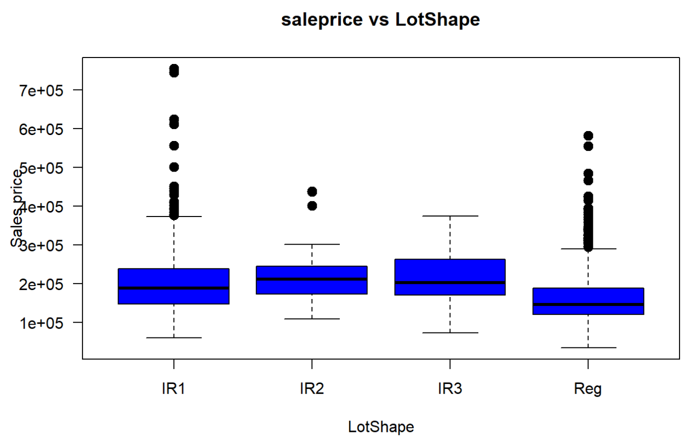
```r
#Box plot for factor variables against SalePrice

char_var = as.array(names(house_trn[, sapply(house_trn, class) == 'factor']))

par(mfrow = c(1,2))
for(i in 1:length(char_var))
boxplot(house_trn$SalePrice ~ house_trn[,char_var[i]], data = mpg,
     xlab   = paste(toString(char_var[i])),
     ylab   = "Sales price",
     main   = paste("saleprice vs", toString(char_var[i])),
     pch    = 20,
     cex    = 2,
     las=1,
     outlier.size=10,
     col    = "blue",
     border = "Black")
```

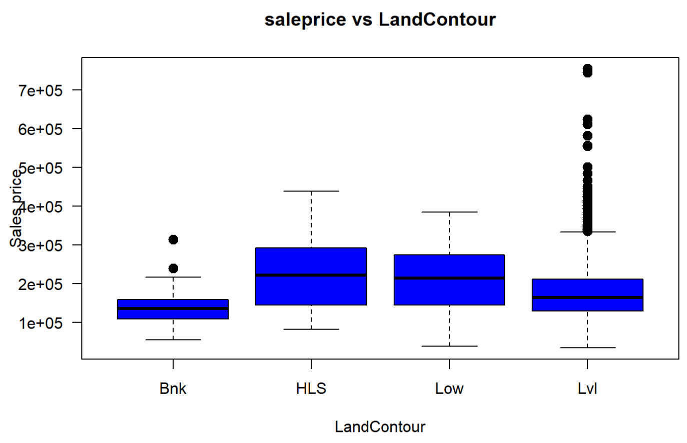

saleprice vs MSSubClass
Sales price
7e+05
6e+05
5e+05
4e+05
3e+05
2e+05
1e+05
120
160
190
30
40
45
50
60
70
75
80
85
90
MSSubClass

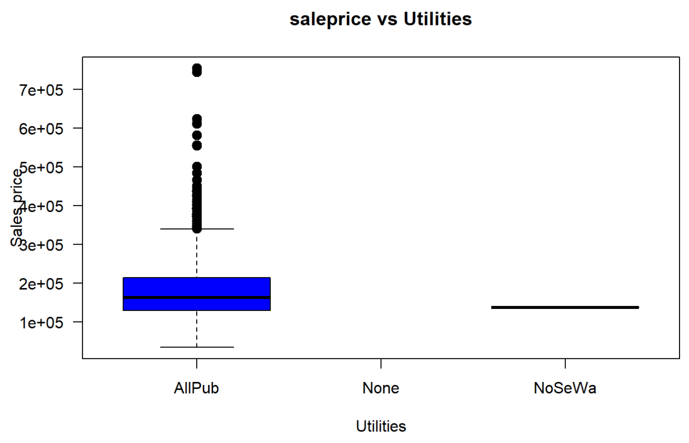

saleprice vs MSZoning
Sales price
7e+05
6e+05
5e+05
4e+05
3e+05
2e+05
1e+05
C (all)
FV
None
RH
RL
RM
MSZoning

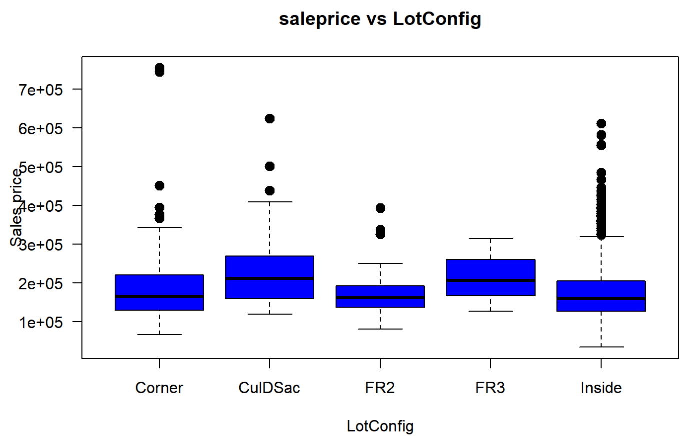

saleprice vs Street
Sales price
7e+05
6e+05
5e+05
4e+05
3e+05
2e+05
1e+05
Grvl
Pave
Street

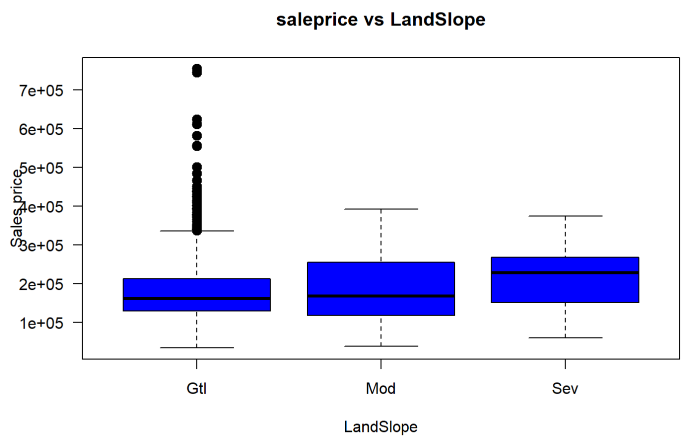

saleprice vs Alley
Sales price
7e+05
6e+05
5e+05
4e+05
3e+05
2e+05
1e+05
Grvl
None
Pave
Alley

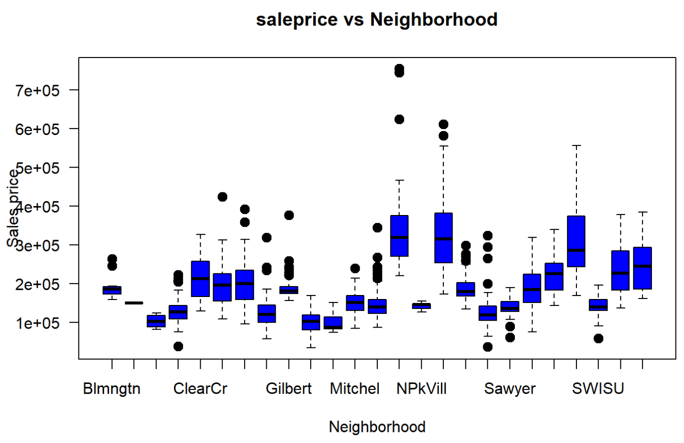

saleprice vs LotShape
Sales price
7e+05
6e+05
5e+05
4e+05
3e+05
2e+05
1e+05
IR1
IR2
IR3
Reg
LotShape

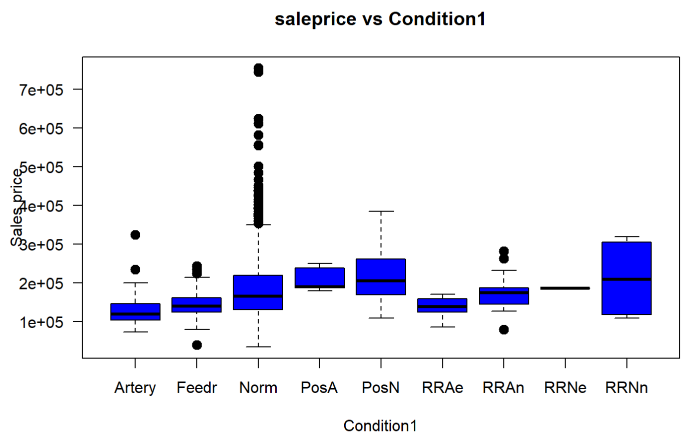

saleprice vs LandContour
Sales price
7e+05
6e+05
5e+05
4e+05
3e+05
2e+05
1e+05
Bnk
HLS
Low
Lvl
LandContour

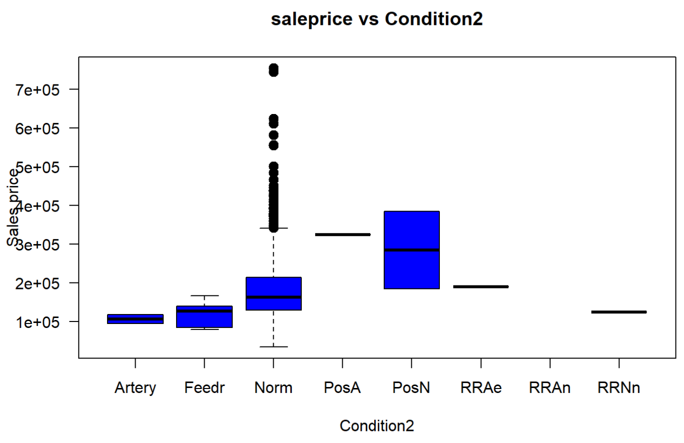

saleprice vs Utilities
Sales price
7e+05
6e+05
5e+05
4e+05
3e+05
2e+05
1e+05
AllPub
None
NoSeWa
Utilities

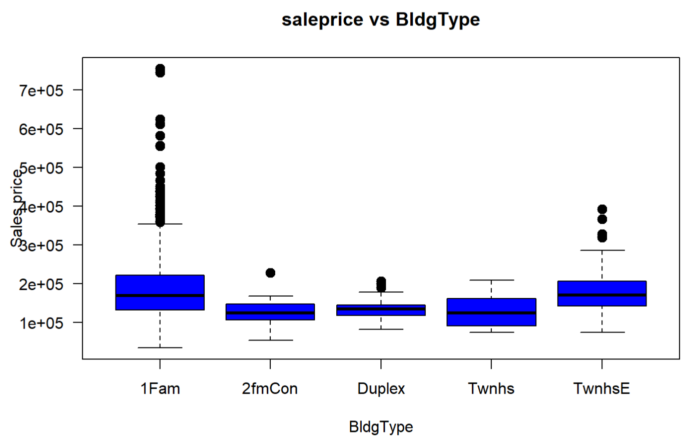

saleprice vs LotConfig
Sales price
7e+05
6e+05
5e+05
4e+05
3e+05
2e+05
1e+05
Corner
CulDSac
FR2
FR3
Inside
LotConfig

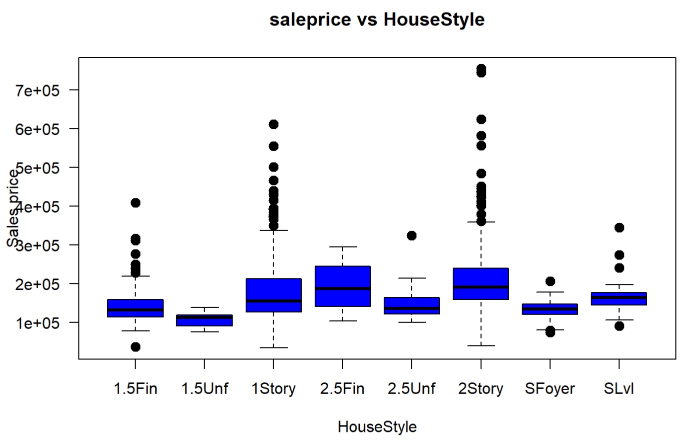

saleprice vs LandSlope
Sales_price
7e+05
6e+05
5e+05
4e+05
3e+05
2e+05
1e+05
Gtl
Mod
Sev
LandSlope

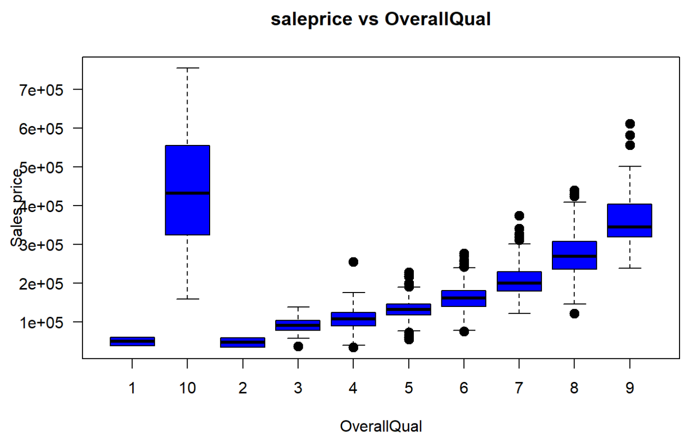

saleprice vs Neighborhood
Sales_price
7e+05
6e+05
5e+05
4e+05
3e+05
2e+05
1e+05
Blmngtn
ClearCr
Gilbert
Mitchel
NPkVill
Sawyer
SWISU
Neighborhood

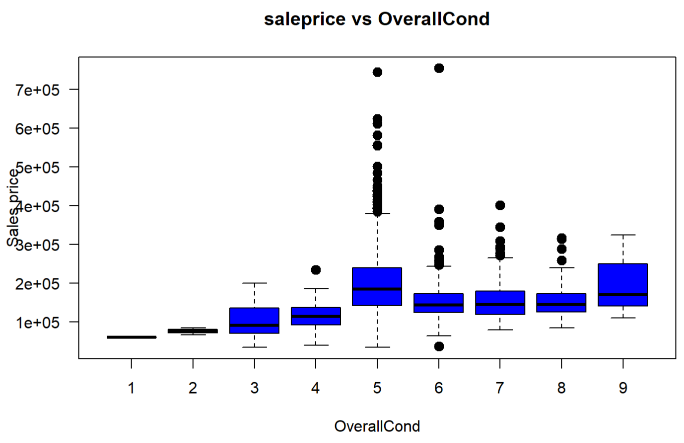

saleprice vs Condition1
Sales_price
7e+05
6e+05
5e+05
4e+05
3e+05
2e+05
1e+05
Artery
Feedr
Norm
PosA
PosN
RRAe
RRAn
RRNe
RRNn
Condition1

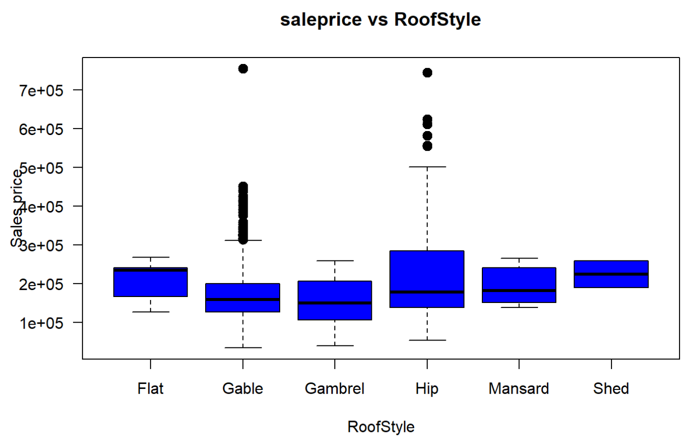

saleprice vs Condition2
Sales_price
7e+05
6e+05
5e+05
4e+05
3e+05
2e+05
1e+05
Artery
Feedr
Norm
PosA
PosN
RRAe
RRAn
RRNn
Condition2

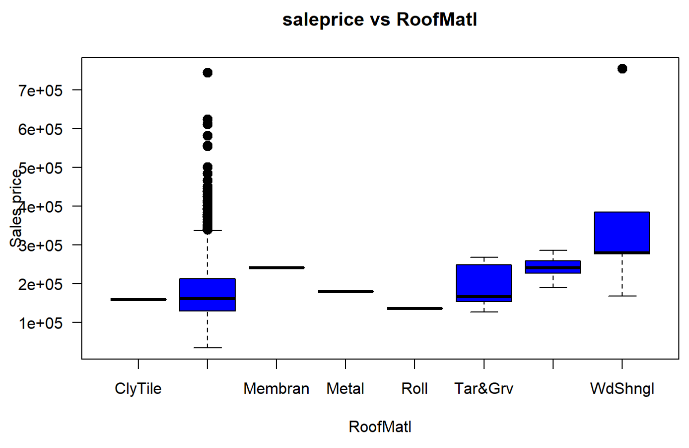

saleprice vs BldgType
Sales_price
7e+05
6e+05
5e+05
4e+05
3e+05
2e+05
1e+05
1Fam
2fmCon
Duplex
Twnhs
TwnhsE
BldgType

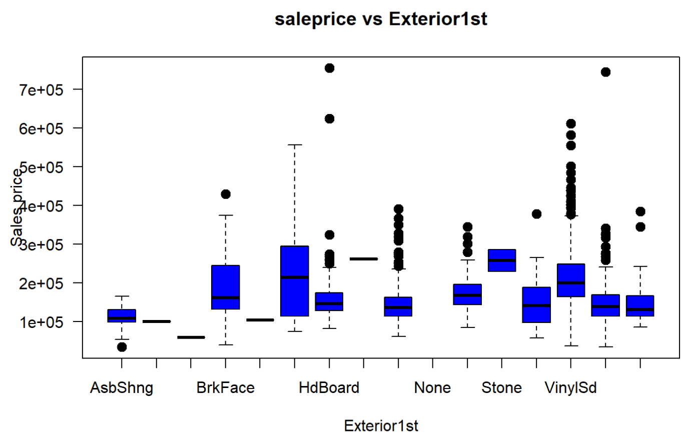

saleprice vs HouseStyle
Sales_price
7e+05
6e+05
5e+05
4e+05
3e+05
2e+05
1e+05
1.5Fin
1.5Unf
1Story
2.5Fin
2.5Unf
2Story
SFoyer
SLvl
HouseStyle

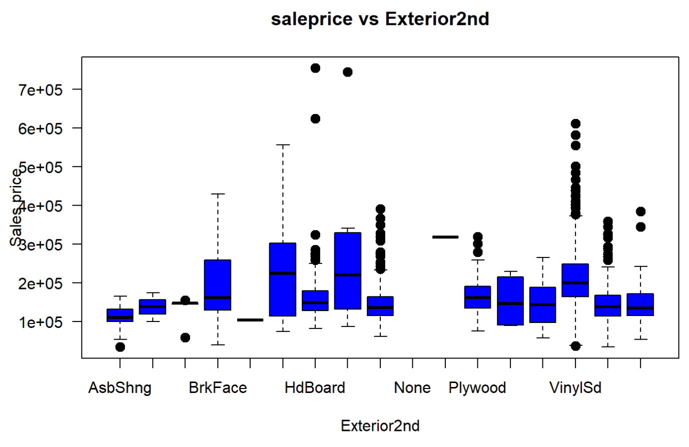

saleprice vs OverallQual
Sales_price
7e+05
6e+05
5e+05
4e+05
3e+05
2e+05
1e+05
1
10
2
3
4
5
6
7
8
9
OverallQual

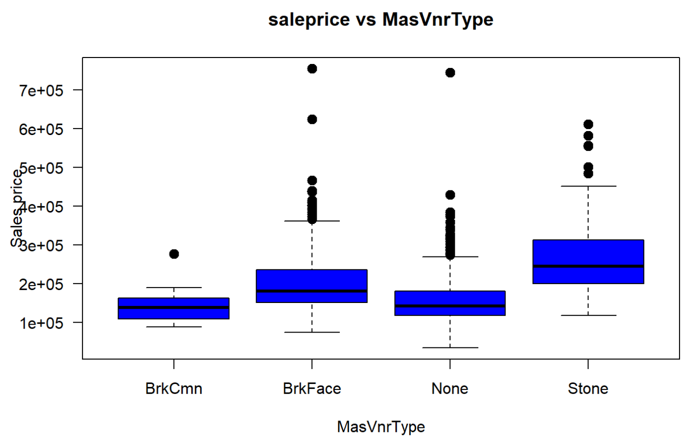

saleprice vs OverallCond
Sales_price
7e+05
6e+05
5e+05
4e+05
3e+05
2e+05
1e+05
1
2
3
4
5
6
7
8
9
OverallCond

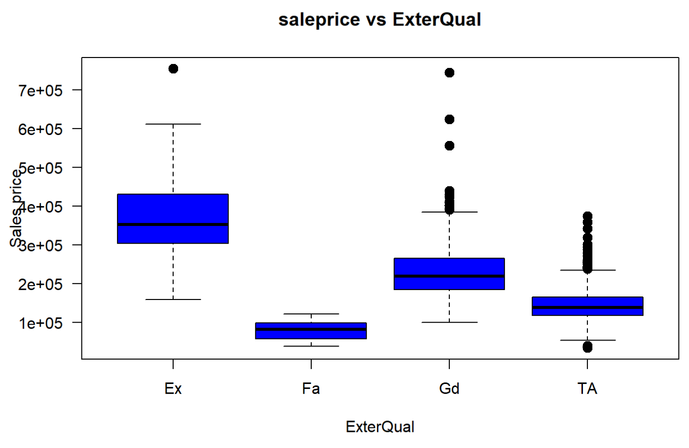

saleprice vs RoofStyle
Sales price
7e+05
6e+05
5e+05
4e+05
3e+05
2e+05
1e+05
Flat
Gable
Gambrel
Hip
Mansard
Shed
RoofStyle

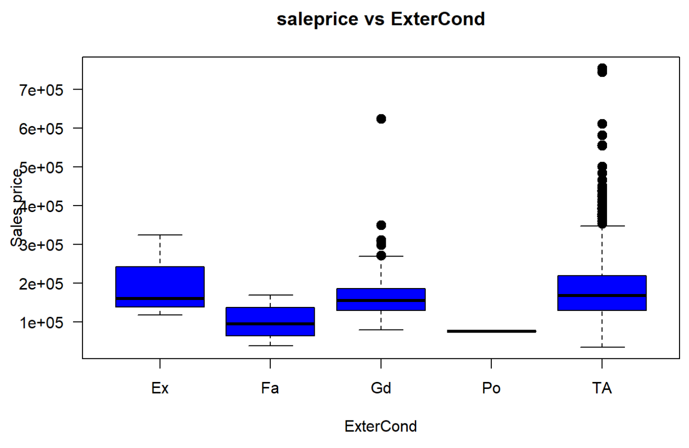

saleprice vs RoofMatl
Sales price
7e+05
6e+05
5e+05
4e+05
3e+05
2e+05
1e+05
ClyTile
Membran
Metal
Roll
Tar&Grv
WdShngl
RoofMatl

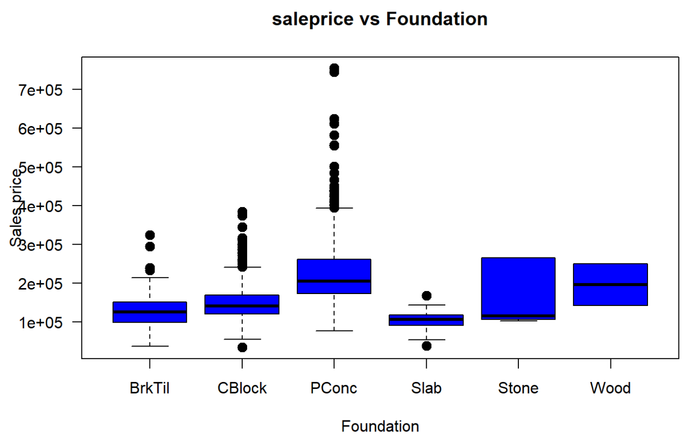

saleprice vs Exterior1st
Sales price
7e+05
6e+05
5e+05
4e+05
3e+05
2e+05
1e+05
AsbShng
BrkFace
HdBoard
None
Stone
VinylSd
Exterior1st


saleprice vs Exterior2nd
Sales price
7e+05
6e+05
5e+05
4e+05
3e+05
2e+05
1e+05
AsbShng
BrkFace
HdBoard
None
Plywood
VinylSd
Exterior2nd

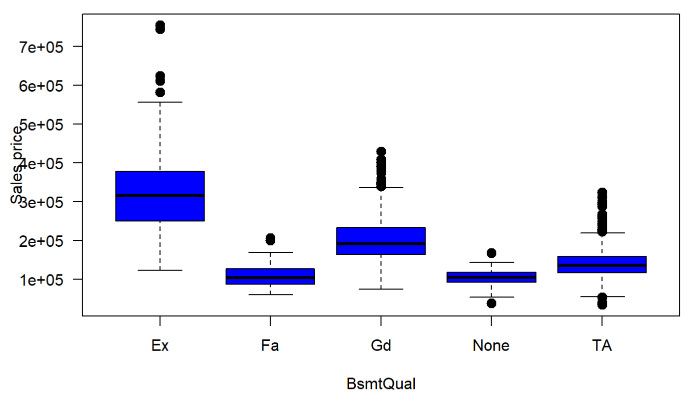

saleprice vs MasVnrType
Sales price
7e+05
6e+05
5e+05
4e+05
3e+05
2e+05
1e+05
BrkCmn
BrkFace
None
Stone
MasVnrType


saleprice vs ExterQual
Sales price
7e+05
6e+05
5e+05
4e+05
3e+05
2e+05
1e+05
Ex
Fa
Gd
TA
ExterQual

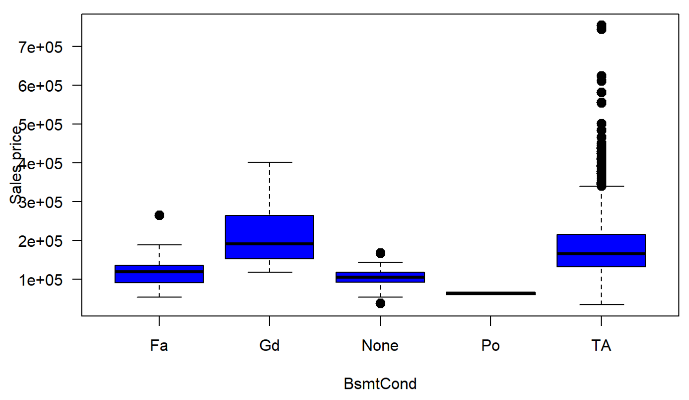

saleprice vs ExterCond
Sales price
7e+05
6e+05
5e+05
4e+05
3e+05
2e+05
1e+05
Ex
Fa
Gd
Po
TA
ExterCond


saleprice vs Foundation
Sales price
7e+05
6e+05
5e+05
4e+05
3e+05
2e+05
1e+05
BrkTil
CBlock
PConc
Slab
Stone
Wood
Foundation

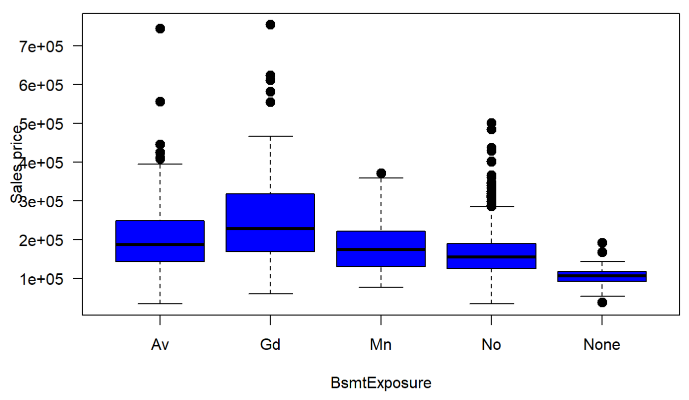
saleprice vs BsmtQual



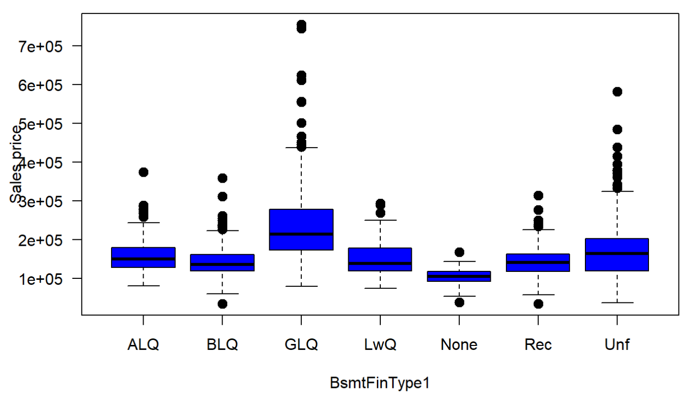
saleprice vs BsmtCond



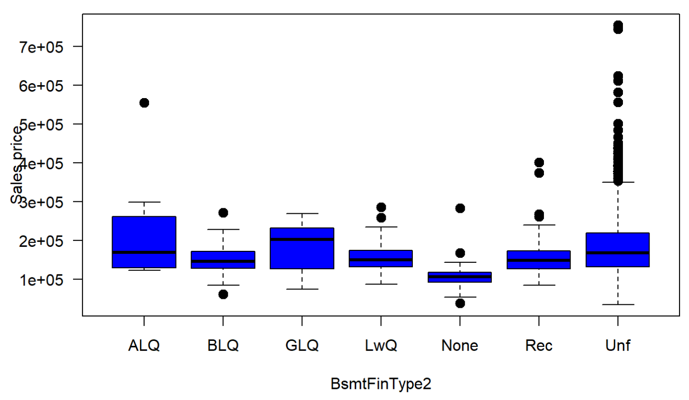
saleprice vs BsmtExposure



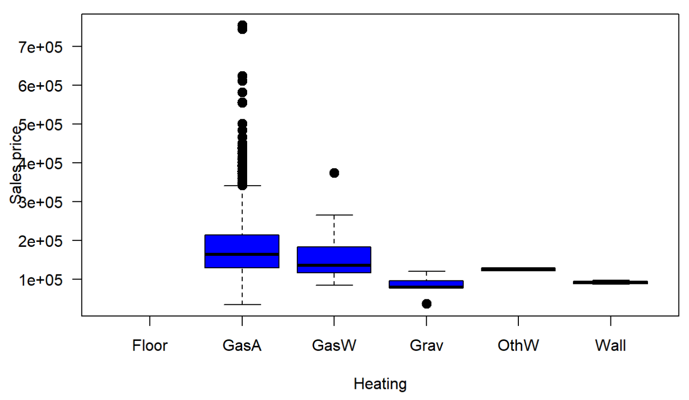
saleprice vs BsmtFinType1



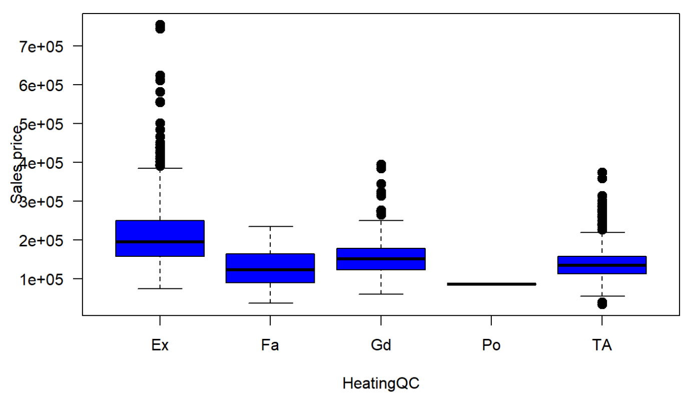
saleprice vs BsmtFinType2



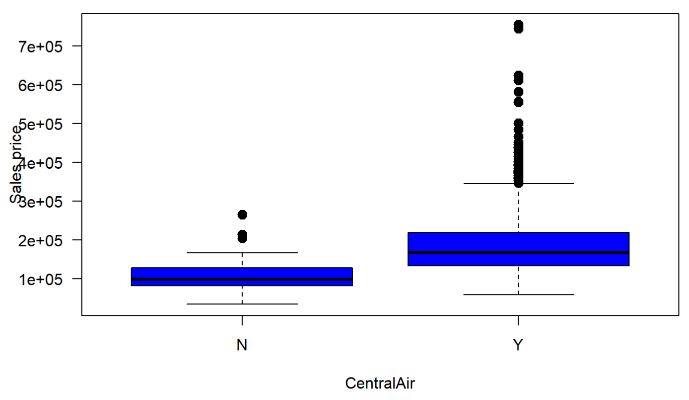
saleprice vs Heating

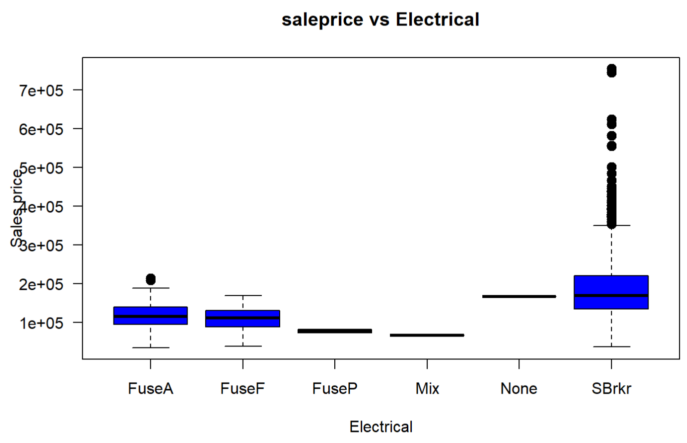


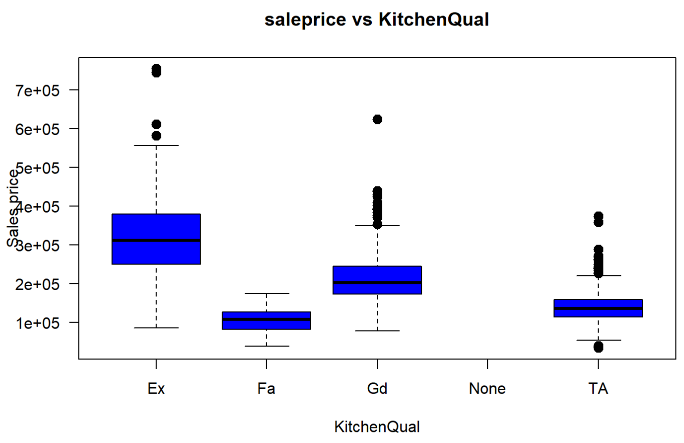
saleprice vs HeatingQC

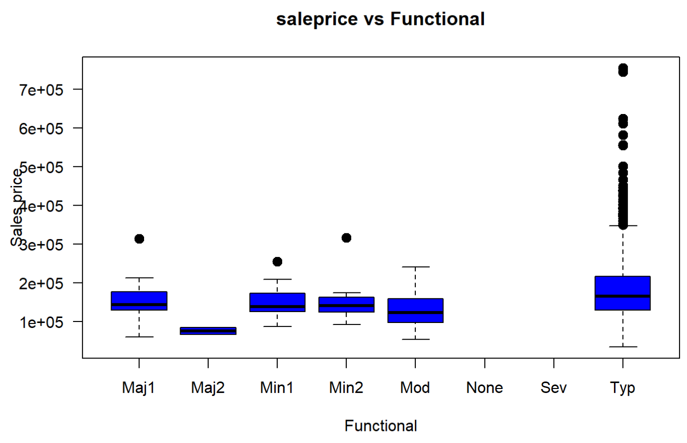


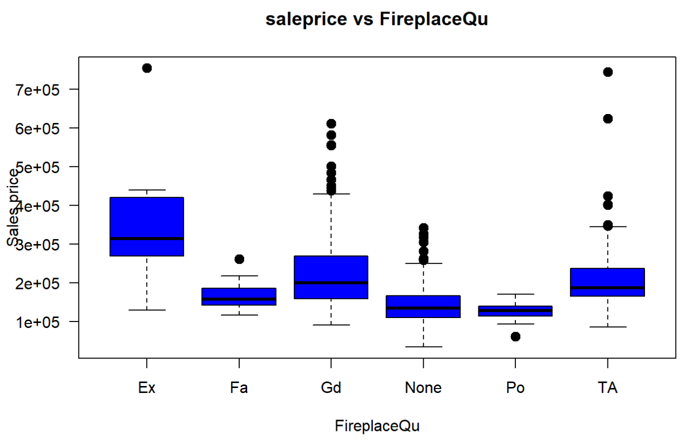
saleprice vs CentralAir

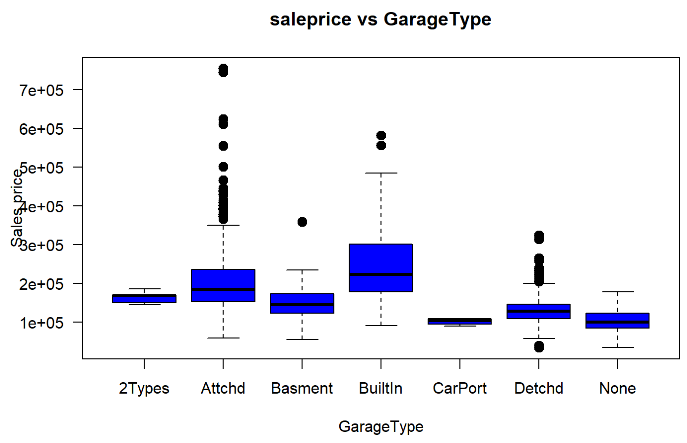


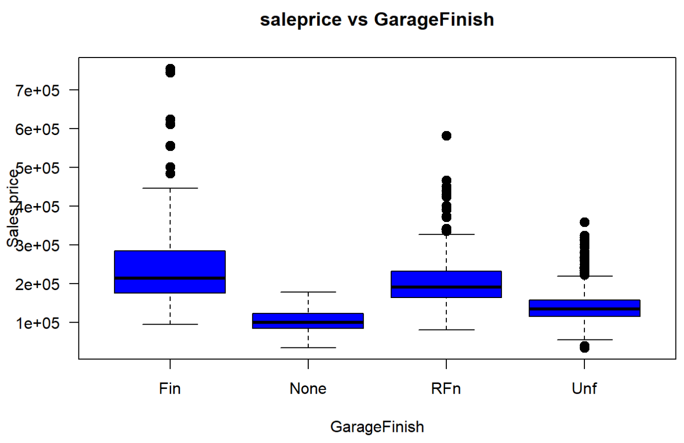

saleprice vs Electrical
Sales price
1e+05
2e+05
3e+05
4e+05
5e+05
6e+05
7e+05
FuseA
FuseF
FuseP
Mix
None
SBrkr
Electrical

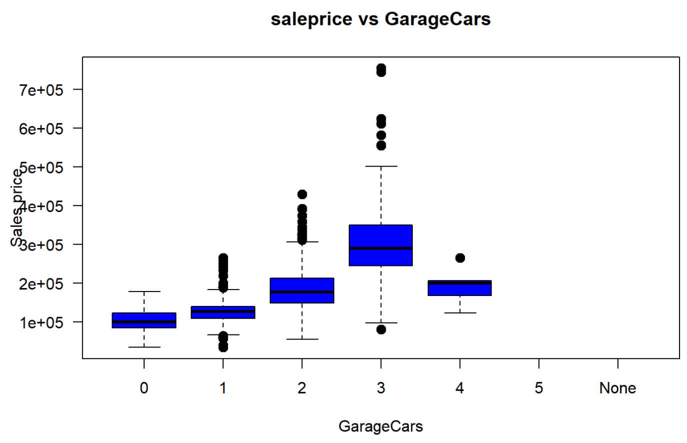

saleprice vs KitchenQual
Sales price
1e+05
2e+05
3e+05
4e+05
5e+05
6e+05
7e+05
Ex
Fa
Gd
None
TA
KitchenQual

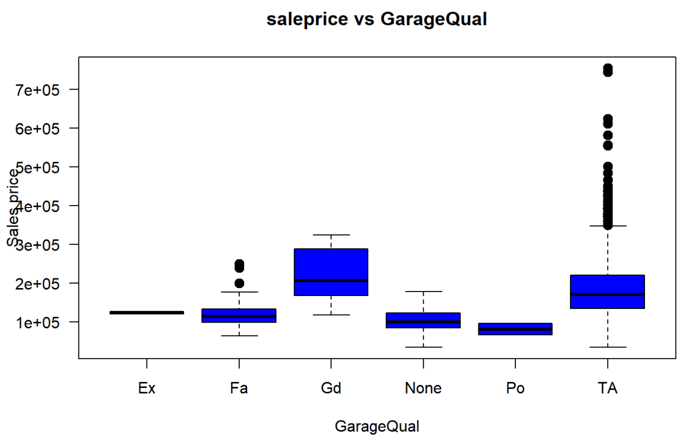

saleprice vs Functional
Sales price
1e+05
2e+05
3e+05
4e+05
5e+05
6e+05
7e+05
Maj1
Maj2
Min1
Min2
Mod
None
Sev
Typ
Functional

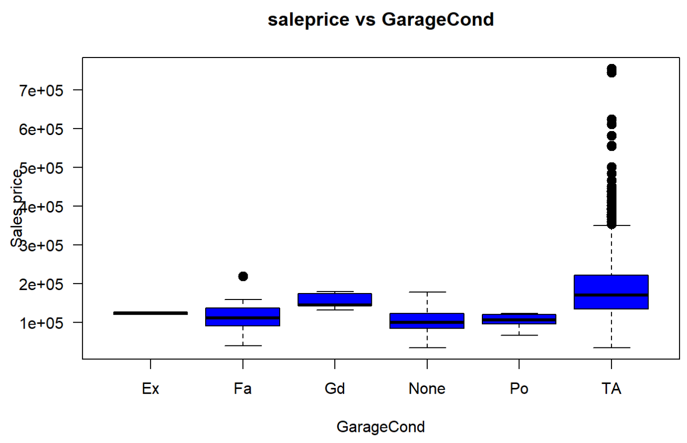

saleprice vs FireplaceQu
Sales price
1e+05
2e+05
3e+05
4e+05
5e+05
6e+05
7e+05
Ex
Fa
Gd
None
Po
TA
FireplaceQu

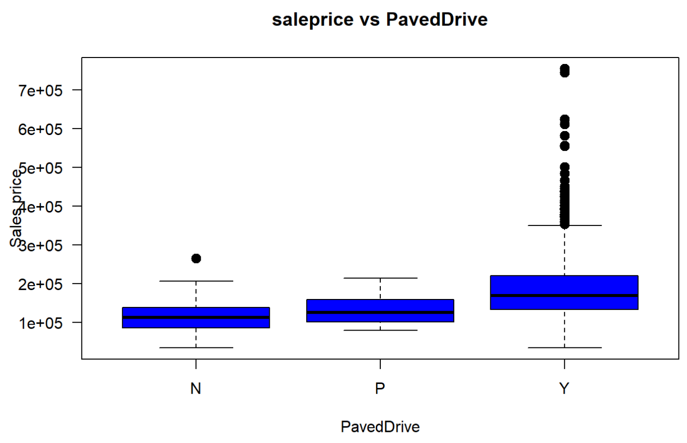

saleprice vs GarageType
Sales price
1e+05
2e+05
3e+05
4e+05
5e+05
6e+05
7e+05
2Types
Attchd
Basment
BuiltIn
CarPort
Detchd
None
GarageType

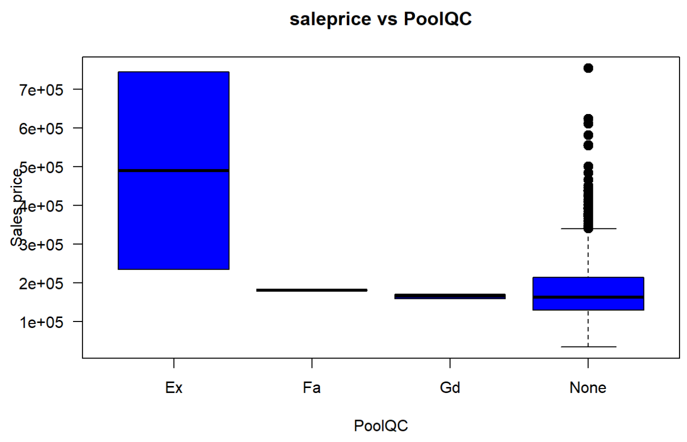

saleprice vs GarageFinish
Sales price
1e+05
2e+05
3e+05
4e+05
5e+05
6e+05
7e+05
Fin
None
RFn
Unf
GarageFinish

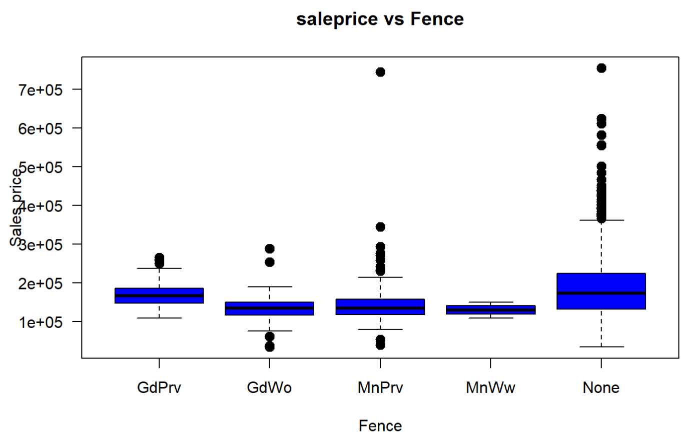

saleprice vs GarageCars
Sales price
1e+05
2e+05
3e+05
4e+05
5e+05
6e+05
7e+05
0
1
2
3
4
5
None
GarageCars

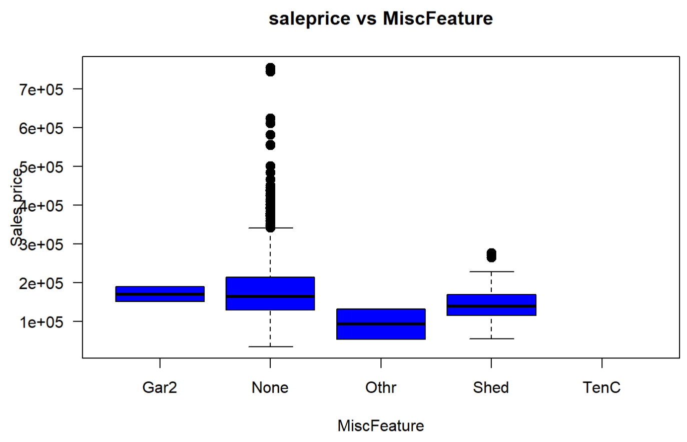

saleprice vs GarageQual
Sales price
1e+05
2e+05
3e+05
4e+05
5e+05
6e+05
7e+05
Ex
Fa
Gd
None
Po
TA
GarageQual

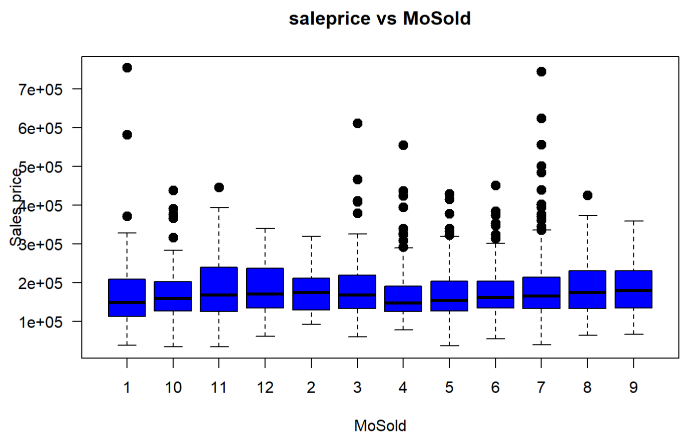

saleprice vs GarageCond
Sales price
7e+05
6e+05
5e+05
4e+05
3e+05
2e+05
1e+05
Ex
Fa
Gd
None
Po
TA
GarageCond

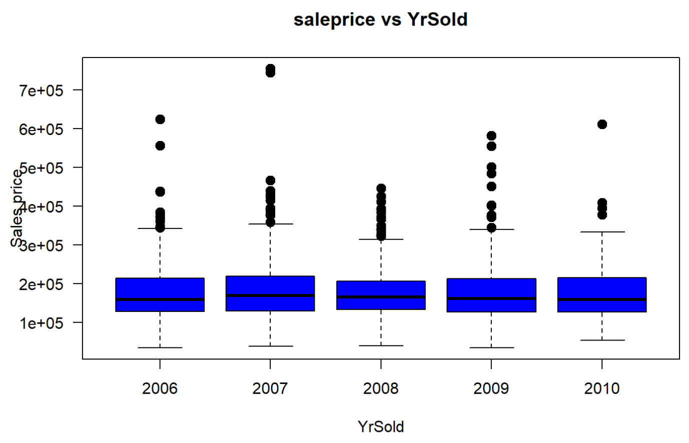

saleprice vs PavedDrive
Sales price
7e+05
6e+05
5e+05
4e+05
3e+05
2e+05
1e+05
N
P
Y
PavedDrive

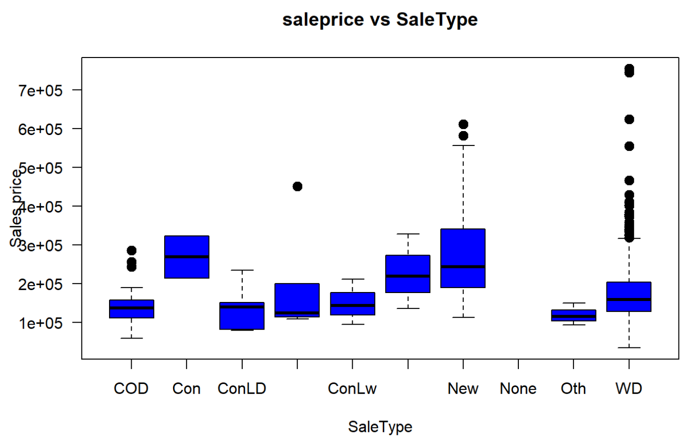

saleprice vs PoolQC
Sales price
7e+05
6e+05
5e+05
4e+05
3e+05
2e+05
1e+05
Ex
Fa
Gd
None
PoolQC

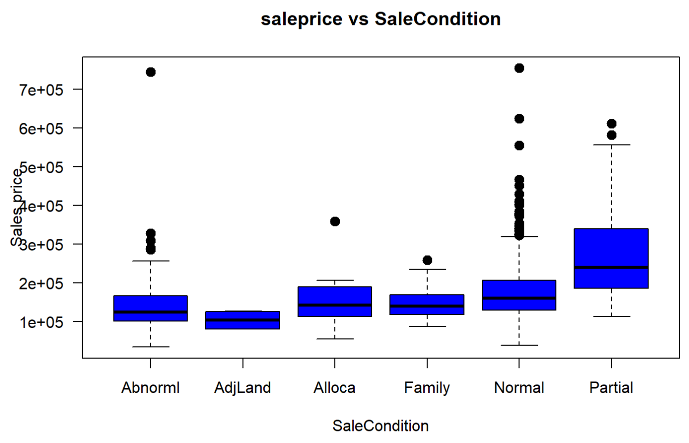

saleprice vs Fence
Sales price
7e+05
6e+05
5e+05
4e+05
3e+05
2e+05
1e+05
GdPrv
GdWo
MnPrv
MnWw
None
Fence


saleprice vs MiscFeature
Sales price
7e+05
6e+05
5e+05
4e+05
3e+05
2e+05
1e+05
Gar2
None
Othr
Shed
TenC
MiscFeature


saleprice vs MoSold
Sales price
7e+05
6e+05
5e+05
4e+05
3e+05
2e+05
1e+05
1
10
11
12
2
3
4
5
6
7
8
9
MoSold







From the above plots we could see that the below variable have significance on SalePrice.

- MSSubClass
- MSZoning
- Neighborhood
- OverallQual
- KitchenQual

Model Building

Based on the above data exploration, we will start building the model with the variables which has high significane on Saleprice.

We will also put a log transformation for the SalePrice as it has right skewedness, which we infered from the histogram plot in the data exploration section.

Additive Model

```
add_mod = lm(log(SalePrice) ~  (MSSubClass +GrLivArea + TotalBsmtSF + OverallQual+ YearBuilt+ FullBath + YearRemod
Add + Neighborhood +TotRmsAbvGrd+ExterQual), data = house_trn)
```

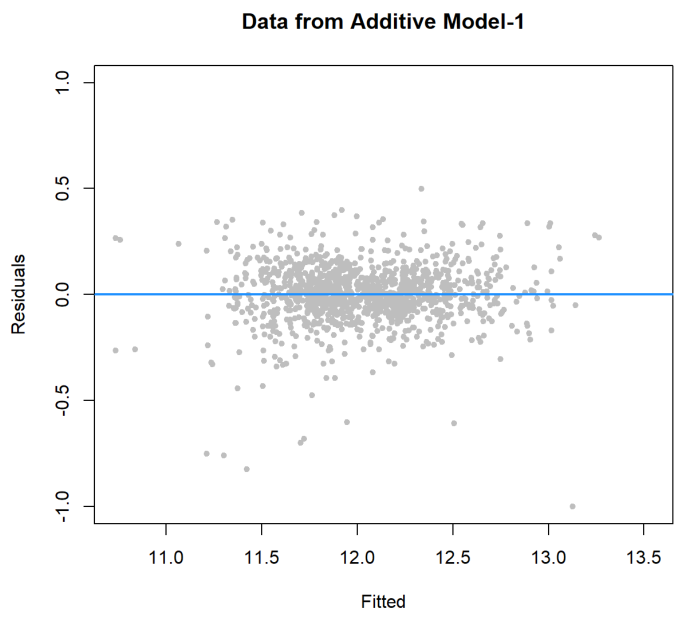
R-Squared

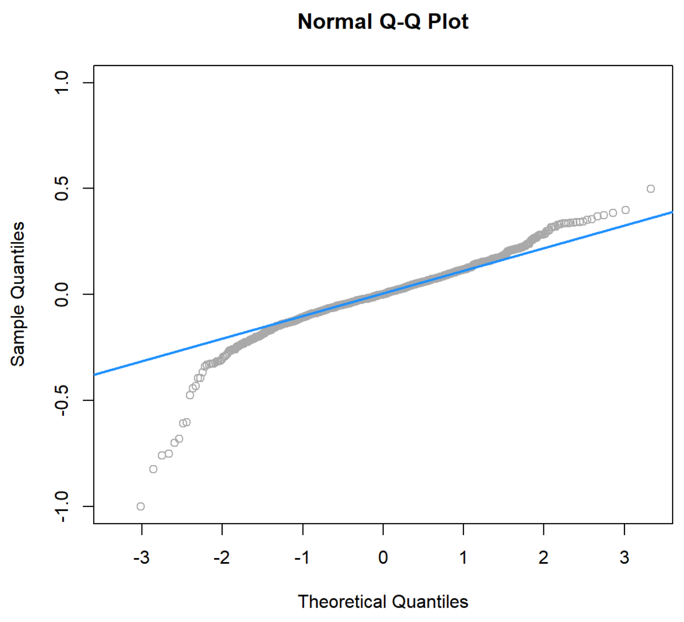
```
(r2_add=summary(add_mod)$r.squared)
```

```
## [1] 0.8659282
```

Q-Q and Fitted vs Residual Plot

```
par(mfrow = c(1, 2))
plot(fitted(add_mod), resid(add_mod), col = "grey", pch = 20,ylim = c(-1, 1),
    xlab = "Fitted", ylab = "Residuals", main = "Data from Additive Model-1")
abline(h = 0, col = "dodgerblue", lwd = 2)

qqnorm(resid(add_mod), main = "Normal Q-Q Plot", col = "darkgrey",ylim = c(-1, 1))
qqline(resid(add_mod), col = "dodgerblue", lwd = 2)
```





BP and Shapiro Test

```
bptest(add_mod)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  add_mod
## BP = 386.15, df = 56, p-value < 2.2e-16
```

```
shapiro.test(resid(add_mod))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(add_mod)
## W = 0.88892, p-value < 2.2e-16
```

Test and Train RMSE

```
 (RMSE_Train_add = sqrt(mean(resid(add_mod) ^ 2)))
```

```
## [1] 0.1466346
```

```
 (RMSE_test_add = sqrt(mean((log(house_tst$SalePrice) - predict(add_mod,house_tst)) ^ 2)))
```

```
## [1] 0.1674523
```

Discussion on test results for Additive model

The above additive model have the following test results

- R^2 value of 0.8659, ie 86.59% of the variance in SalePrice is explained by linear relationship with variables in the `additive model` .
- The model is `violating the constant variance and normality assumptions` which we could see from the QQ plot and fitted vs residual plot.The p values of `BP test` and `Shapiro-wilk test` also confirms the same.
- Train RMSE of 0.1466
- Test RMSE of 0.1675

This model is violating the model assumptions and have R^2 value of 0.8659, we will do further analysis to find a model which doesn't violate the model assumptions and have better R^2.

Additive Model 2

In the above additive model, few variables like MSSubclass which doesn't have significant p-value, so we will remove those non-significant variables and fit another additive model.

```
add_mod2 = lm(log(SalePrice) ~  (GrLivArea + TotalBsmtSF + OverallQual+ YearBuilt+ YearRemodAdd + Neighborhood +To
tRmsAbvGrd+ExterQual), data = house_trn)
```

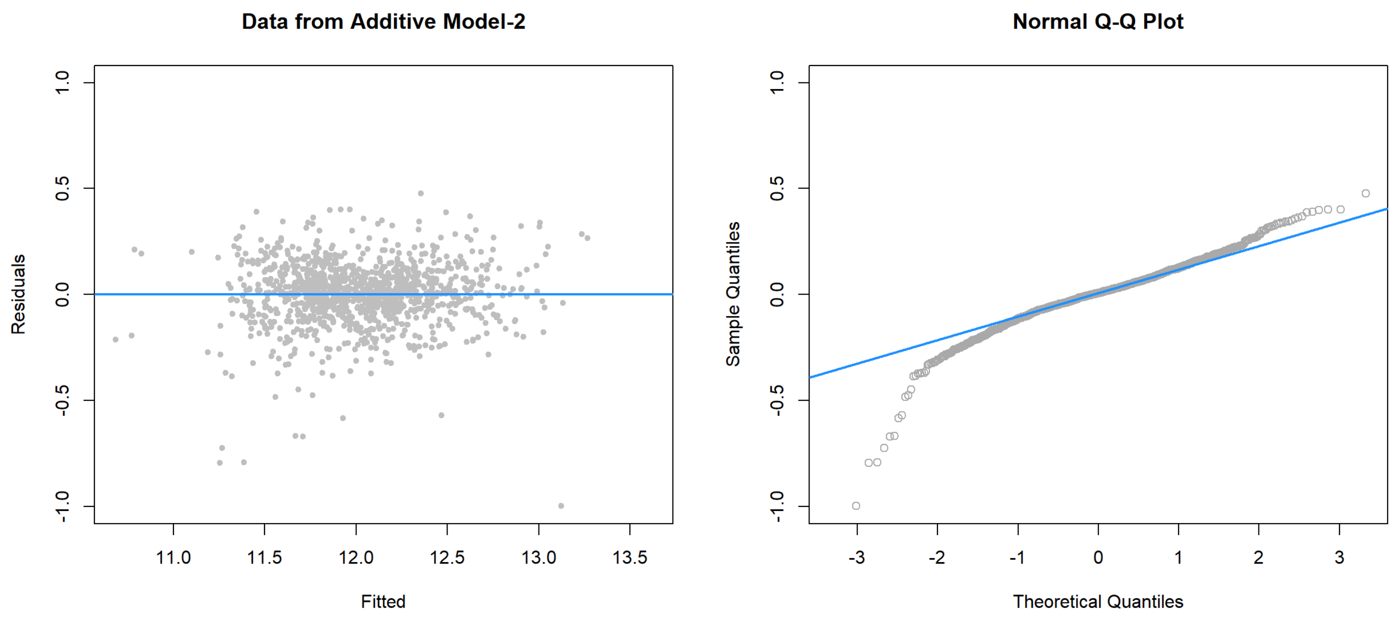
R-Squared

```
(r2_add2=summary(add_mod2)$r.squared)
```

```
## [1] 0.8560228
```

Q-Q and Fitted vs Residual Plot

```
par(mfrow = c(1, 2))
plot(fitted(add_mod2), resid(add_mod2), col = "grey", pch = 20,ylim = c(-1, 1),
    xlab = "Fitted", ylab = "Residuals", main = "Data from Additive Model-2")
abline(h = 0, col = "dodgerblue", lwd = 2)

qqnorm(resid(add_mod2), main = "Normal Q-Q Plot", col = "darkgrey",ylim = c(-1, 1))
qqline(resid(add_mod2), col = "dodgerblue", lwd = 2)
```



BP and Shapiro Test

```
bptest(add_mod2)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  add_mod2
## BP = 361.98, df = 41, p-value < 2.2e-16
```

```
shapiro.test(resid(add_mod2))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(add_mod2)
## W = 0.89806, p-value < 2.2e-16
```

Test and Train RMSE

```
 (RMSE_Train_add2 = sqrt(mean(resid(add_mod2) ^ 2)))
```

```
## [1] 0.1519549
```

```
 (RMSE_test_add2 = sqrt(mean((log(house_tst$SalePrice) - predict(add_mod2,house_tst)) ^ 2)))
```

```
## [1] 0.1739923
```

Discussion on test results for Additive model-2

The above additive model have the following test results

- R^2 value of 0.856, ie 85.60% of the variance in SalePrice is explained by linear relationship with variables in the `additive model` .
- The model is `violating the constant variance and normality assumptions` which we could see from the QQ plot and fitted vs residual plot.The p values of `BP test` and `Shapiro-wilk test` also confirms the same.
- Train RMSE of 0.152
- Test RMSE of 0.174

This model is also violating the model assumptions and have R^2 value of 0.856, which is lower than the previous model.We will do further analysis to find a model which doesn't violate the model assumptions and have better R^2.

Anova of Additive Model 1 and Model 2

```
anova(add_mod2,add_mod)
```

```
## Analysis of Variance Table
## 
## Model 1: log(SalePrice) ~ (GrLivArea + TotalBsmtSF + OverallQual + YearBuilt + 
##     YearRemodAdd + Neighborhood + TotRmsAbvGrd + ExterQual)
## Model 2: log(SalePrice) ~ (MSSubClass + GrLivArea + TotalBsmtSF + OverallQual + 
##     YearBuilt + FullBath + YearRemodAdd + Neighborhood + TotRmsAbvGrd + 
##     ExterQual)
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)    
## 1   1126 26.970                                  
## 2   1111 25.114 15    1.8555 5.4722 8.087e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The annova test has low p-vlaue of 8.086512910^{-11}, which is confirming that the variable which we removed in additive model 2 is not significant so we can continue to find a better model with the variables in additive model-2.

Log Interaction Model

The constant variance and normality assumptions are violated in the above 2 additive models. The reason for this could be that the variables have righ-skewedness, as we commonly say in statistics that the **garbage in, garbage out**.

So we will try to fix the right-skewedness by applying log transformation for the predictor variables.

We will also try some interaction and see, if the R^2 is improving.

```
log_int_mod = lm(log(SalePrice) ~  (log(GrLivArea)+OverallQual+ YearBuilt+ YearRemodAdd + Neighborhood +log(TotRms
AbvGrd)+ExterQual)^2, data = house_trn)
```

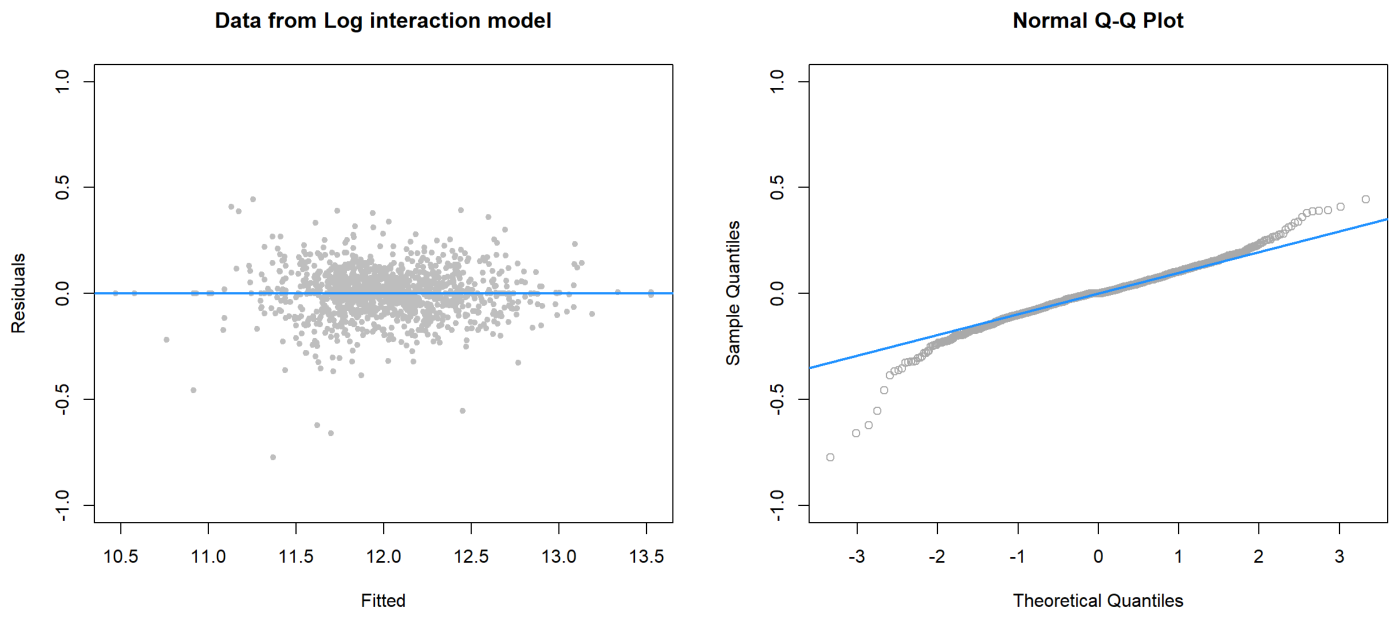
R-Squared

```
(r2_int=summary(log_int_mod)$r.squared)
```

```
## [1] 0.9138646
```

Q-Q and Fitted vs Residual Plot

```
par(mfrow = c(1, 2))
plot(fitted(log_int_mod), resid(log_int_mod), col = "grey", pch = 20,ylim = c(-1, 1),
    xlab = "Fitted", ylab = "Residuals", main = "Data from Log interaction model")
abline(h = 0, col = "dodgerblue", lwd = 2)

qqnorm(resid(log_int_mod), main = "Normal Q-Q Plot", col = "darkgrey",ylim = c(-1, 1))
qqline(resid(log_int_mod), col = "dodgerblue", lwd = 2)
```



BP and Shapiro Test

```
bptest(log_int_mod)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  log_int_mod
## BP = 266.21, df = 287, p-value = 0.8055
```

```
shapiro.test(resid(log_int_mod))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(log_int_mod)
## W = 0.96094, p-value < 2.2e-16
```

Test and Train RMSE

```
 (RMSE_Train_log_int = sqrt(mean(resid(log_int_mod) ^ 2)))
```

```
## [1] 0.1175327
```

```
 (RMSE_test_log_int = sqrt(mean((log(house_tst$SalePrice) - predict(log_int_mod,house_tst)) ^ 2)))
```

```
## [1] 0.3491662
```

Discussion on test results for Log interaction Model

The above log interaction model have the following test results

- R^2 value of 0.9139, ie 91.39% of the variance in SalePrice is explained by linear relationship with variables in the `additive model` .

- The model is `violating the normality assumptions` but the `constant variance assumption is valid` which we could see from the QQ plot and fitted vs residual plot.The high p-value of `BP test` confirms the constant variance and low p-value of `Shapiro-wilk test` shows that the normality assumption is invalid.
- Train RMSE of 0.1175
- Test RMSE of 0.3492

Even though the constant variance is valid for the log interaction model, its still violating the normality assumption and we see better R^2 value of 0.9139 compared to the other 2 additive models.

We will do further analysis and try to fit a model which doesn't violate the model assumptions and still have better R^2 as this model.

Log Interation Model without Influential data

To fix the normality assumption violation, we wil remove the influential observations and fit the above model.

```
#Log Intearction Model

log_int_mod = lm(log(SalePrice) ~  (log(GrLivArea)+OverallQual+ YearBuilt+ YearRemodAdd + Neighborhood +log(TotRms
AbvGrd)+ExterQual)^2, data = house_trn)

#Cooks distance for log interaction model

log_int_mod_cd = cooks.distance(log_int_mod)

#Log interaction model after removing influential points

 int_add_cd_mod = lm(log(SalePrice) ~  (log(GrLivArea)+OverallQual+YearBuilt + YearRemodAdd + Neighborhood +log(To
tRmsAbvGrd)+ExterQual)^2, data = house_trn  , subset = log_int_mod_cd < 4 / length(log_int_mod_cd))
```

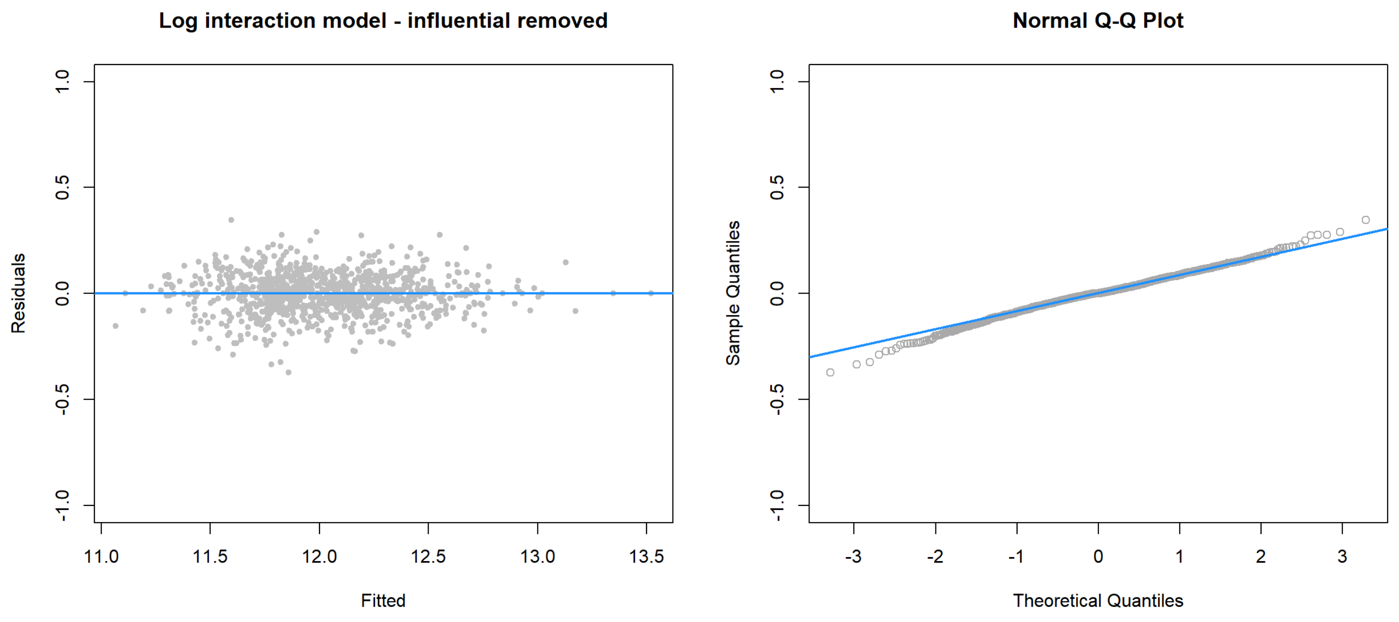
R-Squared

```
(r2_cd=summary(int_add_cd_mod)$r.squared)
```

```
## [1] 0.9294778
```

Q-Q and Fitted vs Residual Plot

```
 par(mfrow = c(1, 2))
plot(fitted(int_add_cd_mod), resid(int_add_cd_mod), col = "grey", pch = 20,ylim = c(-1, 1),
    xlab = "Fitted", ylab = "Residuals", main = "Log interaction model - influential removed")
abline(h = 0, col = "dodgerblue", lwd = 2)

qqnorm(resid(int_add_cd_mod), main = "Normal Q-Q Plot", col = "darkgrey",ylim = c(-1, 1))
qqline(resid(int_add_cd_mod), col = "dodgerblue", lwd = 2)
```



BP and Shapiro Test

```
bptest(int_add_cd_mod)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  int_add_cd_mod
## BP = 209.4, df = 223, p-value = 0.7343
```

```
shapiro.test(resid(int_add_cd_mod))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(int_add_cd_mod)
## W = 0.99296, p-value = 0.0001155
```

Test and Train RMSE

```
(RMSE_Train_log_cd = sqrt(mean(resid(int_add_cd_mod) ^ 2)))
```

```
## [1] 0.09258217
```

```
 house_mod = subset(house_tst, house_tst$OverallQual != "2" & house_tst$Neighborhood !="Blueste" & house_tst$Neigh
borhood !="Veenker" & house_tst$ExterQual !="Fa" )

(RMSE_test_log_cd = sqrt(mean((log(house_tst$SalePrice) - predict(int_add_cd_mod,house_mod)) ^ 2)))
```

```
## [1] 1.336055
```

Discussion on test results - Log interaction model with influential points removed

The above log interaction model without influential points have the following test results

- R^2 value of 0.9295, ie 92.95% of the variance in SalePrice is explained by linear relationship with variables in the `additive model`.

- Normality assumptions and the constant variance assumptions is **valid** for this model which we could see from the QQ plot and fitted vs residual plot.The high p-value of `BP test` and `Shapiro-wilk test` confirms the same.
- Train RMSE of 0.0926
- Test RMSE of 1.3361

This model has improved R^2 value and the model assumptions are also valid but there 378 parameters, which will make the model interpretation difficult.

Now, we will focus on **reducing the model parameters** by using the AIC/BIC search methods.

BIC of Log Interation Model

In this section, we will do a BIC step search to reduce the number of parameters.

```
n = length(resid(int_add_cd_mod))

BIC_Log_cd = step(int_add_cd_mod,k=log(n),trace=0)
```

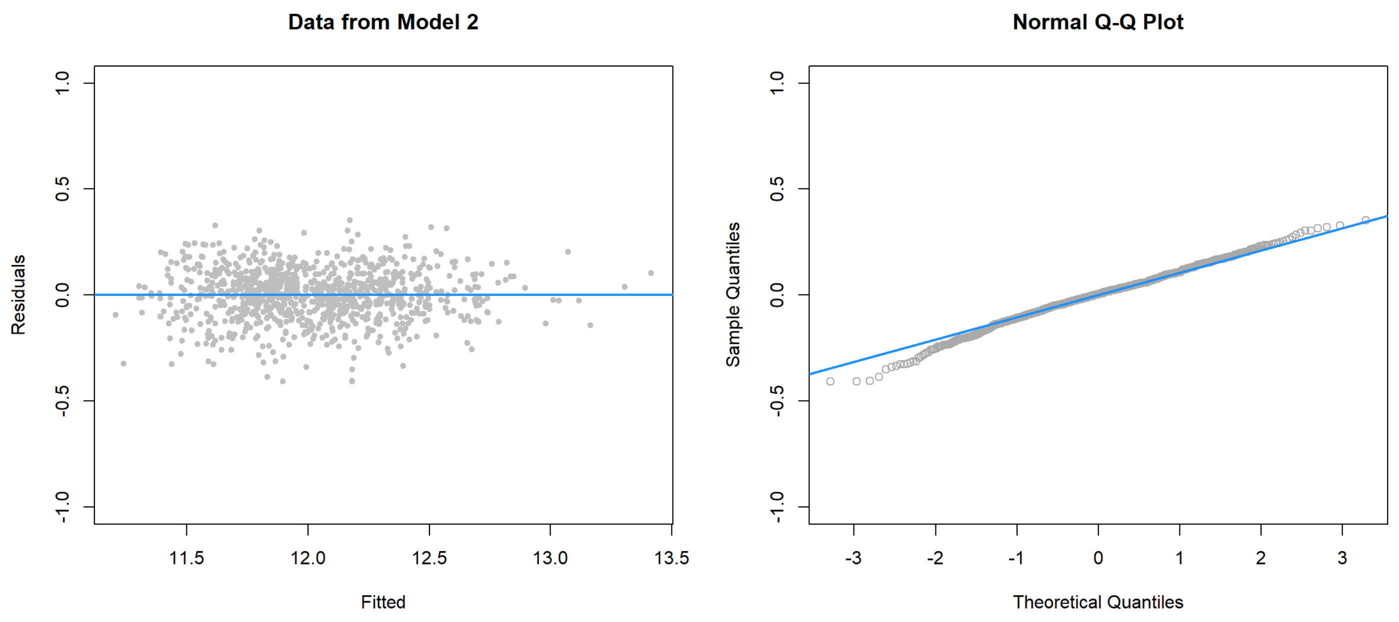
R-Squared

```
(r2_bic=summary(BIC_Log_cd)$r.squared)
```

```
## [1] 0.8883116
```

Q-Q and Fitted vs Residual Plot

```
 par(mfrow = c(1, 2))
plot(fitted(BIC_Log_cd), resid(BIC_Log_cd), col = "grey", pch = 20,ylim = c(-1, 1),
    xlab = "Fitted", ylab = "Residuals", main = "Data from Model 2")
abline(h = 0, col = "dodgerblue", lwd = 2)

qqnorm(resid(BIC_Log_cd), main = "Normal Q-Q Plot", col = "darkgrey",ylim = c(-1, 1))
qqline(resid(BIC_Log_cd), col = "dodgerblue", lwd = 2)
```



BP and Shapiro Test

```
bptest(BIC_Log_cd)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  BIC_Log_cd
## BP = 121.46, df = 39, p-value = 2.134e-10
```

```
shapiro.test(resid(BIC_Log_cd))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(BIC_Log_cd)
## W = 0.99312, p-value = 0.0001453
```

Test and Train RMSE

```
(RMSE_Train_BIC_log_cd = sqrt(mean(resid(BIC_Log_cd) ^ 2)))
```

```
## [1] 0.1165114
```

```
 house_mod = subset(house_tst, house_tst$OverallQual != "2" & house_tst$Neighborhood !="Blueste" & house_tst$Neigh
borhood !="Veenker" & house_tst$ExterQual !="Fa" )

(RMSE_test_BIC_log_cd = sqrt(mean((log(house_tst$SalePrice) - predict(BIC_Log_cd,house_mod)) ^ 2)))
```

```
## [1] 0.4877508
```

Discussion on test results for BIC search Model

The above model built using BIC have the following test results

- R^2 value of 0.8883, ie 88.83% of the variance in SalePrice is explained by linear relationship with variables in the `BIC model` .
- Normality assumptions and the constant variance assumptions are **not valid** for this model which we could see from the QQ plot and fitted vs residual plot.The low p-value of `BP test` and `Shapiro-wilk test` confirms the same.
- Train RMSE of 0.1165
- Test RMSE of 0.4878

This model has reduced R^2 value compared to the log interaction model and log interaction without influential points.But this model has 40parameters which is very less parameters compared to the log interaction model, which make this model interpretable.

Now, we will see if we can use AIC search method to acheive below:

- Less parameters similar to BIC model
- Better R^2
- Valid model assumptions and
- Low train and test RMSE

AIC of Log Interation Model

In this section, we will do a AIC step search to reduce the number of parameters.

```
n = length(resid(int_add_cd_mod))

AIC_Log_cd = step(int_add_cd_mod,trace=0)
```

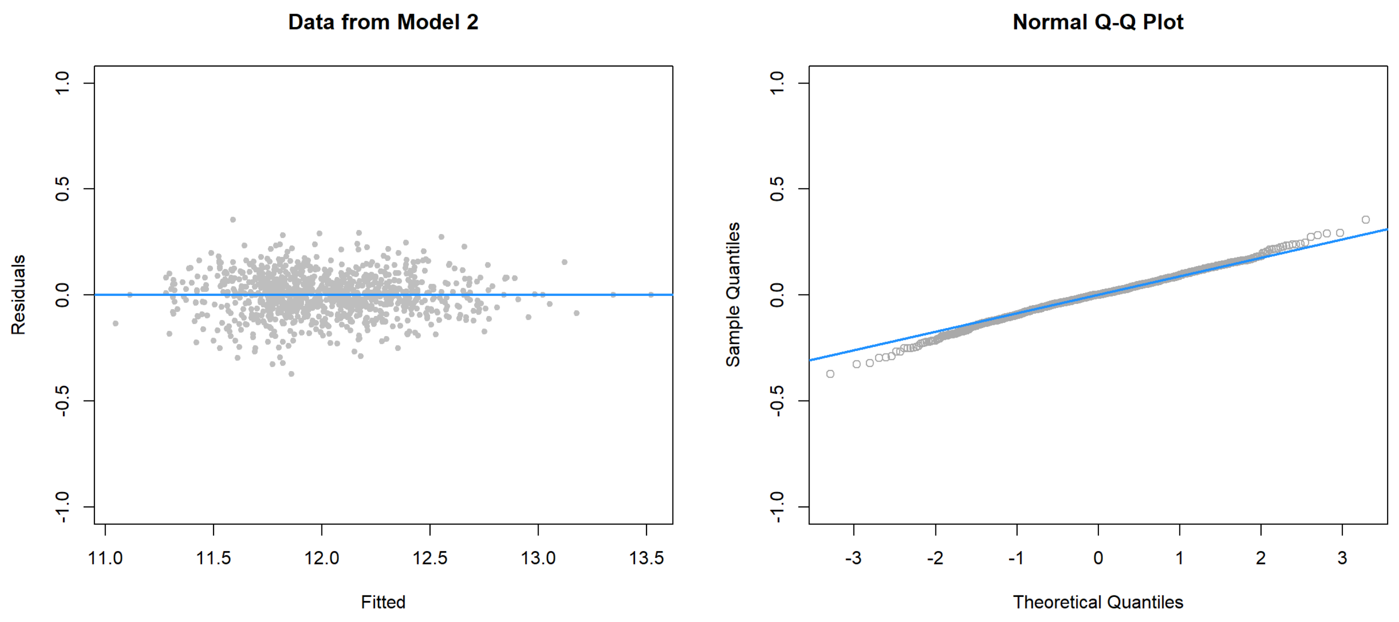
R-Squared

```
(r2_aic=summary(AIC_Log_cd)$r.squared)
```

```
## [1] 0.9229351
```

Q-Q and Fitted vs Residual Plot

```
 par(mfrow = c(1, 2))
plot(fitted(AIC_Log_cd), resid(AIC_Log_cd), col = "grey", pch = 20,ylim = c(-1, 1),
    xlab = "Fitted", ylab = "Residuals", main = "Data from Model 2")
abline(h = 0, col = "dodgerblue", lwd = 2)

qqnorm(resid(AIC_Log_cd), main = "Normal Q-Q Plot", col = "darkgrey",ylim = c(-1, 1))
qqline(resid(AIC_Log_cd), col = "dodgerblue", lwd = 2)
```



BP and Shapiro Test

```
bptest(AIC_Log_cd)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  AIC_Log_cd
## BP = 171.21, df = 156, p-value = 0.1916
```

```
shapiro.test(resid(AIC_Log_cd))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(AIC_Log_cd)
## W = 0.99477, p-value = 0.001615
```

Test and Train RMSE

```
(RMSE_Train_AIC_log_cd = sqrt(mean(resid(AIC_Log_cd) ^ 2)))
```

```
## [1] 0.09678154
```

```
 house_mod = subset(house_tst, house_tst$OverallQual != "2" & house_tst$Neighborhood !="Blueste" & house_tst$Neigh
borhood !="Veenker" & house_tst$ExterQual !="Fa" )

(RMSE_test_AIC_log_cd = sqrt(mean((log(house_tst$SalePrice) - predict(AIC_Log_cd,house_mod)) ^ 2)))
```

```
## [1] 0.4985559
```

Discussion on test results for AIC search Model

The above model built using AIC have the following test results

- R^2 value of 0.9229, ie 92.29% of the variance in SalePrice is explained by linear relationship with variables in the `BIC model` .
- Normality assumptions and the constant variance assumptions are **valid** for this model which we could see from the QQ plot and fitted vs residual plot. The high p-value of `BP test` and `Shapiro-wilk test` confirms the same.
- Train RMSE of 0.0968
- Test RMSE of 0.4986

Also, this model has better R^2 value and this model has 274parameters which is less parameters compared to the log interaction model, which make this model more interpretable than the log interaction model.

Results and Prediction

Preferred model for predicting SalePrice

Based on the discussion points about R^2, RMSE and model assumption discussed in each model section and from the table below we could see that, the model built using the AIC search with log interaction model(**AIC_Log_cd**) is comparitively the good model for predicting the house SalePrice as it has the best R^2 of the 6 models built above and it has comparitively good test RMSE.

Models	R2 Values	Train RMSE	Test RMSE
add_mod	0.8659282	0.1466346	0.1674523
add_mod2	0.8560228	0.1519549	0.1739923
log_int_mod	0.9138646	0.1175327	0.3491662
int_add_cd_mod	0.9294778	0.0925822	1.3360547
BIC_Log_cd	0.8883116	0.1165114	0.4877508
AIC_Log_cd	0.9229351	0.0967815	0.4985559

The AIC log interaction model also has valid constant variance and normality assumption, which makes the model good for interpretation purpose but it may be little difficuly to use the AIC_Log_cd model for interpretation, as it has large number of parameters.

The aim of this project is to build a priction model,usually if we are building a models for prediction purpose we can select the model based on the R^2 and RMSE values alone as the model will be good for prediction and we shouldn't be concerned about the number of parameter.

Prediction on test data

Using our preferred model, lets predict the SalesPrice for the houses in 'Gilbert` neigborhood in test data set and see if the predicted price is matching with the similar kind of house in training data set.

Predicted SalePrice for houses in Gilbert Neighborhood

	GrLivArea	GarageArea	OverallQual	YearBuilt	YearRemodAdd	Neighborhood	TotRmsAbvGrd	ExterQual	SalePrice
1463	1629	482	5	1997	1998	Gilbert	6	TA	144977.4

	GrLivArea	GarageArea	OverallQual	YearBuilt	YearRemodAdd	Neighborhood	TotRmsAbvGrd	ExterQual	SalePrice
1464	1604	470	6	1998	1998	Gilbert	7	TA	180924.6
1466	1655	440	6	1993	1994	Gilbert	7	TA	176284.2
1467	1187	420	6	1992	2007	Gilbert	6	TA	160625.6
1468	1465	393	6	1998	1998	Gilbert	7	TA	175115.1
1469	1341	506	7	1990	1990	Gilbert	5	TA	162938.6
1483	1324	430	6	2005	2005	Gilbert	6	Gd	169230.6
1485	1374	400	7	2004	2004	Gilbert	7	Gd	181816.4
1486	1733	433	7	2004	2004	Gilbert	7	Gd	207771.2
1627	1608	470	6	1997	1997	Gilbert	7	TA	179627.4

Real SalePrice for houses in Gilbert Neighborhood from train data

	GrLivArea	GarageArea	OverallQual	YearBuilt	YearRemodAdd	Neighborhood	TotRmsAbvGrd	ExterQual	SalePrice
51	1470	388	6	1997	1997	Gilbert	6	TA	177000
73	1718	427	7	1998	1998	Gilbert	7	TA	185000
85	1474	400	7	1995	1996	Gilbert	7	TA	168500
87	1560	400	6	2005	2005	Gilbert	6	Gd	174000
96	1470	420	6	1993	1993	Gilbert	6	Ex	185000
112	1430	400	7	2000	2000	Gilbert	7	TA	180000
132	2054	390	6	2000	2000	Gilbert	7	Gd	244000
148	2035	434	7	2001	2001	Gilbert	8	Gd	222500
160	2462	576	7	2005	2006	Gilbert	9	Gd	320000
169	1720	440	7	2004	2004	Gilbert	7	Gd	183500

Conclusion

From the above tables, we could see that the Saleprice predicted from the test data set for the houses in Gilbert neighborhood are having values close to that of the saleprice of houses in Gilbert neighborhood in the train dataset. So our model is doing a pretty job in predicting the SalePrice.