Real Estate Analysis

1. Problem Statement

The aim of this project is to predict price of a real estate property, by analyzing a wide range of house features such as dwelling type, flatness, location, roof material, veneer type, number of rooms, and condition of basement. The project also aims to find out the set of variables that has the most impact on the house price.

2. Data Import and Check

Libraries needed

```
library(knitr)
library(dplyr)
library(corrplot)
library(ggplot2)
library(Metrics)
library(lars)
library(randomForest)
library(data.table)
library(ragg)
```

Data Import

```
train <- read.csv("house_price.csv", stringsAsFactors=FALSE)
test <- read.csv("house_price_test.csv", stringsAsFactors=FALSE)
```

```
names(train)
```

```
##  [1] "Id"            "MSSubClass"    "MSZoning"      "LotFrontage"  
##  [5] "LotArea"       "Street"        "Alley"         "LotShape"     
##  [9] "LandContour"   "Utilities"     "LotConfig"     "LandSlope"    
## [13] "Neighborhood"  "Condition1"    "Condition2"    "BldgType"     
## [17] "HouseStyle"    "OverallQual"   "OverallCond"   "YearBuilt"    
## [21] "YearRemodAdd"  "RoofStyle"     "RoofMatl"      "Exterior1st"  
## [25] "Exterior2nd"   "MasVnrType"    "MasVnrArea"    "ExterQual"    
## [29] "ExterCond"     "Foundation"    "BsmtQual"      "BsmtCond"     
## [33] "BsmtExposure"  "BsmtFinType1"  "BsmtFinSF1"    "BsmtFinType2" 
## [37] "BsmtFinSF2"    "BsmtUnfSF"     "TotalBsmtSF"   "Heating"      
## [41] "HeatingQC"     "CentralAir"    "Electrical"    "X1stFlrSF"    
## [45] "X2ndFlrSF"     "LowQualFinSF"  "GrLivArea"     "BsmtFullBath" 
## [49] "BsmtHalfBath"  "FullBath"      "HalfBath"      "BedroomAbvGr" 
## [53] "KitchenAbvGr"  "KitchenQual"   "TotRmsAbvGrd"  "Functional"   
## [57] "Fireplaces"    "FireplaceQu"   "GarageType"    "GarageYrBlt"  
## [61] "GarageFinish"  "GarageCars"    "GarageArea"    "GarageQual"   
## [65] "GarageCond"    "PavedDrive"    "WoodDeckSF"    "OpenPorchSF"  
## [69] "EnclosedPorch" "X3SsnPorch"    "ScreenPorch"   "PoolArea"     
## [73] "PoolQC"        "Fence"         "MiscFeature"   "MiscVal"      
## [77] "MoSold"        "YrSold"        "SaleType"      "SaleCondition"
## [81] "SalePrice"
```

```
dim(train)
```

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

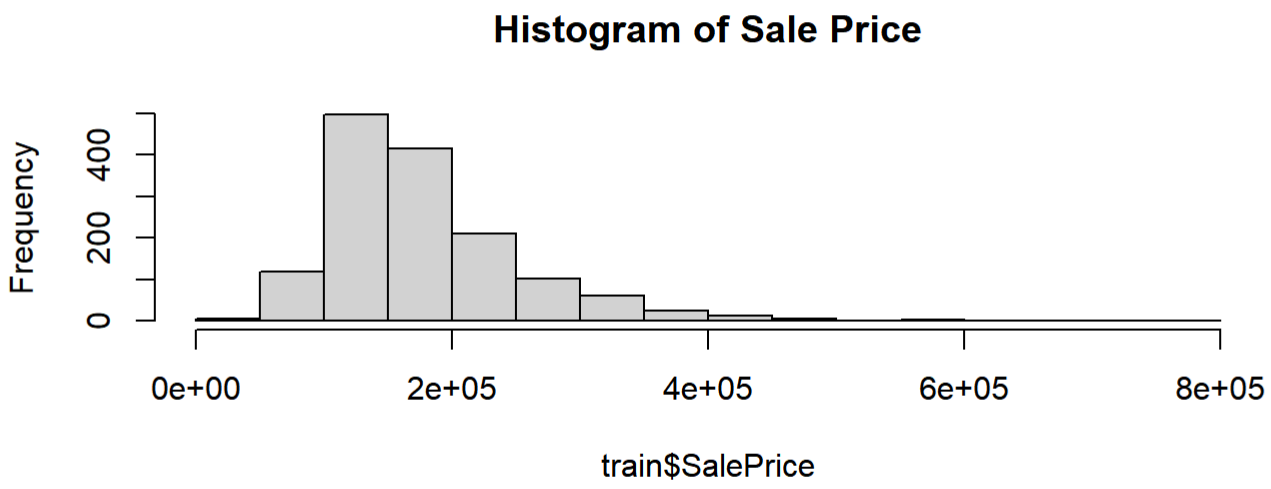
```
str(train[,c(1:10, 81)])
```

```
## 'data.frame':    1460 obs. of  11 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" ...
##  $ SalePrice  : int  208500 181500 223500 140000 250000 143000 307000 200000 129900 118000 ...
```

There are 1 predictor variable (SalePrice) and 79 explanatory variables excluding ID variable. The predictor variable is continuous numeric variable, so I will build a linear model.

Predictor Variable

```
hist(train$SalePrice, main="Histogram of Sale Price")
```



The histogram shows heavy right skewness in the data. we can also see some potential outliers. It seems that there were a few people influencing the price of expensive houses.

```
summary(train$SalePrice)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   34900  129975  163000  180921  214000  755000
```

75 percent of the houses are below 214,000. The median is 163,000 and mean is 180,921. There is a huge gap between these values and the maximum value.

Numeric Variables

```
numericVars <- which(sapply(train, is.numeric))
numericVarNames <- names(numericVars)
cat('There are', length(numericVars), 'numeric variables')
```

```
## There are 38 numeric variables
```

3. Data Pre-processing

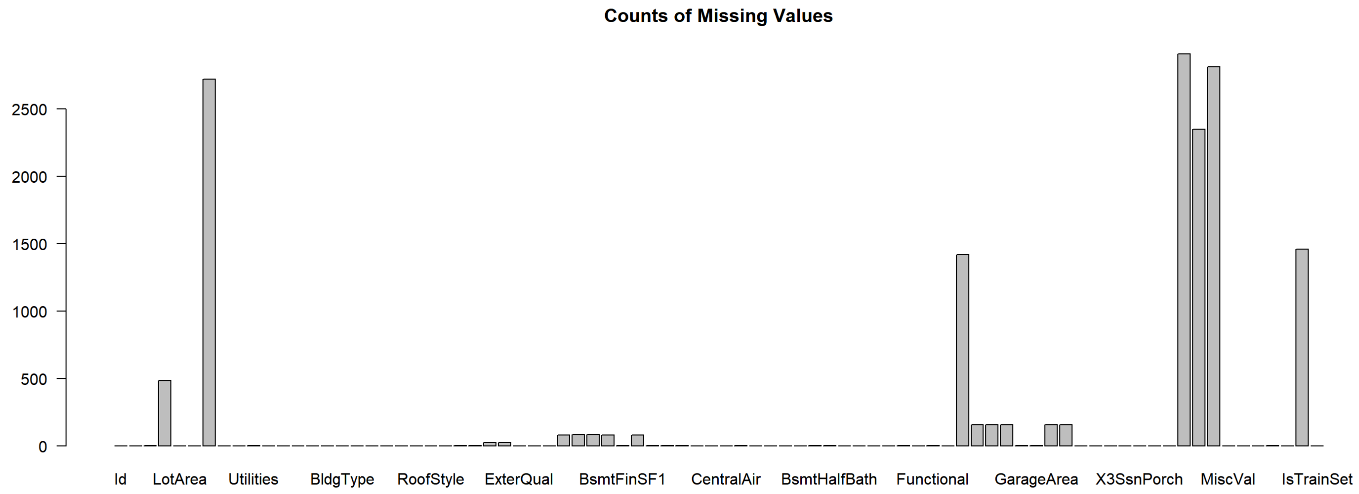
```
# combine the dataset
train$IsTrainSet <- TRUE
test$IsTrainSet <- FALSE
test$SalePrice <- NA
house.full <- rbind(train, test)
```

Missing Values

After summarizing the training set, we found that some columns got too many missing values.

```
Num_NA<-sapply(house.full, function(y)length(which(is.na(y)==T)))
NA_Count<- data.frame(Item=colnames(train),Count=Num_NA)

barplot(height=NA_Count$Count, names=NA_Count$Item,
        hotiz=T, las=1, main= "Counts of Missing Values")
```



```
NA_Count$Item[NA_Count$Count>800]
```

```
## [1] "Alley"       "FireplaceQu" "PoolQC"      "Fence"       "MiscFeature"
## [6] "SalePrice"
```

Among 79 variables, “Alley”, “PoolQC”, “Fence” and “MiscFeature” have amazingly high number of missing values.

```
house.full$Alley <- NULL
house.full$PoolQC <- NULL
house.full$Fence <- NULL
house.full$MiscFeature <- NULL
dim(house.full)
```

```
## [1] 2919   78
```

I have decided to remove those variables. After that, the number of effective variables has shrunken to 75 (excluding id).

Then, I created the dataset ‘Num’ to sort out the numeric variables in particular for the convenience of descriptive analysis.

```
# Numeric Variables
Num<-sapply(house.full,is.numeric)
Num<-house.full[,Num]

dim(Num)
```

```
## [1] 2919   38
```

```
for(i in 1:77){
  if(is.character(house.full[,i])){
    house.full[,i] <- as.factor(house.full[,i])
  }
}

for(i in 1:77){
  if(is.factor(house.full[,i])){
    house.full[,i] <- as.integer(house.full[,i])
  }
}

# Test
house.full$Street[1:50]
```

```
##  [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [39] 2 2 2 2 2 2 2 2 2 2 2 2
```

Finally, for the remaining missing values, I decided to replace them with zero directly.

```
house.full[is.na(house.full)] <- 0
Num[is.na(Num)] <- 0
```

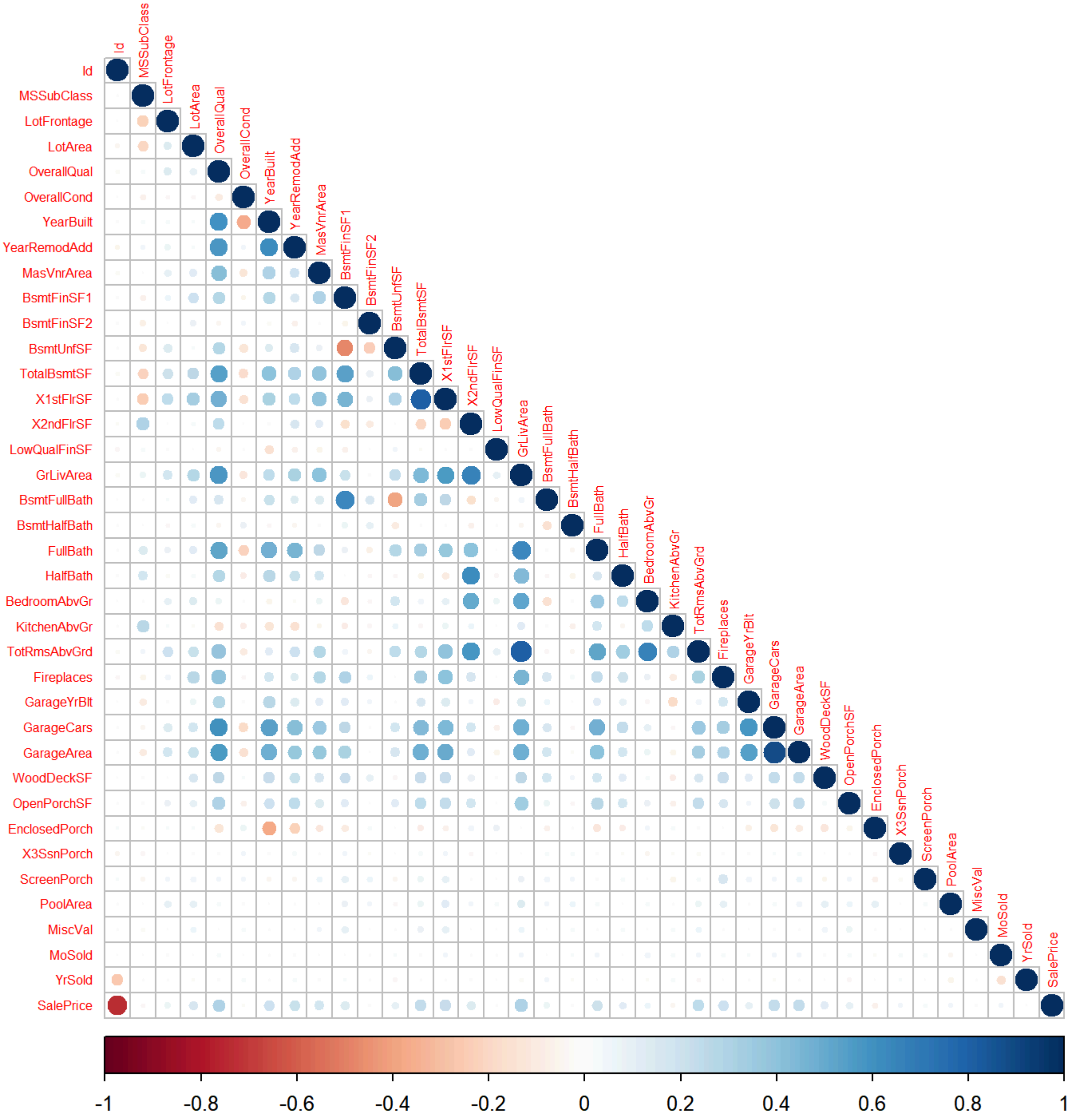
```
# Separate the dataset
house.train <- house.full[house.full$IsTrainSet == TRUE, ]
house.test <- house.full[house.full$IsTrainSet == FALSE, ]
```

3. Descriptive Analysis

Correlation Matrix

I first draw a corrplot of numeric variables. Those with strong correlation with sale price are examined.

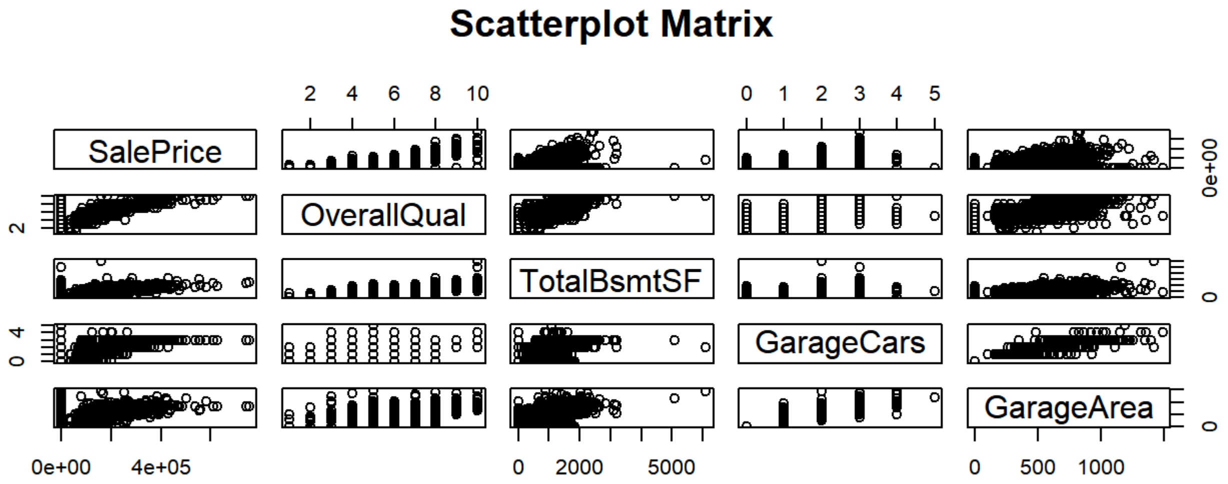
```
correlations<- cor(Num, use="everything")
corrplot(correlations, method="circle", type="lower", sig.level = 0.01, insig = "blank",
         tl.cex = 0.5)
```



According to the plot, ‘OverallQual’,‘TotalBsmtSF’,‘GarageCars’ and ‘GarageArea’ have relative strong correlation with each other. Therefore, as an example, we plot the correlation among those four variables and SalePrice.

Scatterplot Matrix

```
pairs(~SalePrice+OverallQual+TotalBsmtSF+GarageCars+GarageArea,data=house.full,
      main="Scatterplot Matrix")
```

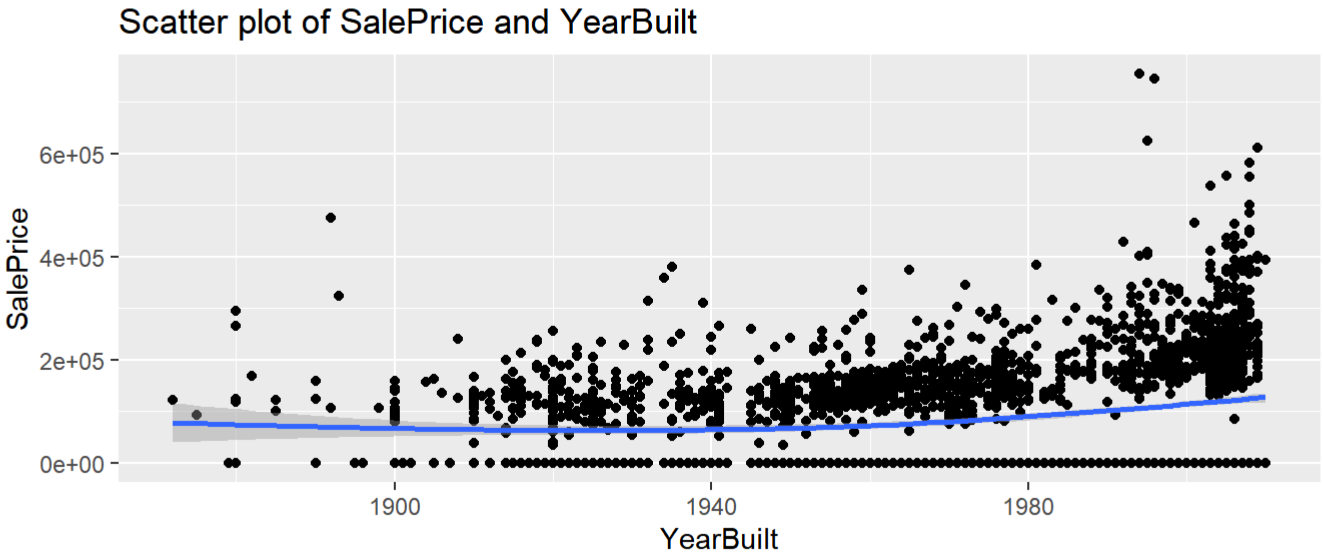


The dependent variable (SalePrice) looks having decent linearity when plotting with other variables. However, it is also obvious that some independent variables also have linear relationship with others. The problem of multicollinearity is obvious and should be treated when the quantity of variables in regression formula is huge.

The final descriptive analysis I put here would be the relationship between year variables and Sale Price

Sale Price and Year Built

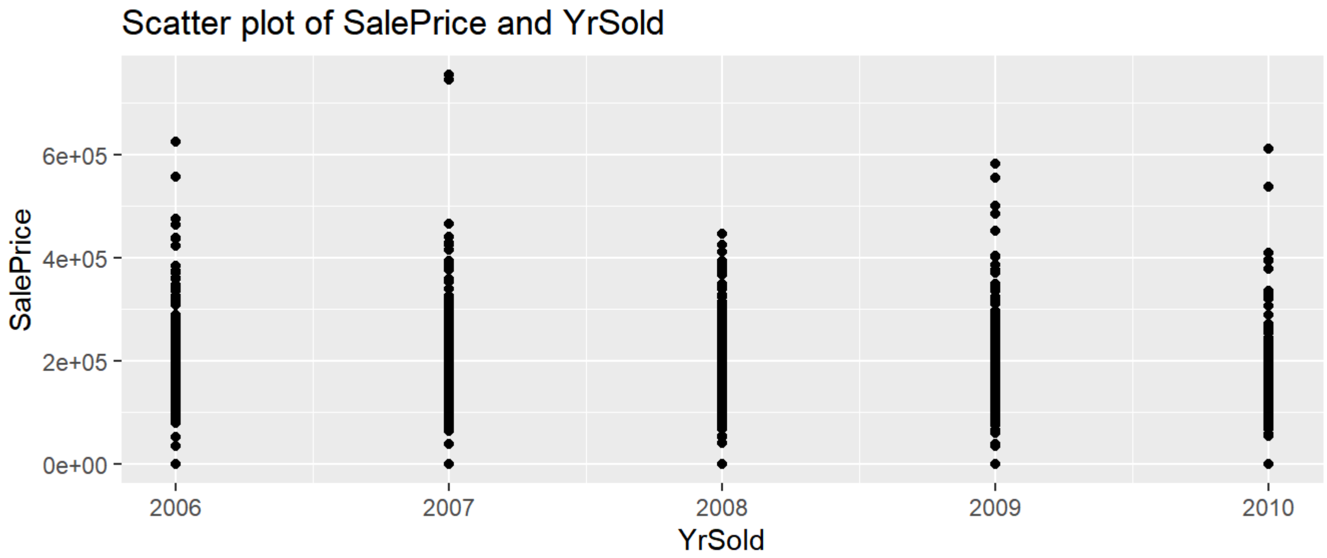
```
p<- ggplot(house.full,
           aes(x= YearBuilt, y=SalePrice))+geom_point()+geom_smooth()
p + ggtitle("Scatter plot of SalePrice and YearBuilt")
```



We can see that there is an increasing trend between the house price and the year built.

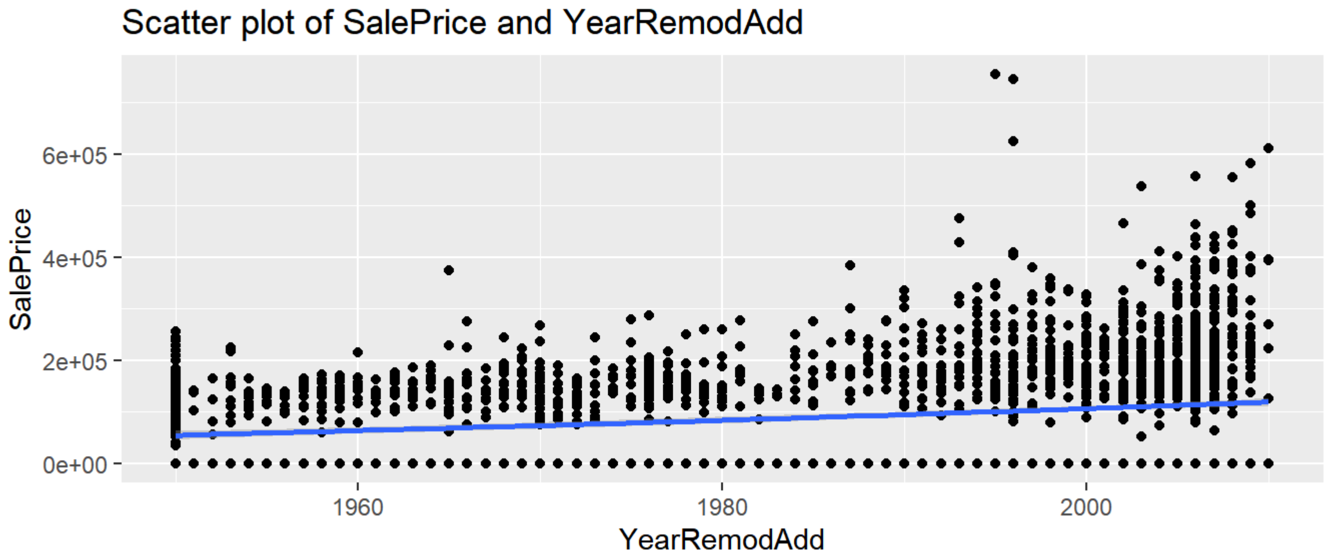
Sale Price and Year Sold

```
p<- ggplot(house.full,
           aes(x= YrSold, y=SalePrice))+geom_point()+geom_smooth()
p + ggtitle("Scatter plot of SalePrice and YrSold")
```


Scatter plot of SalePrice and YrSold

Sale Price and Year Remodeled

```
p<- ggplot(house.full,
           aes(x=YearRemodAdd, y=SalePrice))+geom_point()+geom_smooth()
p + ggtitle("Scatter plot of SalePrice and YearRemodAdd")
```


Scatter plot of SalePrice and YearRemodAdd

There doesn’t seem to be a relationship between the price of the houses and the year when the houses were sold. However, it appears that there is some relationship between their price and the year when they were remodeled.

4. Model Building

Before implementing models, I first divided the data set into 2 parts: a training set and a test set that can be used for evaluation. For this analysis, I decided to split it with the ratio of 6:4.

```
# Split the data into Training and Test Set Ratio: 6:4
Training_Inner<- house.train[1:floor(length(house.train[,1])*0.6),]
Test_Inner<- house.train[(length(Training_Inner[,1])+1):1460,]
```

I will fit three regression models to the training set and choose the most suitable one by checking RMSE value.

Model 1: Linear Regression

The first and simplest but useful model is linear regression model. As the first step, I put all variables into the model.

```
reg1<- lm(SalePrice~., data = Training_Inner)
summary(reg1)
```

```
##
## Call:
## lm(formula = SalePrice ~ ., data = Training_Inner)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -353089  -12679   -1303   12824  192934
##
## Coefficients: (4 not defined because of singularities)
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.727e+06  1.577e+06   1.095 0.273758
## Id             5.145e+00  4.020e+00   1.280 0.200927
## MSSubClass    -7.137e+01  5.163e+01  -1.382 0.167288
## MSZoning      -6.364e+02  1.736e+03  -0.367 0.714012
## LotFrontage    6.470e+01  3.604e+01   1.795 0.073004 .
## LotArea        4.153e-01  1.070e-01   3.882 0.000112 ***
## Street         3.691e+04  1.589e+04   2.323 0.020440 *
## LotShape       4.445e+01  8.016e+02   0.055 0.955793
## LandContour    7.364e+02  1.642e+03   0.449 0.653842
## Utilities             NA         NA      NA       NA
## LotConfig     -6.138e+02  6.372e+02  -0.963 0.335765
## LandSlope      2.892e+03  4.470e+03   0.647 0.517902
## Neighborhood   3.793e+02  1.855e+02   2.045 0.041205 *
## Condition1    -1.559e+03  1.232e+03  -1.265 0.206123
## Condition2    -1.396e+04  3.704e+03  -3.768 0.000177 ***
## BldgType      -1.807e+03  1.719e+03  -1.051 0.293528
## HouseStyle    -4.448e+02  7.872e+02  -0.565 0.572191
## OverallQual    1.141e+04  1.419e+03   8.041 3.18e-15 ***
## OverallCond    3.711e+03  1.273e+03   2.916 0.003642 **
## YearBuilt      1.703e+02  9.029e+01   1.886 0.059677 .
## YearRemodAdd   1.613e+02  7.953e+01   2.028 0.042881 *
## RoofStyle     -2.340e+02  1.378e+03  -0.170 0.865179
## RoofMatl       4.512e+03  1.698e+03   2.658 0.008023 **
## Exterior1st   -9.145e+02  5.870e+02  -1.558 0.119652
## Exterior2nd    5.244e+02  5.397e+02   0.972 0.331482
## MasVnrType     2.804e+03  1.801e+03   1.556 0.120022
## MasVnrArea     2.041e+01  6.895e+00   2.960 0.003163 **
## ExterQual     -1.039e+04  2.300e+03  -4.516 7.25e-06 ***
## ExterCond      1.164e+03  1.451e+03   0.802 0.422761
## Foundation     8.281e+02  2.005e+03   0.413 0.679632
## BsmtQual      -5.292e+03  1.561e+03  -3.391 0.000730 ***
## BsmtCond       2.841e+03  1.571e+03   1.809 0.070836 .
## BsmtExposure  -1.741e+03  1.059e+03  -1.645 0.100461
## BsmtFinType1  -6.466e+02  7.383e+02  -0.876 0.381417
## BsmtFinSF1     3.276e+01  6.506e+00   5.035 5.92e-07 ***
## BsmtFinType2   3.471e+02  1.261e+03   0.275 0.783161
## BsmtFinSF2     3.321e+01  9.484e+00   3.502 0.000488 ***
## BsmtUnfSF      1.880e+01  6.286e+00   2.990 0.002873 **
## TotalBsmtSF           NA         NA      NA       NA
## Heating        8.894e+02  4.435e+03   0.201 0.841110
## HeatingQC     -4.339e+02  7.401e+02  -0.586 0.557848
## CentralAir     3.364e+02  5.438e+03   0.062 0.950687
## Electrical     -1.552e+02  1.065e+03  -0.146 0.884206    
## X1stFlrSF       4.090e+01  7.292e+00   5.609 2.80e-08 ***
## X2ndFlrSF       4.442e+01  5.716e+00   7.771 2.38e-14 ***
## LowQualFinSF    1.412e+01  2.072e+01   0.682 0.495722    
## GrLivArea              NA         NA      NA       NA    
## BsmtFullBath    3.577e+03  2.956e+03   1.210 0.226697    
## BsmtHalfBath    8.845e+02  4.631e+03   0.191 0.848583    
## FullBath        1.571e+03  3.192e+03   0.492 0.622820    
## HalfBath       -1.432e+02  3.039e+03  -0.047 0.962442    
## BedroomAbvGr   -7.237e+03  1.989e+03  -3.638 0.000292 ***
## KitchenAbvGr   -2.120e+04  6.048e+03  -3.506 0.000481 ***
## KitchenQual    -6.035e+03  1.712e+03  -3.526 0.000446 ***
## TotRmsAbvGrd    5.880e+03  1.429e+03   4.115 4.28e-05 ***
## Functional      3.443e+03  1.116e+03   3.084 0.002112 ** 
## Fireplaces      8.333e+03  2.886e+03   2.888 0.003982 ** 
## FireplaceQu    -1.838e+03  8.944e+02  -2.055 0.040163 *  
## GarageType      1.503e+03  7.585e+02   1.981 0.047928 *  
## GarageYrBlt    -1.238e+01  6.951e+00  -1.781 0.075301 .  
## GarageFinish   -1.084e+03  1.732e+03  -0.626 0.531760    
## GarageCars      5.398e+03  3.421e+03   1.578 0.114970    
## GarageArea      2.444e+01  1.146e+01   2.132 0.033311 *  
## GarageQual     -1.988e+03  2.104e+03  -0.945 0.344931    
## GarageCond      2.155e+03  2.436e+03   0.885 0.376474    
## PavedDrive      4.501e+03  2.523e+03   1.784 0.074791 .  
## WoodDeckSF      2.218e+01  8.999e+00   2.464 0.013935 *  
## OpenPorchSF    -1.909e+01  1.673e+01  -1.141 0.254095    
## EnclosedPorch   2.808e+00  1.845e+01   0.152 0.879115    
## X3SsnPorch      8.680e+00  3.110e+01   0.279 0.780229    
## ScreenPorch     5.784e+01  1.861e+01   3.108 0.001951 ** 
## PoolArea       -4.756e+01  3.736e+01  -1.273 0.203345    
## MiscVal         3.482e-01  1.799e+00   0.194 0.846521    
## MoSold         -6.543e+02  3.799e+02  -1.722 0.085389 .  
## YrSold         -1.208e+03  7.821e+02  -1.544 0.122944    
## SaleType       -3.676e+02  6.456e+02  -0.569 0.569284    
## SaleCondition   4.554e+03  1.005e+03   4.529 6.81e-06 ***
## IsTrainSetTRUE         NA         NA      NA       NA    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 28640 on 802 degrees of freedom
## Multiple R-squared:  0.8856, Adjusted R-squared:  0.8752 
## F-statistic: 85.02 on 73 and 802 DF,  p-value: < 2.2e-16
```

R squared is not bad. This full model can explain about 88 percent of the variability.

However, many variables do not have statistical significance and there is a potential over fitting problem. Therefore, the variable selection process is should be involved in model construction.

I will manually build a model by checking the result of Hypothesis testing.

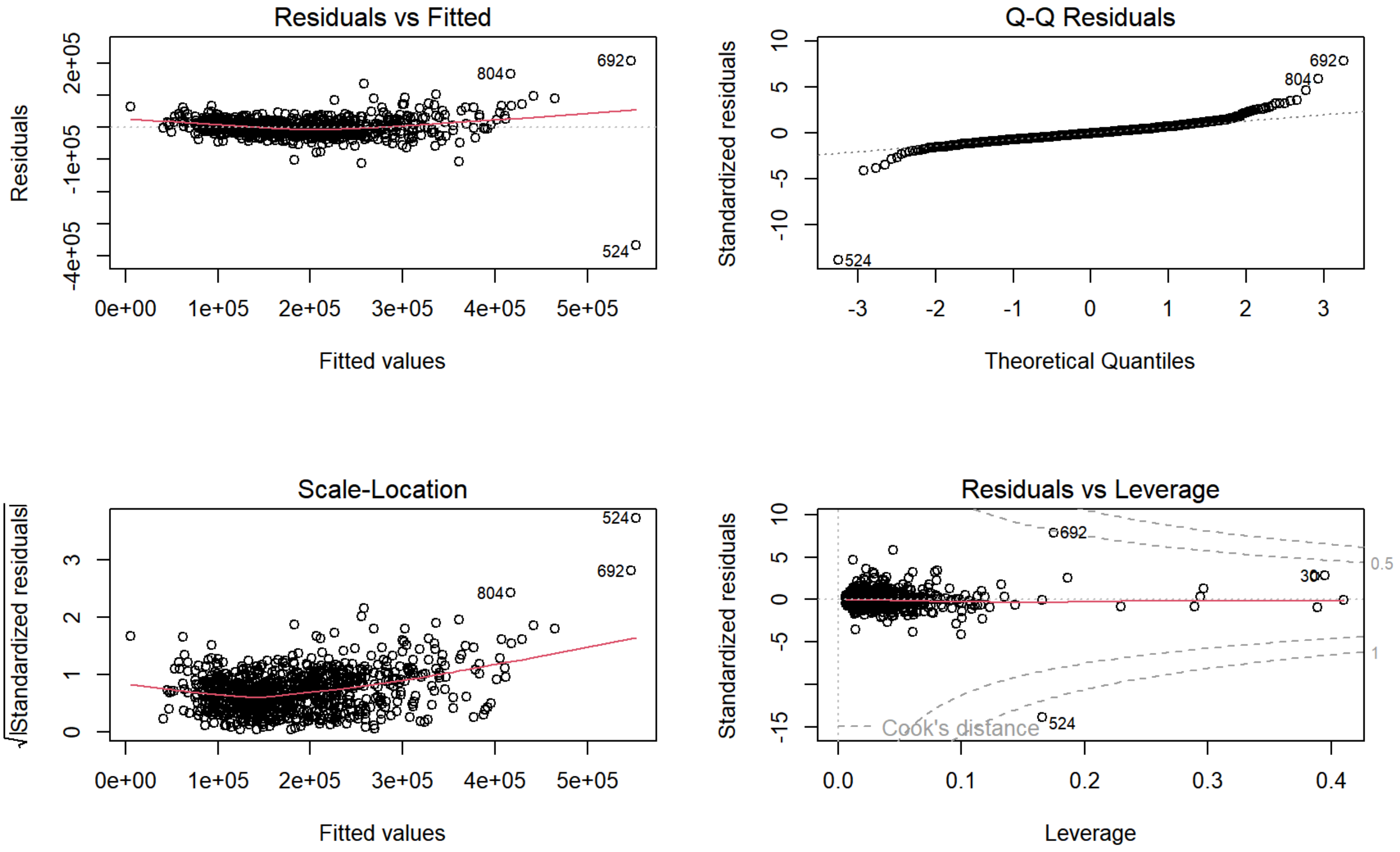
```
reg2 <- lm(SalePrice ~ LotFrontage + LotArea + Street + Condition2 + OverallQual + OverallCond + YearBuilt + RoofMatl + ExterQual +BsmtQual + BsmtFinSF1 + BsmtFinSF2 + BsmtUnfSF + X1stFlrSF + X2ndFlrSF + BedroomAbvGr + KitchenAbvGr + KitchenQual + TotRmsAbvGrd + Functional + Fireplaces + FireplaceQu + GarageYrBlt + GarageArea + WoodDeckSF + ScreenPorch + SaleCondition, Training_Inner)

summary(reg2)
```

```
## 
## Call:
## lm(formula = SalePrice ~ LotFrontage + LotArea + Street + Condition2 + 
##     OverallQual + OverallCond + YearBuilt + RoofMatl + ExterQual + 
##     BsmtQual + BsmtFinSF1 + BsmtFinSF2 + BsmtUnfSF + X1stFlrSF + 
##     X2ndFlrSF + BedroomAbvGr + KitchenAbvGr + KitchenQual + TotRmsAbvGrd + 
##     Functional + Fireplaces + FireplaceQu + GarageYrBlt + GarageArea + 
##     WoodDeckSF + ScreenPorch + SaleCondition, data = Training_Inner)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -367581  -14344   -1044   12192  207483 
## 
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)    
## (Intercept)   -5.880e+05  1.083e+05  -5.430 7.38e-08 ***
## LotFrontage    8.940e+01  3.258e+01   2.744 0.006191 ** 
## LotArea        4.926e-01  9.218e-02   5.344 1.17e-07 ***
## Street         4.476e+04  1.522e+04   2.941 0.003358 ** 
## Condition2    -1.267e+04  3.571e+03  -3.548 0.000410 ***
## OverallQual    1.106e+04  1.332e+03   8.306 3.91e-16 ***
## OverallCond    5.358e+03  1.068e+03   5.017 6.40e-07 ***
## YearBuilt      2.718e+02  5.153e+01   5.275 1.69e-07 ***
## RoofMatl       4.610e+03  1.640e+03   2.811 0.005058 ** 
## ExterQual     -1.121e+04  2.207e+03  -5.081 4.63e-07 ***
## BsmtQual      -5.892e+03  1.353e+03  -4.354 1.50e-05 ***
## BsmtFinSF1     4.102e+01  5.383e+00   7.621 6.75e-14 ***
## BsmtFinSF2     3.333e+01  7.666e+00   4.347 1.55e-05 ***
## BsmtUnfSF      2.279e+01  5.426e+00   4.200 2.95e-05 ***
## X1stFlrSF      4.041e+01  6.596e+00   6.127 1.37e-09 ***
## X2ndFlrSF      4.246e+01  4.385e+00   9.683  < 2e-16 ***
## BedroomAbvGr  -7.298e+03  1.872e+03  -3.899 0.000104 ***
## KitchenAbvGr  -2.782e+04  5.277e+03  -5.272 1.71e-07 ***
## KitchenQual   -6.308e+03  1.652e+03  -3.819 0.000143 ***
## TotRmsAbvGrd   6.936e+03  1.364e+03   5.084 4.55e-07 ***
## Functional     3.414e+03  1.088e+03   3.137 0.001765 ** 
## Fireplaces     9.207e+03  2.841e+03   3.241 0.001238 ** 
## FireplaceQu   -2.089e+03  8.690e+02  -2.404 0.016416 *  
## GarageYrBlt   -8.822e+00  2.847e+00  -3.099 0.002009 ** 
## GarageArea     4.504e+01  7.768e+00   5.798 9.47e-09 ***
## WoodDeckSF     2.333e+01  8.836e+00   2.640 0.008433 ** 
## ScreenPorch    5.986e+01  1.824e+01   3.283 0.001071 ** 
## SaleCondition  4.802e+03  9.749e+02   4.926 1.01e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 28990 on 848 degrees of freedom
## Multiple R-squared:  0.876,  Adjusted R-squared:  0.872 
## F-statistic: 221.9 on 27 and 848 DF,  p-value: < 2.2e-16
```

Still, the R-squared is not bad. Now, all the variables are significant.

```
layout(matrix(c(1,2,3,4), 2, 2, byrow = TRUE))
plot(reg2)
```



```
#par(mfrow=c(1,1))
par(mar=c(0,0,0,0))
```

The diagnosis of residuals is also not bad. The normal Q-Q plot indicates normality.

We check the performance of linear regression model with RMSE value.

Model Performance

```
Prediction_1<- predict(reg2, newdata= Test_Inner)
mse_lr <- mse(log(Test_Inner$SalePrice),log(Prediction_1))
rmse_lr <- rmse(log(Test_Inner$SalePrice),log(Prediction_1))
cat('MSE (Mean Square Error): ', mse_lr, '₩n','RMSE (Root Mean Square Error): ', rmse_lr)
```

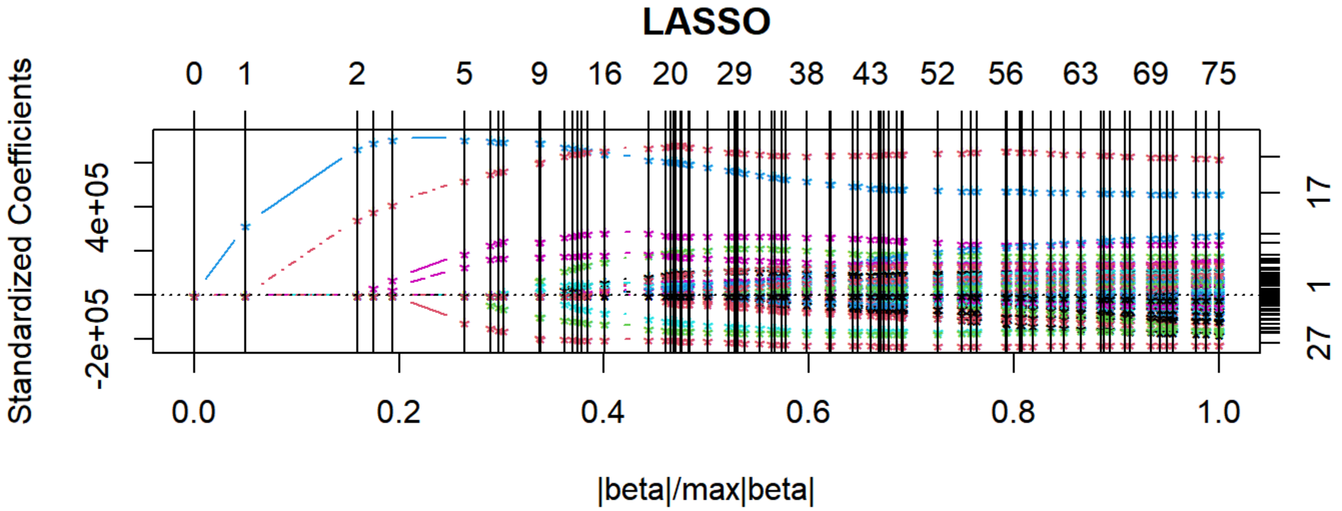
```
## MSE (Mean Square Error):  0.03353686
##  RMSE (Root Mean Square Error):  0.1831307
```

Model 2: Lasso Regression

For the avoidance of multi-colinearity, implementing LASSO regression is not a bad idea. Transferring the variables into the form of matrix, we can automate the selection of variables by implementing 'lars' method in Lars package.

```
Independent_variable <- as.matrix(Training_Inner[,1:76])
Dependent_variable <- as.matrix(Training_Inner[,77])

laa<- lars(Independent_variable, Dependent_variable, type = 'lasso')
plot(laa)
```



The plot is messy as the quantity of variables is intimidating. Despite that, we can still use R to find out the model with least multi-collinearity. The selection procedure is based on the value of Marrow's cp, an important indicator of multi-colinearity. The prediction can be done by the script-chosen best step and RMSE can be used to assess the model.

Model Performance

```
best_step <- laa$df[which.min(laa$Cp)]
Prediction_2<- predict.lars(laa, newx =as.matrix(Test_Inner[,1:76]), s=best_step, type= "fit")
mse_lasso <- rmse(log(Test_Inner$SalePrice),log(Prediction_2$fit))
rmse_lasso <- rmse(log(Test_Inner$SalePrice),log(Prediction_2$fit))
cat('MSE (Mean Square Error): ', mse_lasso, '₩n','RMSE (Root Mean Square Error): ', rmse_lasso)
```

```
## MSE (Mean Square Error):  0.1623512
##  RMSE (Root Mean Square Error):  0.1623512
```

Model 3: Random Forest

The other model I chose to fit in the training set is Random Forest model. The model, prediction and RMSE calculation can be found below:

```
test <- as.data.frame(test)
for_1 <- randomForest(SalePrice~., data=Training_Inner)
prediction_3 <- predict(for_1, newdata=Test_Inner)
mse_rf <- mse(log(Test_Inner$SalePrice), log(prediction_3))
rmse_rf <- rmse(log(Test_Inner$SalePrice), log(prediction_3))
cat('MSE (Mean Square Error): ', mse_rf, '₩n','RMSE (Root Mean Square Error): ', rmse_rf)
```

```
## MSE (Mean Square Error):  0.02213802 
##  RMSE (Root Mean Square Error):  0.1487885
```

5. Final Prediction

Among the three models, Random Forest may produce the best result within the training set.

Summary of Final Model

```
print(for_1)
```

```
## 
## Call:
##  randomForest(formula = SalePrice ~ ., data = Training_Inner) 
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 25
## 
##           Mean of squared residuals: 999586359
##                     % Var explained: 84.77
```

```
pred <- predict(for_1, newdata=house.test)
output.df <- as.data.frame(pred)
output.df <- tibble::rownames_to_column(output.df, "row_names")
output.df <- setnames(output.df, old = c('row_names', 'pred'),
         new = c('Id', 'SalePrice'))
```

6. Conclusion

We were interested in investigating the relationship between the house price and a list of variables and building a model to predict house prices.

we conclude that our final model is Random Forest because of better performance. Using Random Forest, the model explains about 85.13 percent of the variance.

The performance of the model is: * MSE (Mean Square Error): 0.02227288 * RMSE (Root Mean Square Error): 0.149241