Import libraries

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.3     ✓ readr     2.1.4
## ✓ forcats   1.0.0     ✓ stringr   1.5.0
## ✓ ggplot2   3.4.3     ✓ tibble    3.2.1
## ✓ lubridate 1.9.3     ✓ tidyr     1.3.0
## ✓ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✗ dplyr::filter() masks stats::filter()
## ✗ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
e errors
```

```
library(dplyr)
library(tibble)
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 4.3.2
```

```
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
##
## Loaded glmnet 4.1-8
```

```
library(reshape2)
```

```
## Warning: package 'reshape2' was built under R version 4.3.2
```

```
##
## Attaching package: 'reshape2'
##
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```r
library(faraway)
library(lmtest)
```

```
## Loading required package: zoo
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
library(ggplot2)
library(MASS)
```

```
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
```

Load the dataset

For the data documentation, click here (https://jse.amstat.org/v19n3/decock/DataDocumentation.txt).

```r
houses_df = read.csv("ames_houses_data.csv")

# The professor asked us to only use the first 1000 observations
houses_df = houses_df[1:1000,]

sprintf("The dataset has %d rows", nrow(houses_df))
```

```
## [1] "The dataset has 1000 rows"
```

```r
sprintf("The dataset has %d columns", ncol(houses_df))
```

```
## [1] "The dataset has 82 columns"
```

```r
head(houses_df)
```

```
##   Order  price       PID area MS.SubClass MS.Zoning Lot.Frontage Lot.Area
## 1     1 215000 526301100 1656          20        RL          141    31770
## 2     2 105000 526350040  896          20        RH           80    11622
## 3     3 172000 526351010 1329          20        RL           81    14267
## 4     4 244000 526353030 2110          20        RL           93    11160
## 5     5 189900 527105010 1629          60        RL           74    13830
## 6     6 195500 527105030 1604          60        RL           78     9978
##   Street Alley Lot.Shape Land.Contour Utilities Lot.Config Land.Slope
## 1   Pave  <NA>       IR1          Lvl    AllPub     Corner        Gtl
## 2   Pave  <NA>       Reg          Lvl    AllPub     Inside        Gtl
## 3   Pave  <NA>       IR1          Lvl    AllPub     Corner        Gtl
## 4   Pave  <NA>       Reg          Lvl    AllPub     Corner        Gtl
## 5   Pave  <NA>       IR1          Lvl    AllPub     Inside        Gtl
## 6   Pave  <NA>       IR1          Lvl    AllPub     Inside        Gtl
##   Neighborhood Condition.1 Condition.2 Bldg.Type House.Style Overall.Qual
## 1        NAmes        Norm        Norm      1Fam      1Story            6
## 2        NAmes       Feedr        Norm      1Fam      1Story            5
## 3        NAmes        Norm        Norm      1Fam      1Story            6
## 4        NAmes        Norm        Norm      1Fam      1Story            7
## 5      Gilbert        Norm        Norm      1Fam      2Story            5
## 6      Gilbert        Norm        Norm      1Fam      2Story            6
##   Overall.Cond Year.Built Year.Remod.Add Roof.Style Roof.Matl Exterior.1st
## 1            5       1960           1960        Hip   CompShg      BrkFace
## 2            6       1961           1961      Gable   CompShg      VinylSd
## 3            6       1958           1958        Hip   CompShg      Wd Sdng
## 4            5       1968           1968        Hip   CompShg      BrkFace
## 5            5       1997           1998      Gable   CompShg      VinylSd
## 6            6       1998           1998      Gable   CompShg      VinylSd
##   Exterior.2nd Mas.Vnr.Type Mas.Vnr.Area Exter.Qual Exter.Cond Foundation
## 1      Plywood        Stone          112         TA         TA     CBlock
## 2      VinylSd         None            0         TA         TA     CBlock
## 3      Wd Sdng      BrkFace          108         TA         TA     CBlock
## 4      BrkFace         None            0         Gd         TA     CBlock
## 5      VinylSd         None            0         TA         TA      PConc
## 6      VinylSd      BrkFace           20         TA         TA      PConc
##   Bsmt.Qual Bsmt.Cond Bsmt.Exposure BsmtFin.Type.1 BsmtFin.SF.1 BsmtFin.Type.2
## 1        TA        Gd            Gd            BLQ          639            Unf
## 2        TA        TA            No            Rec          468            LwQ
## 3        TA        TA            No            ALQ          923            Unf
## 4        TA        TA            No            ALQ         1065            Unf
## 5        Gd        TA            No            GLQ          791            Unf
## 6        TA        TA            No            GLQ          602            Unf
##   BsmtFin.SF.2 Bsmt.Unf.SF Total.Bsmt.SF Heating Heating.QC Central.Air
## 1            0         441          1080    GasA         Fa           Y
## 2          144         270           882    GasA         TA           Y
## 3            0         406          1329    GasA         TA           Y
## 4            0        1045          2110    GasA         Ex           Y
## 5            0         137           928    GasA         Gd           Y
## 6            0         324           926    GasA         Ex           Y
##   Electrical X1st.Flr.SF X2nd.Flr.SF Low.Qual.Fin.SF Bsmt.Full.Bath
## 1      SBrkr        1656           0               0              1
## 2      SBrkr         896           0               0              0
```

```
## 3      SBrkr        1329          0               0            0
## 4      SBrkr        2110          0               0            1
## 5      SBrkr         928        701               0            0
## 6      SBrkr         926        678               0            0
##   Bsmt.Half.Bath Full.Bath Half.Bath Bedroom.AbvGr Kitchen.AbvGr Kitchen.Qual
## 1              0         1         0             3             1           TA
## 2              0         1         0             2             1           TA
## 3              0         1         1             3             1           Gd
## 4              0         2         1             3             1           Ex
## 5              0         2         1             3             1           TA
## 6              0         2         1             3             1           Gd
##   TotRms.AbvGrd Functional Fireplaces Fireplace.Qu Garage.Type Garage.Yr.Blt
## 1             7        Typ          2           Gd      Attchd          1960
## 2             5        Typ          0         <NA>      Attchd          1961
## 3             6        Typ          0         <NA>      Attchd          1958
## 4             8        Typ          2           TA      Attchd          1968
## 5             6        Typ          1           TA      Attchd          1997
## 6             7        Typ          1           Gd      Attchd          1998
##   Garage.Finish Garage.Cars Garage.Area Garage.Qual Garage.Cond Paved.Drive
## 1           Fin           2         528          TA          TA           P
## 2           Unf           1         730          TA          TA           Y
## 3           Unf           1         312          TA          TA           Y
## 4           Fin           2         522          TA          TA           Y
## 5           Fin           2         482          TA          TA           Y
## 6           Fin           2         470          TA          TA           Y
##   Wood.Deck.SF Open.Porch.SF Enclosed.Porch X3Ssn.Porch Screen.Porch Pool.Area
## 1          210            62              0           0            0         0
## 2          140             0              0           0          120         0
## 3          393            36              0           0            0         0
## 4            0             0              0           0            0         0
## 5          212            34              0           0            0         0
## 6          360            36              0           0            0         0
##   Pool.QC Fence Misc.Feature Misc.Val Mo.Sold Yr.Sold Sale.Type Sale.Condition
## 1    <NA>  <NA>         <NA>        0       5    2010       WD          Normal
## 2    <NA> MnPrv         <NA>        0       6    2010       WD          Normal
## 3    <NA>  <NA>         Gar2    12500       6    2010       WD          Normal
## 4    <NA>  <NA>         <NA>        0       4    2010       WD          Normal
## 5    <NA> MnPrv         <NA>        0       3    2010       WD          Normal
## 6    <NA>  <NA>         <NA>        0       6    2010       WD          Normal
```

Remove columns with missing values

```
# Remove columns with missing values
houses_df = houses_df[ , colSums(is.na(houses_df)) == 0]

# Remove Pool.Area since all its values are "0"
houses_df = houses_df[ , !(names(houses_df) %in% c("Pool.Area"))]

sprintf("The dataset has %d rows", nrow(houses_df))
```

```
## [1] "The dataset has 1000 rows"
```

```
sprintf("The dataset has %d columns", ncol(houses_df))
```

```
## [1] "The dataset has 64 columns"
```

Convert char columns to factor

```
houses_df[sapply(houses_df, is.character)] <- lapply(houses_df[sapply(houses_df, is.character)], as.factor)
```

Creating a full model and applying BIC and AIC technique and comparing

```
index <- sample(seq_len(nrow(houses_df)), size = 0.8 * nrow(houses_df))
X_train <- houses_df[index, ]
X_test <- houses_df[-index, ]

full_model <- lm(price ~ . -PID -price, data = X_train)
```

BIC elimination

```
library(MASS)
full_model.step.bic <-
  stepAIC(full_model, direction = "backward", k=log(1000), trace = 0)
model.bic <- eval(full_model.step.bic$call)
summary(model.bic)
```

```
##
## Call:
## lm(formula = price ~ area + MS.SubClass + Lot.Area + Land.Slope +
##     Overall.Qual + Overall.Cond + Year.Built + Mas.Vnr.Type +
##     Exter.Qual + BsmtFin.SF.1 + BsmtFin.SF.2 + Bsmt.Unf.SF +
##     X2nd.Flr.SF + Bedroom.AbvGr + Kitchen.AbvGr + Kitchen.Qual +
##     Garage.Cars + Sale.Condition, data = X_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -100908  -10631      90   11507  151971
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -7.998e+05  8.628e+04  -9.270  < 2e-16 ***
## area                   5.515e+01  4.319e+00  12.769  < 2e-16 ***
## MS.SubClass           -9.860e+01  2.340e+01  -4.214 2.81e-05 ***
## Lot.Area               1.173e+00  1.239e-01   9.466  < 2e-16 ***
## Land.SlopeMod          4.718e+03  4.076e+03   1.158  0.24738
## Land.SlopeSev         -8.095e+04  1.445e+04  -5.600 2.98e-08 ***
## Overall.Qual           8.306e+03  1.068e+03   7.775 2.41e-14 ***
## Overall.Cond           4.134e+03  8.285e+02   4.989 7.51e-07 ***
## Year.Built             4.330e+02  4.295e+01  10.081  < 2e-16 ***
## Mas.Vnr.TypeBrkCmn    -5.509e+03  1.535e+04  -0.359  0.71979
## Mas.Vnr.TypeBrkFace   -1.734e+03  1.021e+04  -0.170  0.86521
## Mas.Vnr.TypeNone       4.397e+02  1.018e+04   0.043  0.96556
## Mas.Vnr.TypeStone      1.517e+04  1.038e+04   1.461  0.14435
## Exter.QualFa          -4.286e+04  9.316e+03  -4.601 4.92e-06 ***
## Exter.QualGd          -3.721e+04  5.620e+03  -6.620 6.72e-11 ***
## Exter.QualTA          -4.433e+04  6.297e+03  -7.040 4.26e-12 ***
## BsmtFin.SF.1           4.488e+01  3.681e+00  12.192  < 2e-16 ***
## BsmtFin.SF.2           3.231e+01  5.982e+00   5.401 8.82e-08 ***
## Bsmt.Unf.SF            2.015e+01  3.652e+00   5.518 4.68e-08 ***
## X2nd.Flr.SF            1.136e+01  4.264e+00   2.664  0.00788 **
## Bedroom.AbvGr         -5.755e+03  1.439e+03  -4.001 6.93e-05 ***
## Kitchen.AbvGr         -1.143e+04  4.151e+03  -2.753  0.00605 **
## Kitchen.QualFa        -3.673e+04  7.343e+03  -5.001 7.05e-07 ***
## Kitchen.QualGd        -3.674e+04  4.394e+03  -8.362 2.88e-16 ***
## Kitchen.QualTA        -4.209e+04  4.984e+03  -8.444  < 2e-16 ***
## Garage.Cars            7.188e+03  1.495e+03   4.809 1.82e-06 ***
## Sale.ConditionAlloca   2.271e+04  8.645e+03   2.627  0.00879 **
## Sale.ConditionFamily   1.282e+04  7.562e+03   1.695  0.09045 .
## Sale.ConditionNormal   1.730e+04  3.404e+03   5.082 4.69e-07 ***
## Sale.ConditionPartial  3.581e+04  5.534e+03   6.471 1.73e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22240 on 770 degrees of freedom
## Multiple R-squared:  0.9155, Adjusted R-squared:  0.9123
## F-statistic: 287.8 on 29 and 770 DF,  p-value: < 2.2e-16
```

AIC elimination

```
full_model.step.aic <-
  stepAIC(full_model, direction = "backward", k=7, trace = 0)
model.aic <- eval(full_model.step.aic$call)
summary(model.aic)
```

```
##
## Call:
## lm(formula = price ~ area + MS.SubClass + Lot.Area + Land.Slope +
##     Overall.Qual + Overall.Cond + Year.Built + Mas.Vnr.Type +
##     Exter.Qual + BsmtFin.SF.1 + BsmtFin.SF.2 + Bsmt.Unf.SF +
##     X2nd.Flr.SF + Bedroom.AbvGr + Kitchen.AbvGr + Kitchen.Qual +
##     Garage.Cars + Sale.Condition, data = X_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -100908  -10631      90   11507  151971
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -7.998e+05  8.628e+04  -9.270  < 2e-16 ***
## area                   5.515e+01  4.319e+00  12.769  < 2e-16 ***
## MS.SubClass           -9.860e+01  2.340e+01  -4.214 2.81e-05 ***
## Lot.Area               1.173e+00  1.239e-01   9.466  < 2e-16 ***
## Land.SlopeMod          4.718e+03  4.076e+03   1.158  0.24738
## Land.SlopeSev         -8.095e+04  1.445e+04  -5.600 2.98e-08 ***
## Overall.Qual           8.306e+03  1.068e+03   7.775 2.41e-14 ***
## Overall.Cond           4.134e+03  8.285e+02   4.989 7.51e-07 ***
## Year.Built             4.330e+02  4.295e+01  10.081  < 2e-16 ***
## Mas.Vnr.TypeBrkCmn    -5.509e+03  1.535e+04  -0.359  0.71979
## Mas.Vnr.TypeBrkFace   -1.734e+03  1.021e+04  -0.170  0.86521
## Mas.Vnr.TypeNone       4.397e+02  1.018e+04   0.043  0.96556
## Mas.Vnr.TypeStone      1.517e+04  1.038e+04   1.461  0.14435
## Exter.QualFa          -4.286e+04  9.316e+03  -4.601 4.92e-06 ***
## Exter.QualGd          -3.721e+04  5.620e+03  -6.620 6.72e-11 ***
## Exter.QualTA          -4.433e+04  6.297e+03  -7.040 4.26e-12 ***
## BsmtFin.SF.1           4.488e+01  3.681e+00  12.192  < 2e-16 ***
## BsmtFin.SF.2           3.231e+01  5.982e+00   5.401 8.82e-08 ***
## Bsmt.Unf.SF            2.015e+01  3.652e+00   5.518 4.68e-08 ***
## X2nd.Flr.SF            1.136e+01  4.264e+00   2.664  0.00788 **
## Bedroom.AbvGr         -5.755e+03  1.439e+03  -4.001 6.93e-05 ***
## Kitchen.AbvGr         -1.143e+04  4.151e+03  -2.753  0.00605 **
## Kitchen.QualFa        -3.673e+04  7.343e+03  -5.001 7.05e-07 ***
## Kitchen.QualGd        -3.674e+04  4.394e+03  -8.362 2.88e-16 ***
## Kitchen.QualTA        -4.209e+04  4.984e+03  -8.444  < 2e-16 ***
## Garage.Cars            7.188e+03  1.495e+03   4.809 1.82e-06 ***
## Sale.ConditionAlloca   2.271e+04  8.645e+03   2.627  0.00879 **
## Sale.ConditionFamily   1.282e+04  7.562e+03   1.695  0.09045 .
## Sale.ConditionNormal   1.730e+04  3.404e+03   5.082 4.69e-07 ***
## Sale.ConditionPartial  3.581e+04  5.534e+03   6.471 1.73e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22240 on 770 degrees of freedom
## Multiple R-squared:  0.9155, Adjusted R-squared:  0.9123
## F-statistic: 287.8 on 29 and 770 DF,  p-value: < 2.2e-16
```

Predictor selection using lasso

Lasso regression is a regularization technique that can perform predictor selection. In other words, it can set the regression coefficients to zero.

We note that Lasso has demonstrated a superior ability to identify more relevant predictors compared to AIC and BIC. As a result, we will be adopting the predictors identified by Lasso for future analyses.

```
# model.matrix() returns the design matrix X
# remove the bias column (all 1s)
X <- model.matrix(price ~ ., houses_df)[, -1]
y <- houses_df$price
new_houses_df = data.frame(X, y)
names(new_houses_df)[names(new_houses_df) == "y"] <- "price"
```

```
# alpha = 1 is lasso regression
fit_lasso <- glmnet(X, y, alpha = 1)
```

To make our analysis more manageable, we will use a high λ value to identify the top 10-15 predictors. We will not use any of the other predictors for our models.

```
fit_lasso = glmnet(X, y, alpha = 1, lambda = 9000)

# [-c(1)] removes "intercept"
selected_predictors = rownames(coef(fit_lasso, s = 'lambda.min'))[coef(fit_lasso, s = 'lambda.mi
n')[,1]!= 0][-c(1)]
cat(selected_predictors, sep="   ")
```

```
## area   Lot.Area   NeighborhoodNridgHt   Overall.Qual   Year.Built   Year.Remod.Add   Mas.Vnr.
TypeStone   Exter.QualTA   BsmtFin.SF.1   Total.Bsmt.SF   X1st.Flr.SF   Garage.Cars   Garage.Are
a
```

area: above ground floor area
Lot.Area: area of the land that comes with the house
NeighborhoodNridgHt: whether the house is in the Northridge Heights neighborhood (yes/no)
Overall.Qual: construction quality of the house (1 - 10)
Year.Built: year the house was built
Year.Remod.Add: year the house was remodeled (same as Year.Built if house was never remodeled)
Mas.Vnr.TypeStone: whether the house has a stone masonry veneer (yes/no)
Exter.QualTA: whether the construction quality of the exterior of the house is average (yes/no)
BsmtFin.SF.1: area of finished parts of basement
TotalBsmt.SF: total area of basement
X1st.Flr.SF: area of first floor
Garage.Cars: how many cars can fit in the garage
Garage.Area: area of the garage

```
# Paste the selected predictors into a formula string
right_hand_side = paste(selected_predictors, collapse=" + ")
formula_string = paste("price ~", right_hand_side, collapse = "")
linear = lm(formula_string, data=new_houses_df)
```

```
summary(linear)
```

```
##
## Call:
## lm(formula = formula_string, data = new_houses_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -120264  -14007     -89   14357  206453
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -1.027e+06  1.202e+05  -8.542  < 2e-16 ***
## area                 5.216e+01  2.625e+00  19.868  < 2e-16 ***
## Lot.Area             9.587e-01  1.194e-01   8.031 2.75e-15 ***
## NeighborhoodNridgHt  3.126e+04  4.453e+03   7.019 4.15e-12 ***
## Overall.Qual         1.336e+04  1.120e+03  11.925  < 2e-16 ***
## Year.Built           2.193e+02  4.360e+01   5.031 5.80e-07 ***
## Year.Remod.Add       2.780e+02  5.942e+01   4.679 3.29e-06 ***
## Mas.Vnr.TypeStone    1.707e+04  3.441e+03   4.962 8.20e-07 ***
## Exter.QualTA        -1.023e+04  2.480e+03  -4.123 4.06e-05 ***
## BsmtFin.SF.1         2.899e+01  2.374e+00  12.211  < 2e-16 ***
## Total.Bsmt.SF        2.213e+01  3.870e+00   5.718 1.42e-08 ***
## X1st.Flr.SF          2.439e+00  4.413e+00   0.553   0.5806
## Garage.Cars          3.463e+03  2.759e+03   1.255   0.2097
## Garage.Area          2.188e+01  9.551e+00   2.291   0.0222 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 28260 on 986 degrees of freedom
## Multiple R-squared:  0.8721, Adjusted R-squared:  0.8704
## F-statistic: 517.1 on 13 and 986 DF,  p-value: < 2.2e-16
```

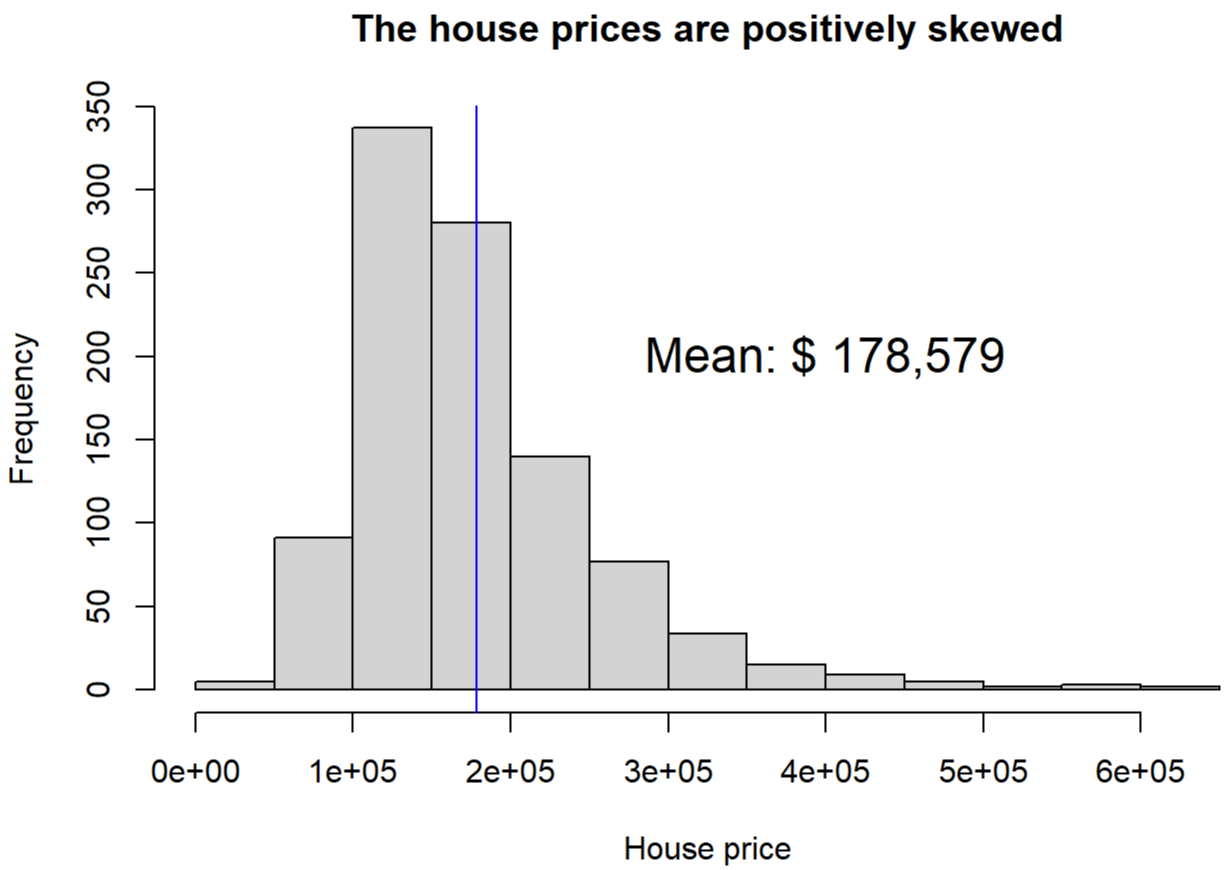
The p-value for each predictor corresponds to the hypothesis test:
$H_0 : \beta_j = 0$
$H_a : \beta_j \neq 0$

Predictors with small p-values smaller than 0.05 have a significant linear relationship with the target (house price), given that the other predictors are used in the model.

Exploratory data analysis

Histogram of target variable

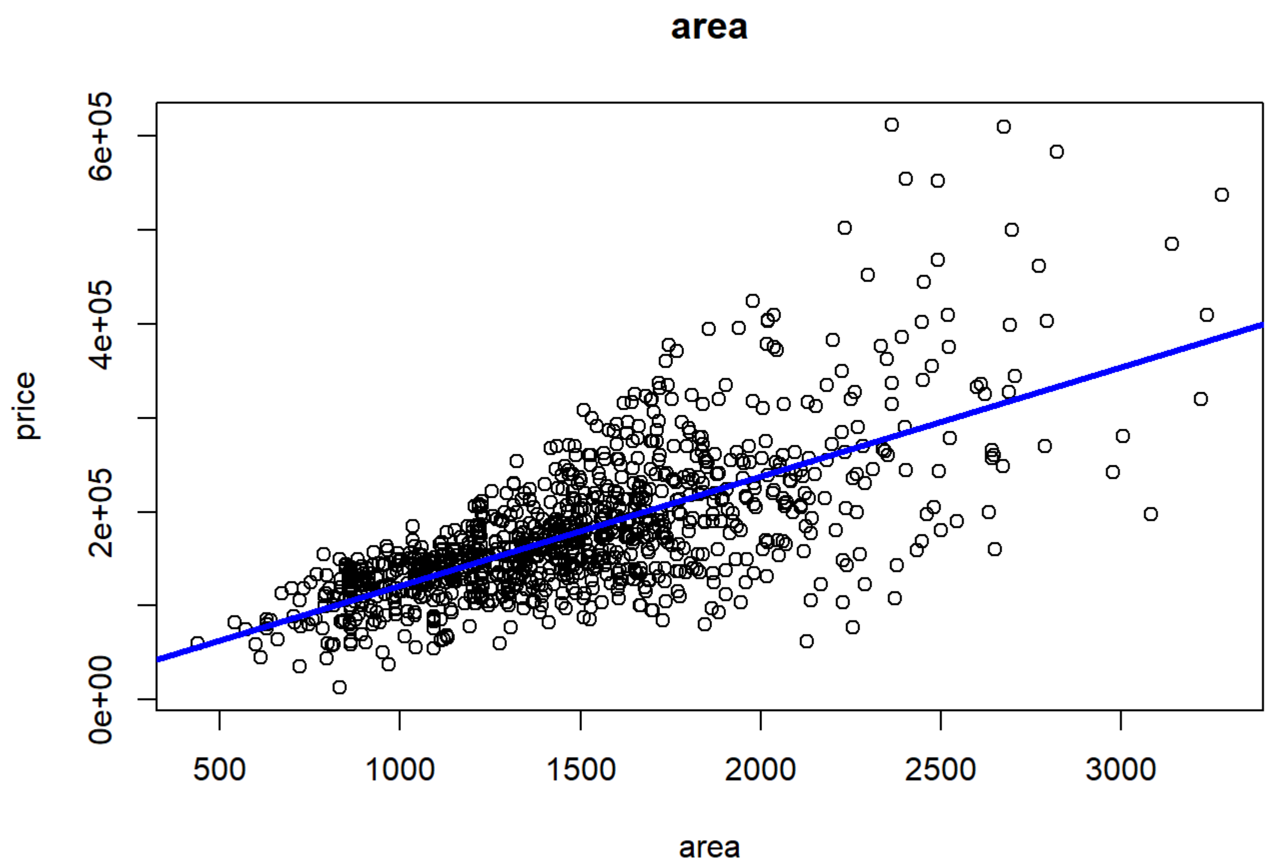
```
hist(houses_df$price,
     main="The house prices are positively skewed",
     xlab="House price")
abline(v=mean(houses_df$price),col="blue")
mean_price = format(round(mean(houses_df$price), 0), nsmall=0, big.mark=",")
text(4e+05, 200, paste("Mean: $", mean_price), cex=1.5)
```

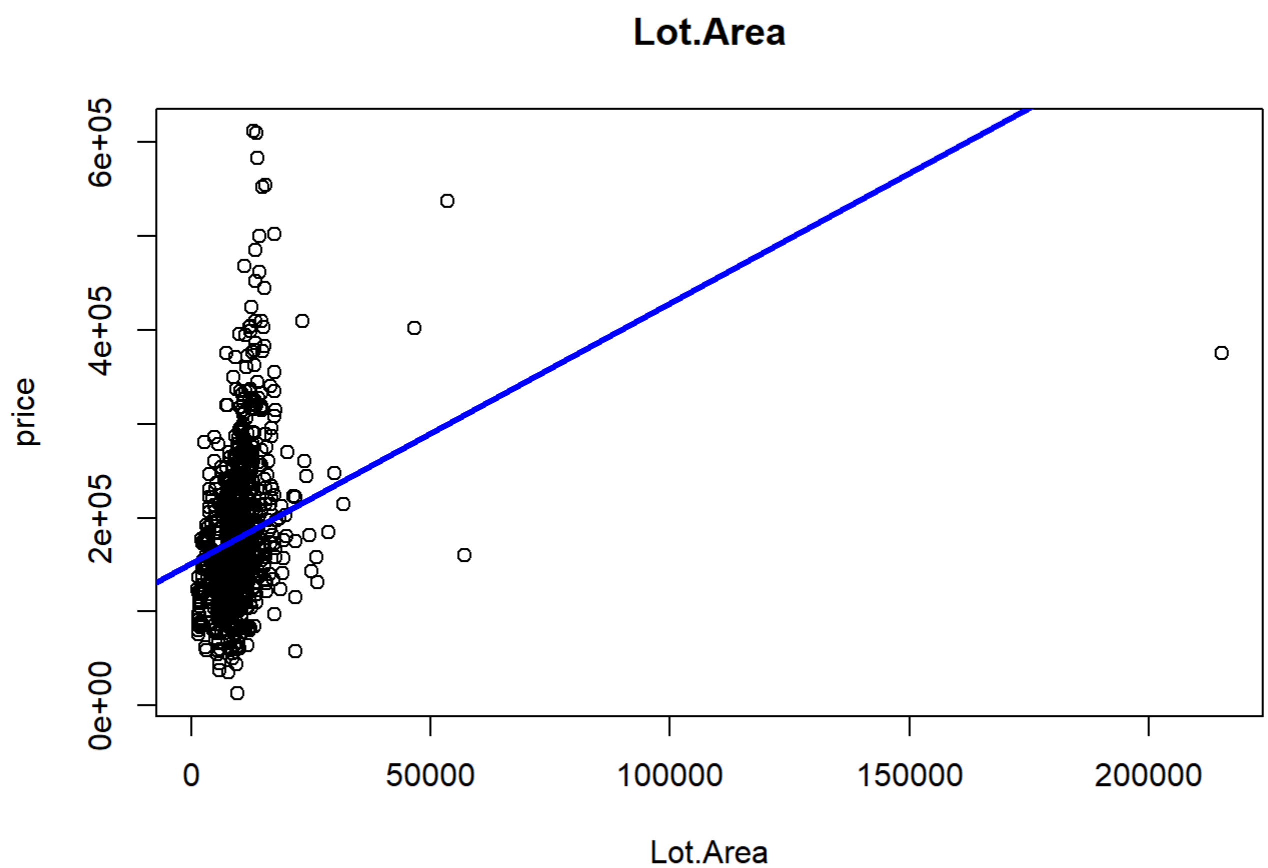


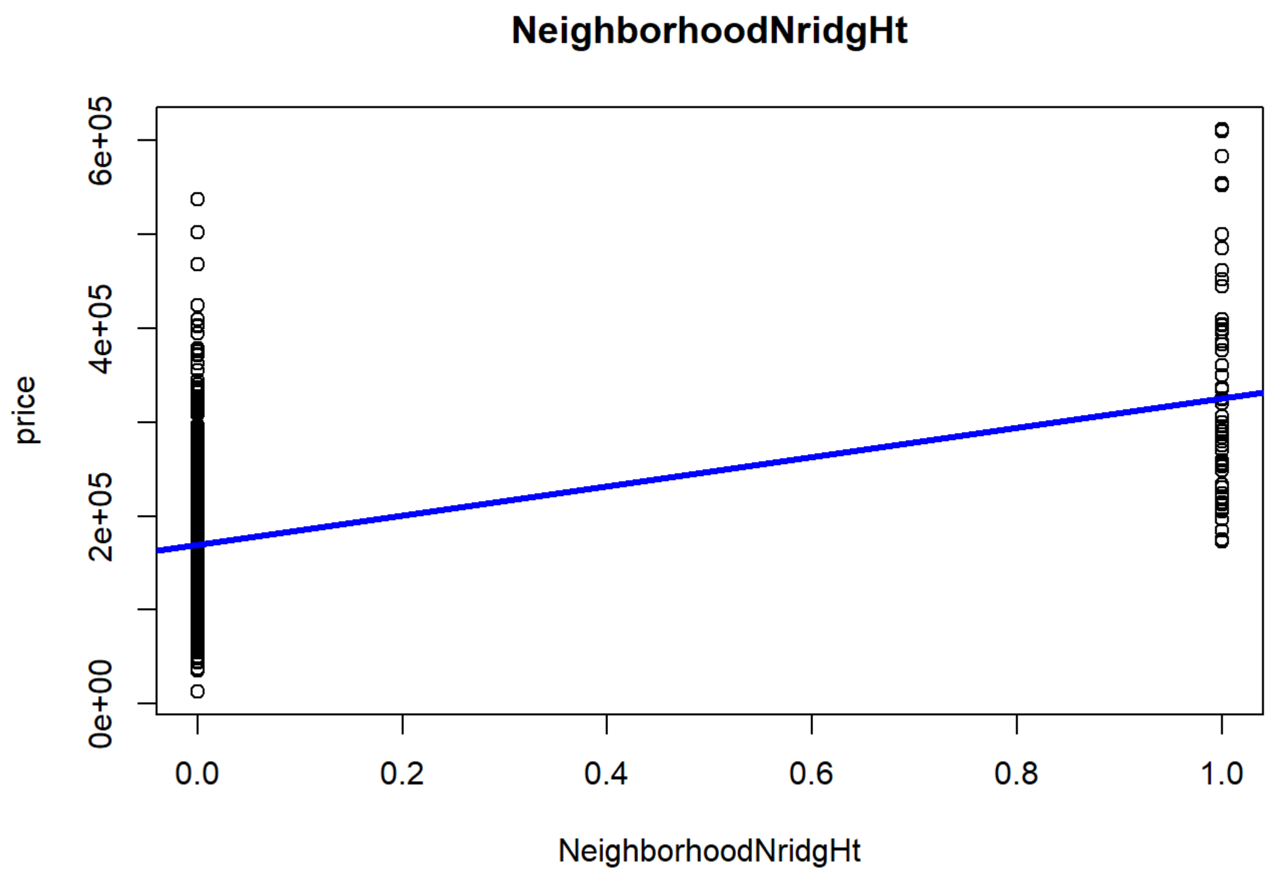
The target (house price) is positively skewed because there are a few houses that are abnormally expensive.

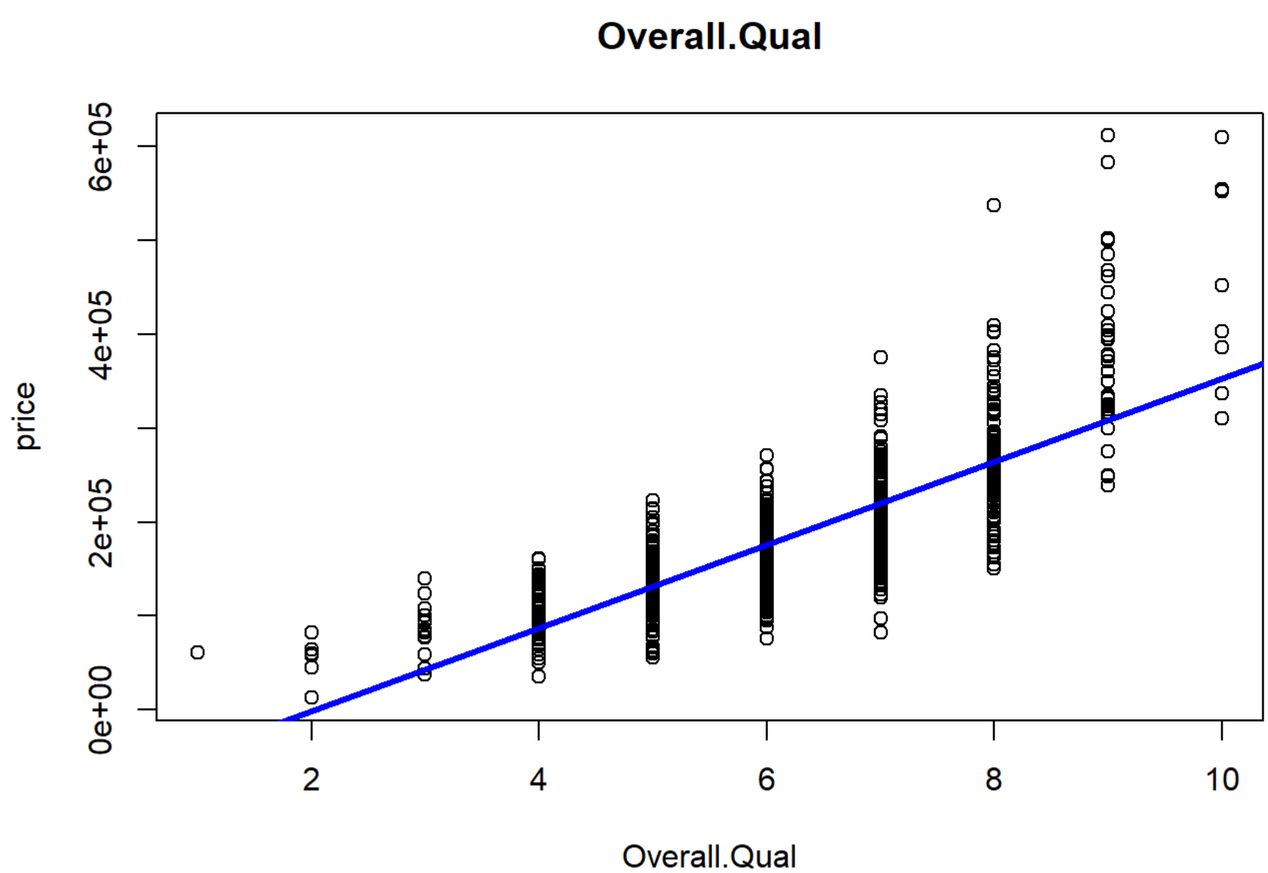
Scatterplots of each predictor with house price

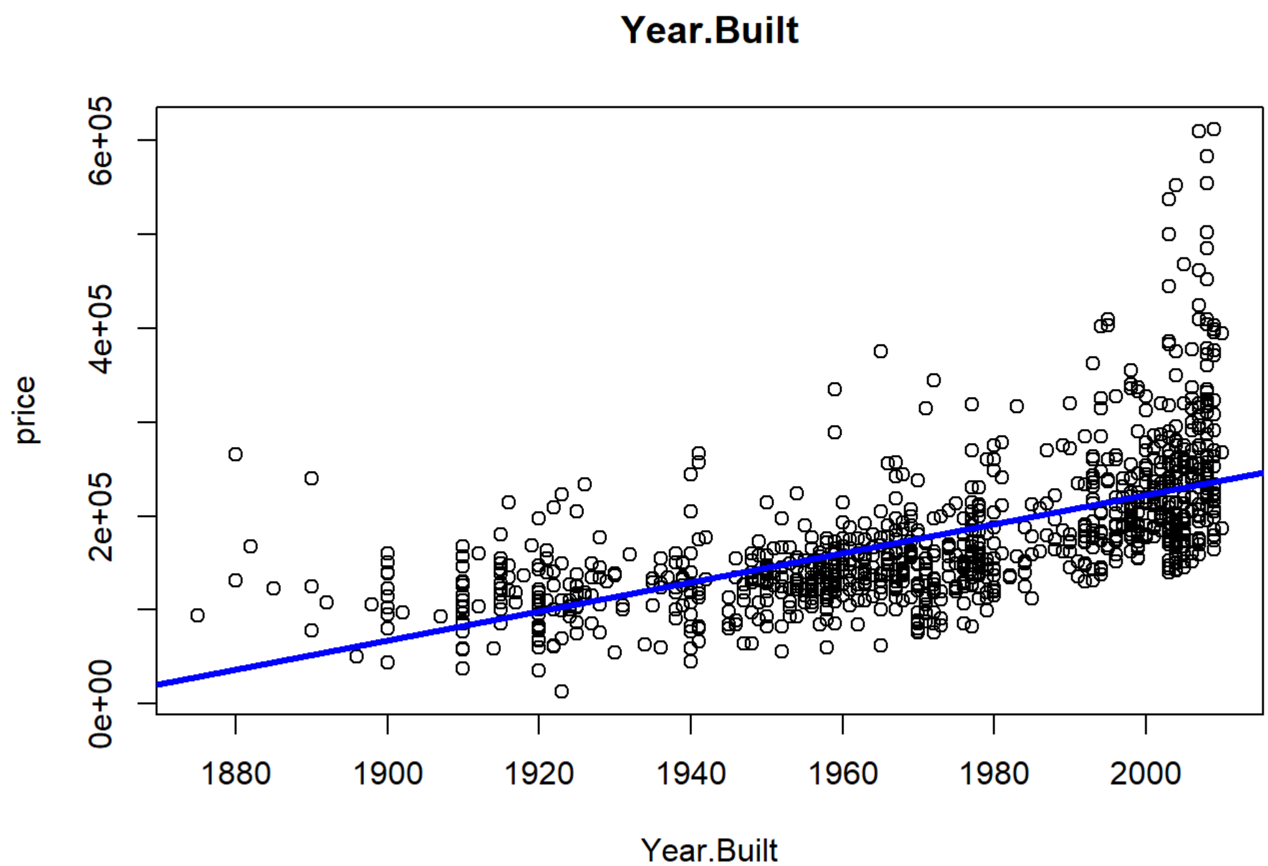
```
for (predictor in selected_predictors) {
  plot(price ~ eval(parse(text = predictor)),
     data=new_houses_df,
     xlab=predictor,
     main=predictor)

  predictor_linear = lm(price ~ eval(parse(text = predictor)), data=new_houses_df)
  abline(predictor_linear, lwd = 3, lty = 1, col = "blue")
}
```

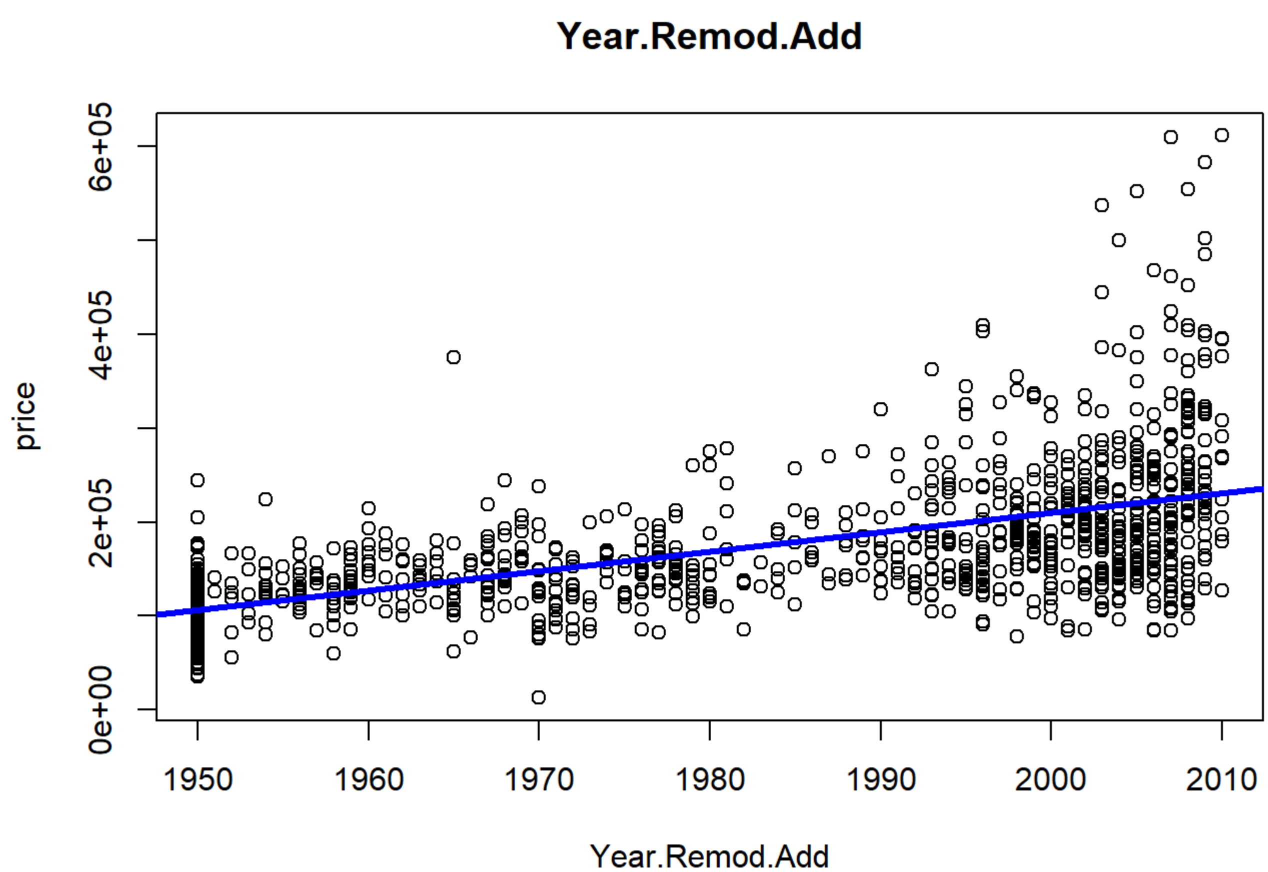

area
price
0e+00
2e+05
4e+05
6e+05
500
1000
1500
2000
2500
3000
area

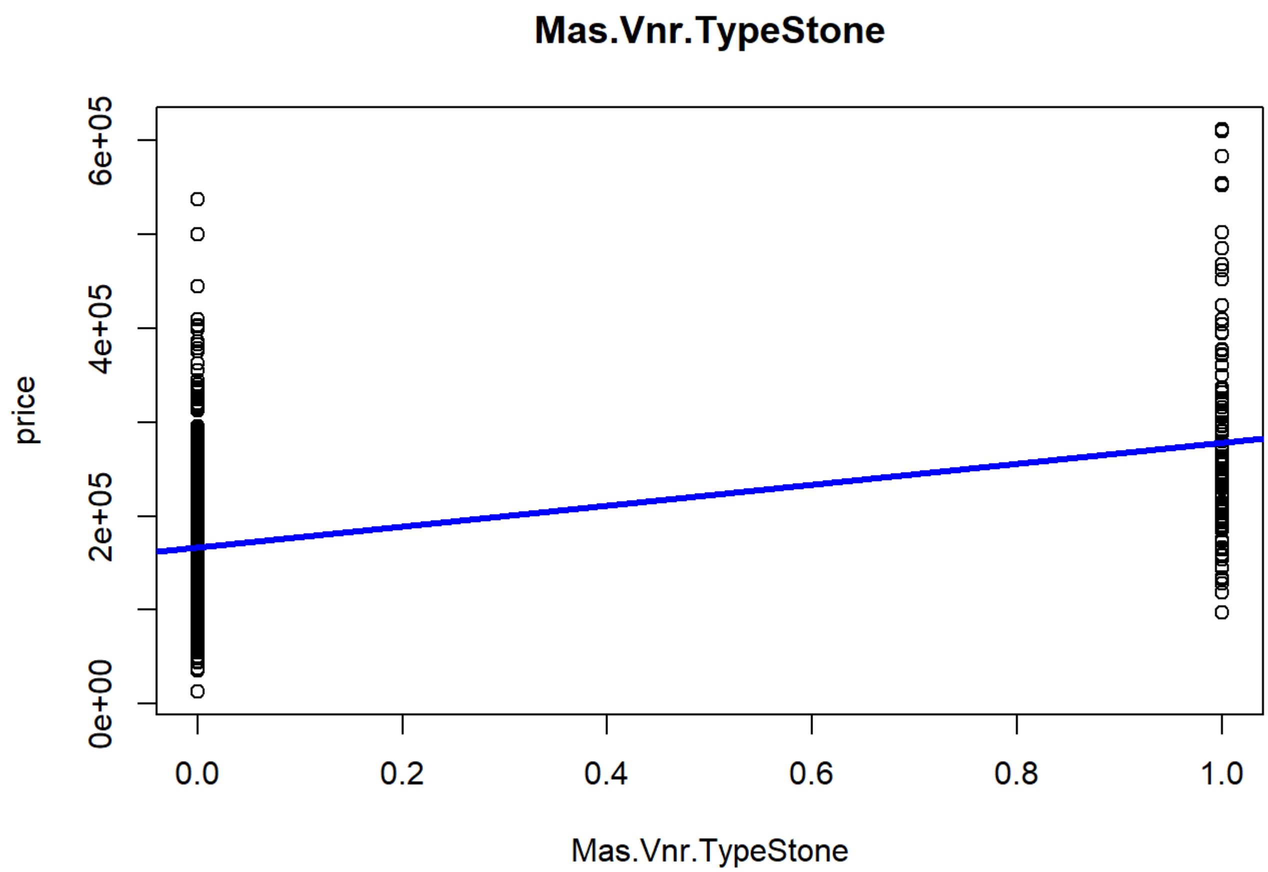

Lot.Area
price
0e+00
2e+05
4e+05
6e+05
0
50000
100000
150000
200000
Lot.Area

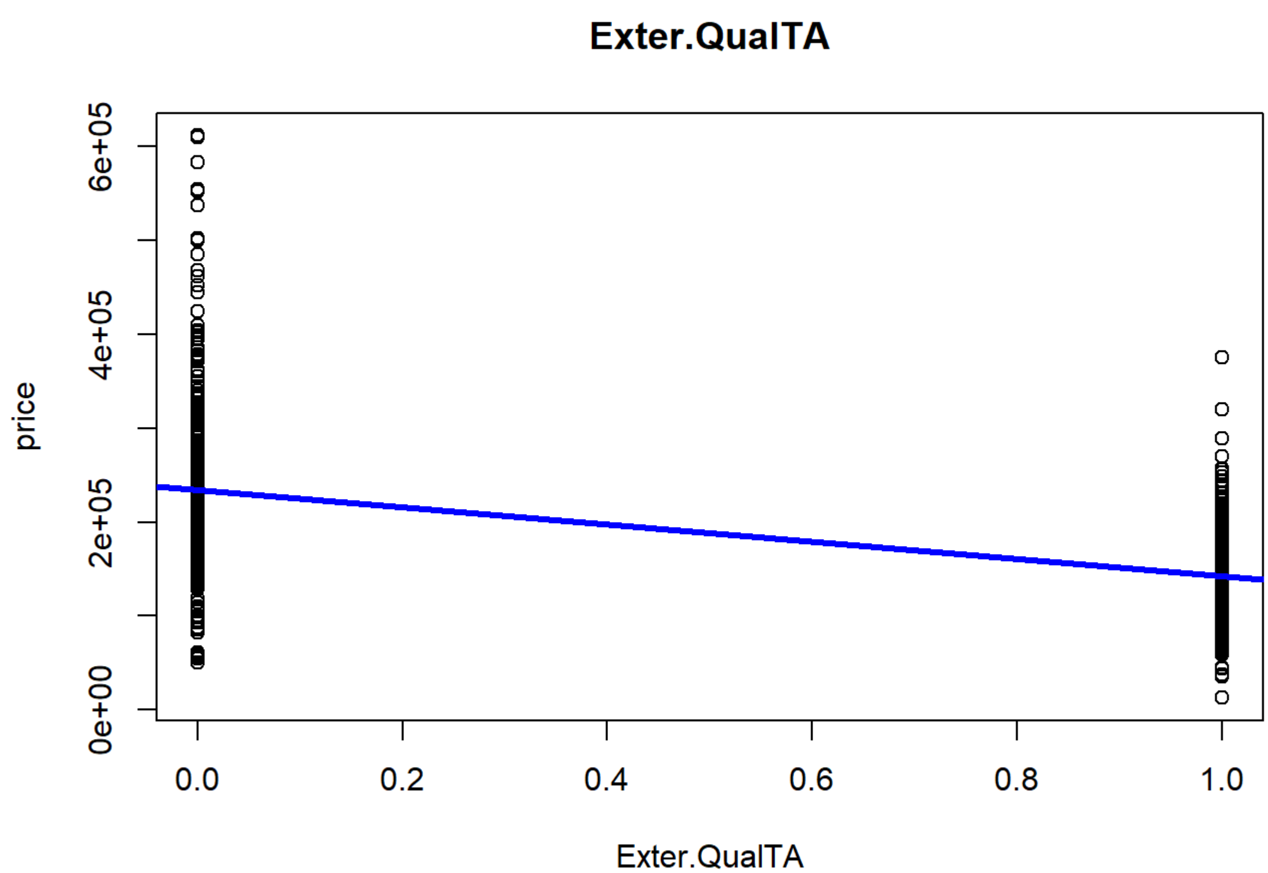

NeighborhoodNridgHt
price
0e+00
2e+05
4e+05
6e+05
0.0
0.2
0.4
0.6
0.8
1.0
NeighborhoodNridgHt

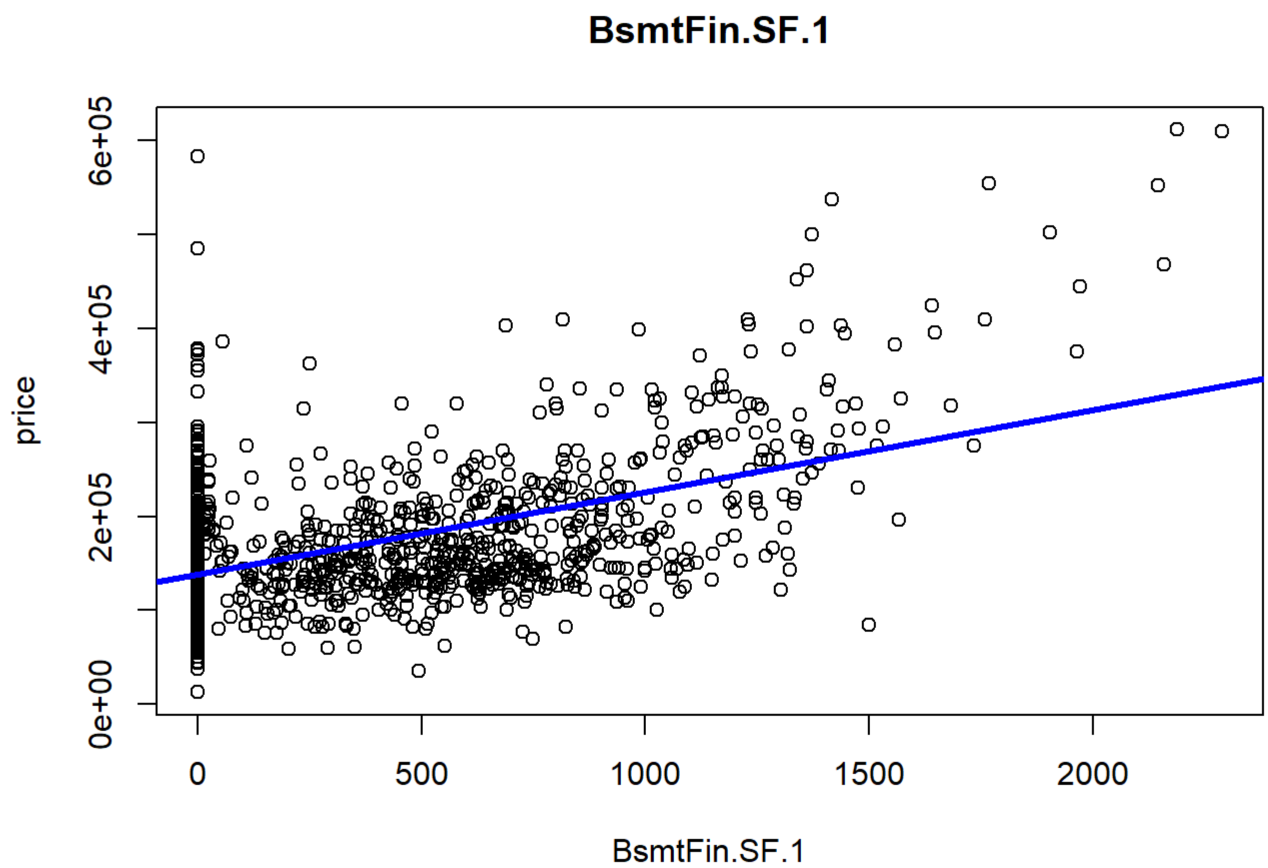

Overall.Qual
price
0e+00
2e+05
4e+05
6e+05
2
4
6
8
10
Overall.Qual

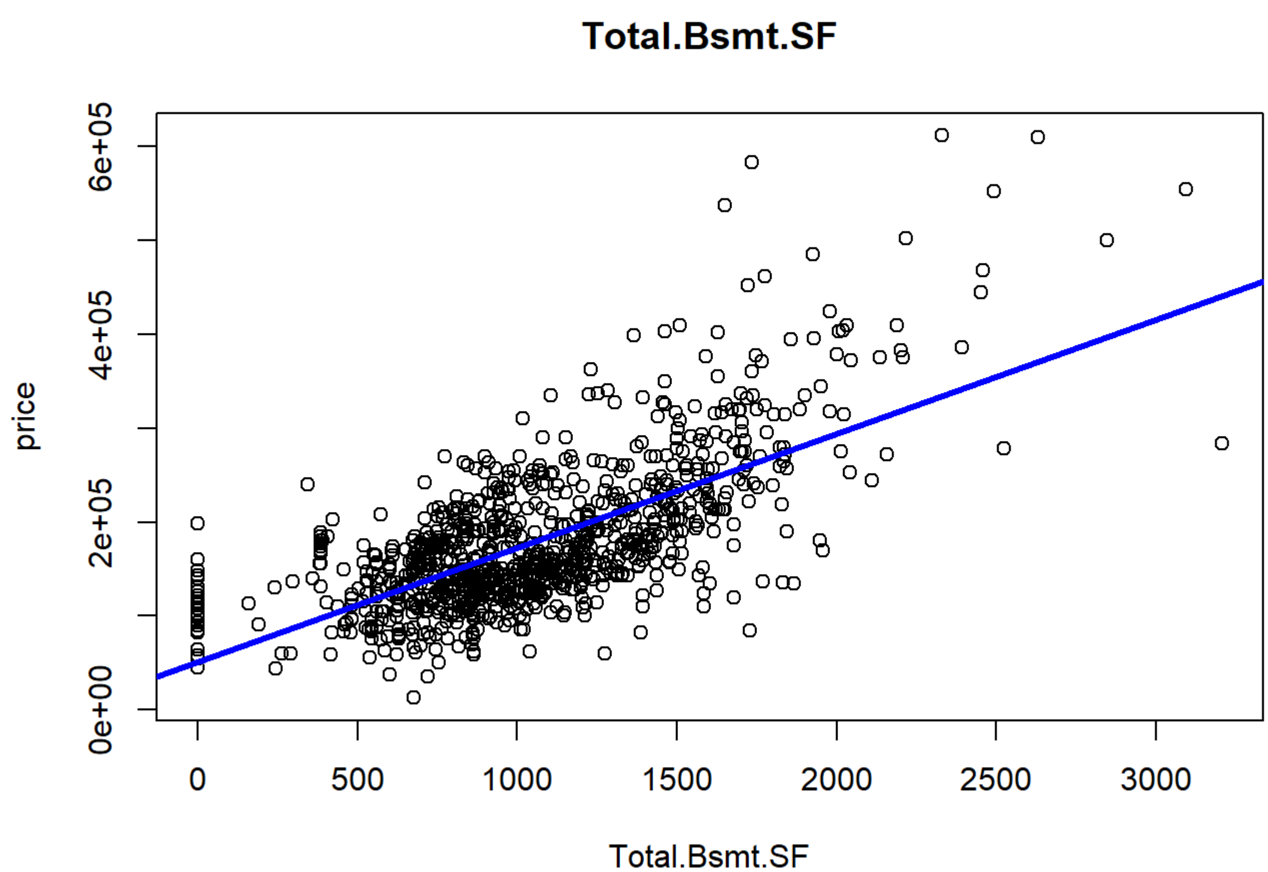

Year.Built
price
0e+00
2e+05
4e+05
6e+05
1880
1900
1920
1940
1960
1980
2000
Year.Built

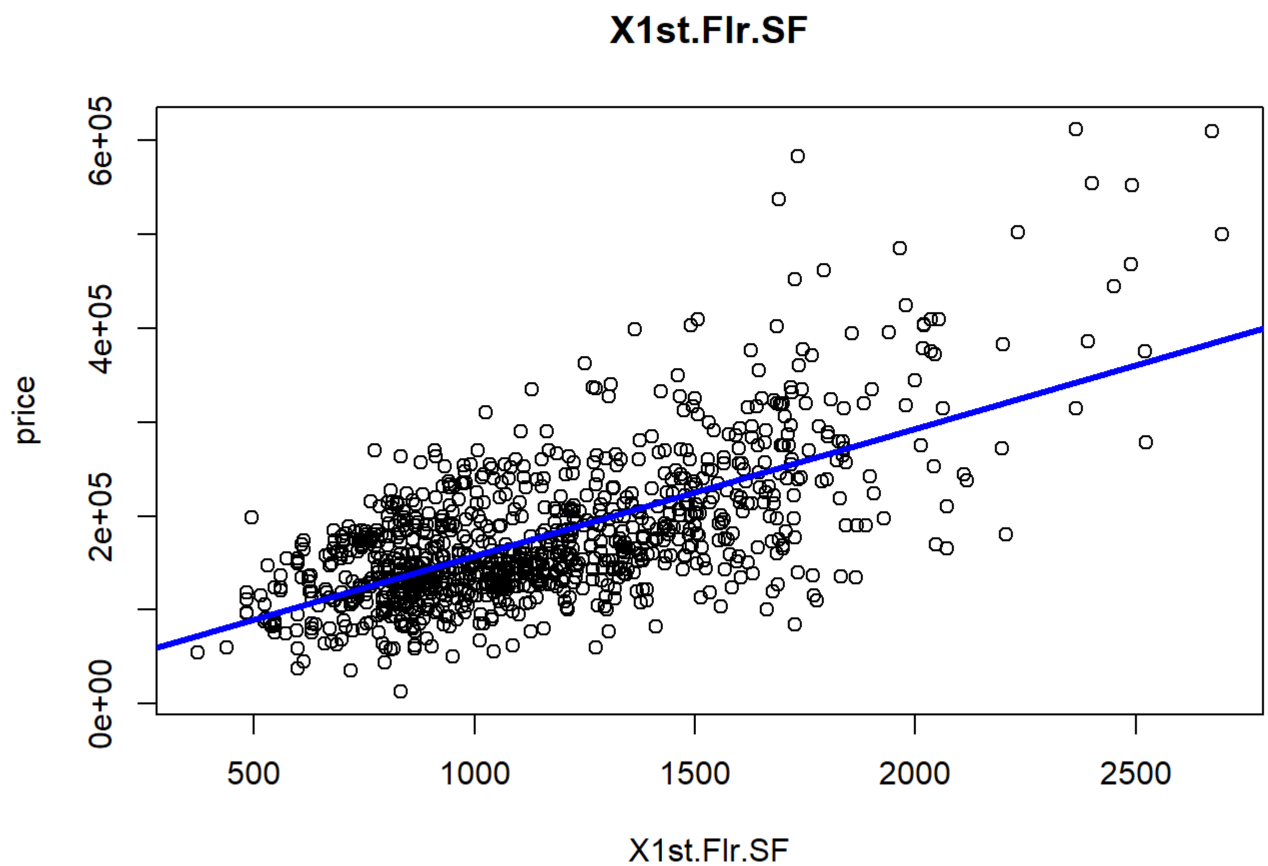

Year.Remod.Add
price
0e+00
2e+05
4e+05
6e+05
1950
1960
1970
1980
1990
2000
2010
Year.Remod.Add

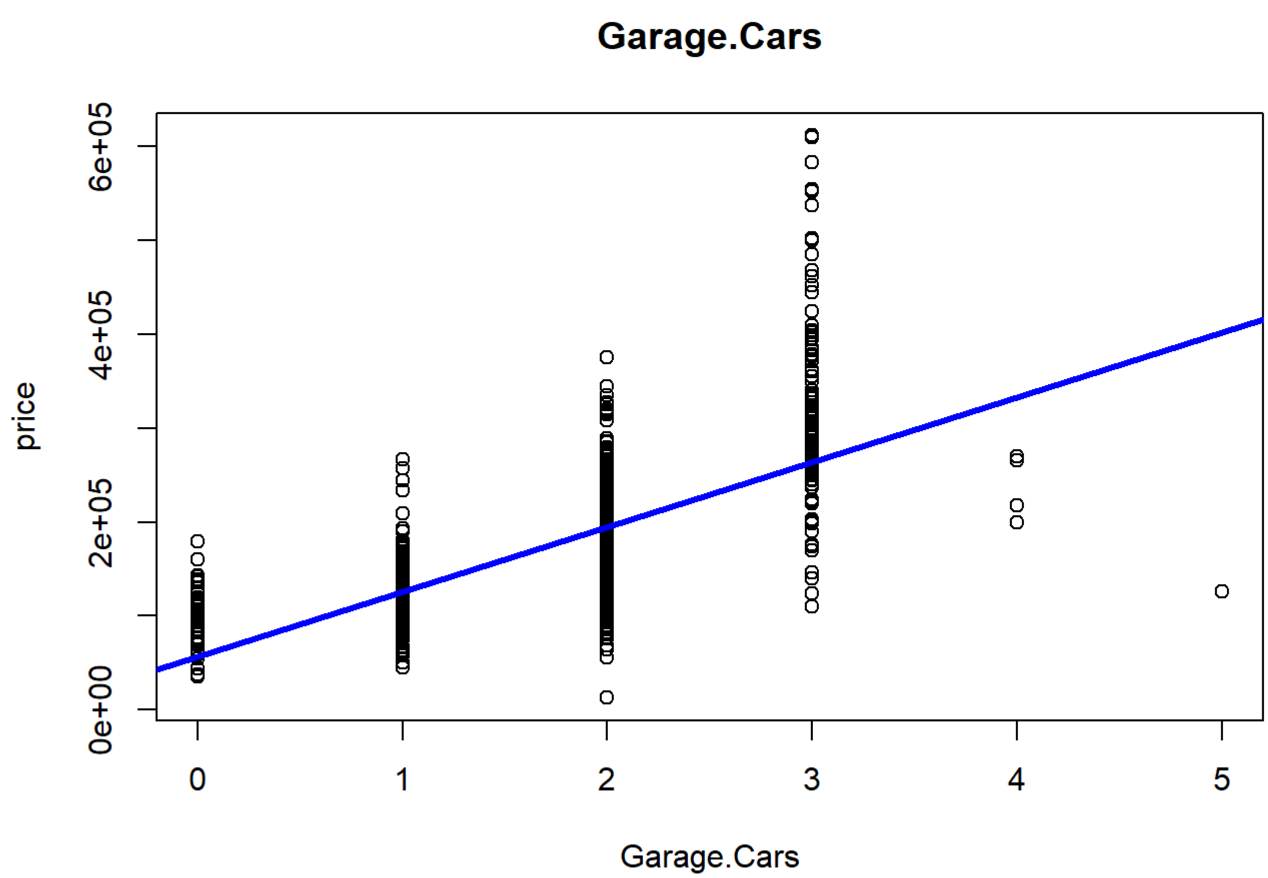

Mas.Vnr.TypeStone
price
0e+00
2e+05
4e+05
6e+05
0.0
0.2
0.4
0.6
0.8
1.0
Mas.Vnr.TypeStone

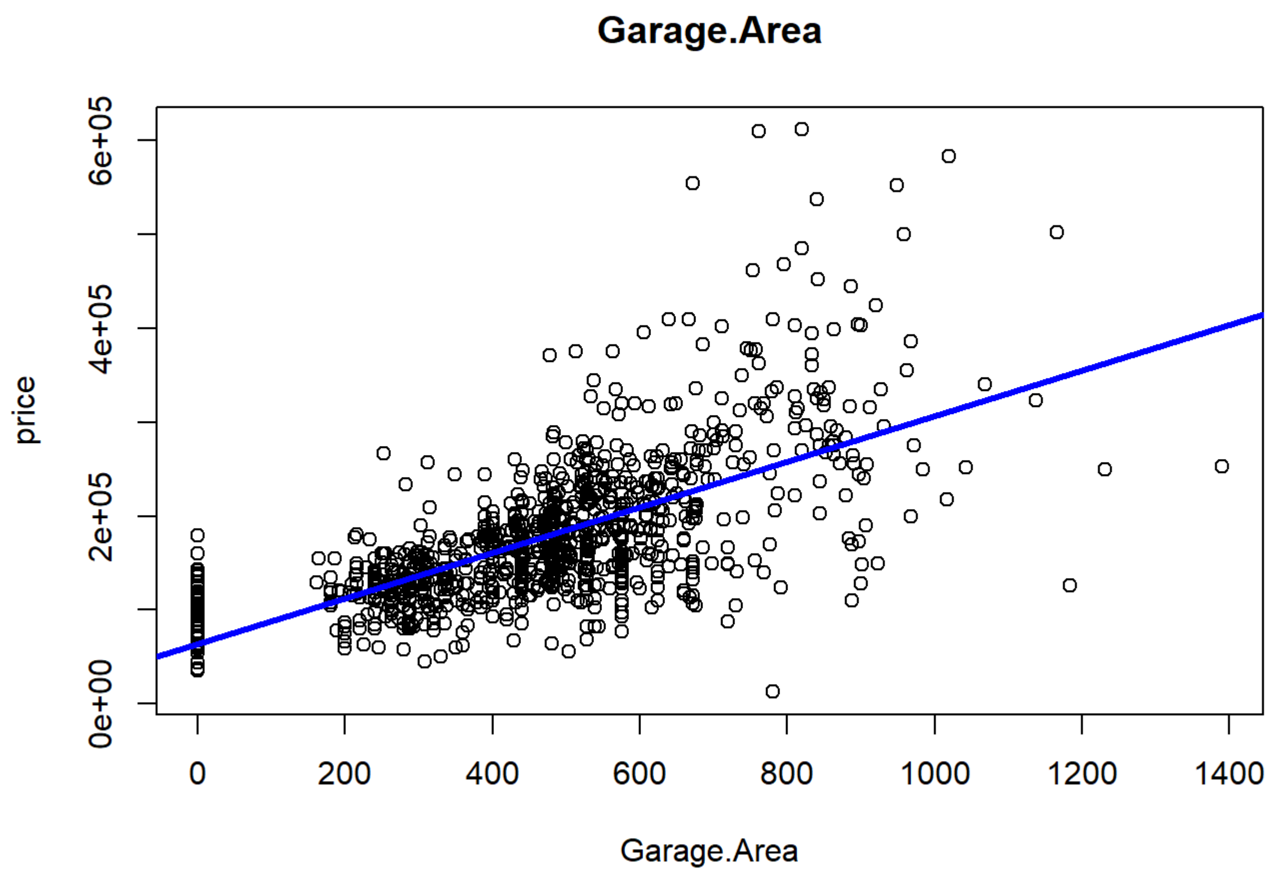

Exter.QualTA
price
0e+00
2e+05
4e+05
6e+05
0.0
0.2
0.4
0.6
0.8
1.0
Exter.QualTA


BsmtFin.SF.1
price
0e+00
2e+05
4e+05
6e+05
0
500
1000
1500
2000
BsmtFin.SF.1


Total.Bsmt.SF
price
0e+00
2e+05
4e+05
6e+05
0
500
1000
1500
2000
2500
3000
Total.Bsmt.SF


X1st.Flr.SF
price
0e+00
2e+05
4e+05
6e+05
500
1000
1500
2000
2500
X1st.Flr.SF


Garage.Cars
price
0e+00
2e+05
4e+05
6e+05
0
1
2
3
4
5
Garage.Cars



Most of the predictors have an approximately linear relationship with house price, except `Year.Built` , `Overall.Qual` , and `Garage.Cars` . We will ignore this for now but will address it in our final model.

From the `Lot.Area` plot, we see that there is one house with an abnormally large lot. We will ignore this for now but will address it later.

Correlation with target

```
df_numeric = dplyr::select_if(new_houses_df[,c(selected_predictors, "price")], is.numeric)

# returns the correlation of each predictor with house price
corr_with_price = cor(df_numeric)[,"price"]
corr_with_price_ordered = as.data.frame(corr_with_price[order(-corr_with_price)])
colnames(corr_with_price_ordered) = "Correlation"
corr_with_price_ordered
```

```
##                     Correlation
## price                1.0000000
## Overall.Qual         0.8047572
## area                 0.6878451
## Total.Bsmt.SF        0.6866271
## X1st.Flr.SF          0.6609781
## Garage.Cars          0.6606549
## Garage.Area          0.6506753
## Year.Built           0.5975933
## Year.Remod.Add       0.5510788
## BsmtFin.SF.1         0.5017362
## NeighborhoodNridgHt  0.4732493
## Mas.Vnr.TypeStone    0.4301969
## Lot.Area             0.2807551
## Exter.QualTA        -0.5761468
```

`Overall.Qual` and `area` were the most strongly correlated with house price. This means these predictors had the strongest linear relationships with house price.

Removing multicollinearity

Multicollinear predictors are correlated with each other. This increases the variance of their estimated coefficients. Hence, we want to remove multicollinear predictors to improve the interpretability of our model. A multicollinear predictor has a high variance inflation factor.

Variance inflation factors

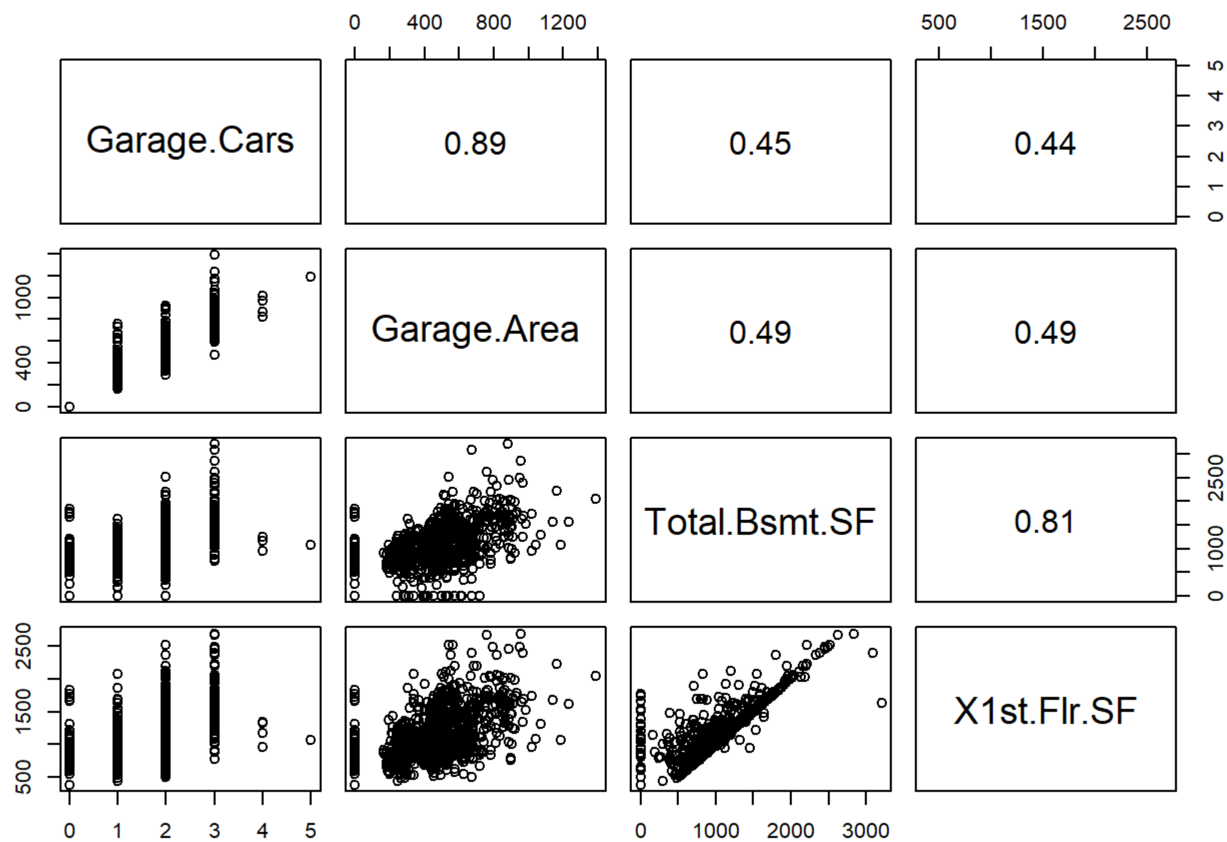
```
vif(linear)
```

```
##                area            Lot.Area NeighborhoodNridgHt        Overall.Qual
##            1.853816            1.123700            1.400065            3.178820
##          Year.Built      Year.Remod.Add   Mas.Vnr.TypeStone        Exter.QualTA
##            2.175976            1.918768            1.369253            1.838789
##        BsmtFin.SF.1       Total.Bsmt.SF         X1st.Flr.SF         Garage.Cars
##            1.427942            3.674831            3.559611            5.376516
##         Garage.Area
##            5.038021
```

`Garage.Cars` , `Garage.Area` , `X1st.Flr.SF` , `Total.Bsmt.SF` have large variance inflation factors, so we will investigate the relationships between them using a pair plot.

```
# Used to plot the correlations in the pair plot
panel.cor <- function(x, y) {
  usr <- par("usr")
  on.exit(par("usr"))
  par(usr = c(0, 1, 0, 1))
  r <- round(cor(x, y), digits = 2)
  # text size
  text(0.5, 0.5, r, cex=1.5)
}
```

```
pairs(~ Garage.Cars + Garage.Area + Total.Bsmt.SF + X1st.Flr.SF, data=new_houses_df, upper.panel
=panel.cor)
```




```
sprintf("[Garage.Area - Price] Correlation: %.2f", cor(new_houses_df$Garage.Area, new_houses_df
$price))
```

```
## [1] "[Garage.Area - Price] Correlation: 0.65"
```

```
sprintf("[Garage.Cars - Price] Correlation: %.2f", cor(new_houses_df$Garage.Cars, new_houses_df
$price))
```

```
## [1] "[Garage.Cars - Price] Correlation: 0.66"
```

```
sprintf("[Total.Bsmt.SF - Price] Correlation: %.2f", cor(new_houses_df$Total.Bsmt.SF, new_houses
_df$price))
```

```
## [1] "[Total.Bsmt.SF - Price] Correlation: 0.69"
```

```
sprintf("[X1st.Flr.SF - Price] Correlation: %.2f", cor(new_houses_df$X1st.Flr.SF, new_houses_df
$price))
```

```
## [1] "[X1st.Flr.SF - Price] Correlation: 0.66"
```

`Garage.Cars` and `Garage.Area` are strongly positively correlated (r = 0.89). This is because a larger garage can fit more cars. We will remove `Garage.Area` since it is less correlated with the target than `Garage.Cars`.

Similarly, `Total.Bsmt.SF` and `X1st.Flr.SF` are strongly positively correlated (r = 0.81). This is because a house with a large 1st floor also tends to have a large basement. We will remove `X1st.Flr.SF` since it is less correlated with the target than `Total.Bsmt.SF`.

```
predictors_subset = selected_predictors[-which(selected_predictors %in% c('Garage.Area', 'X1st.F
lr.SF'))]

# Paste the new predictors into a formula string
right_hand_side = paste(predictors_subset, collapse=" + ")
formula_string = paste("price ~", right_hand_side, collapse = "")
linear= lm(formula_string, data=new_houses_df)
```

```
summary(linear)
```

```
##
## Call:
## lm(formula = formula_string, data = new_houses_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -122048  -13935    -495   14540  210562
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -1.022e+06  1.203e+05  -8.496  < 2e-16 ***
## area                 5.248e+01  2.465e+00  21.294  < 2e-16 ***
## Lot.Area             9.794e-01  1.186e-01   8.258 4.73e-16 ***
## NeighborhoodNridgHt  3.186e+04  4.453e+03   7.154 1.64e-12 ***
## Overall.Qual         1.331e+04  1.118e+03  11.900  < 2e-16 ***
## Year.Built           2.184e+02  4.368e+01   5.000 6.79e-07 ***
## Year.Remod.Add       2.771e+02  5.951e+01   4.656 3.66e-06 ***
## Mas.Vnr.TypeStone    1.759e+04  3.430e+03   5.127 3.55e-07 ***
## Exter.QualTA        -1.036e+04  2.484e+03  -4.172 3.29e-05 ***
## BsmtFin.SF.1         2.922e+01  2.376e+00  12.297  < 2e-16 ***
## Total.Bsmt.SF        2.447e+01  2.857e+00   8.567  < 2e-16 ***
## Garage.Cars          8.594e+03  1.656e+03   5.191 2.53e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 28320 on 988 degrees of freedom
## Multiple R-squared:  0.8713, Adjusted R-squared:  0.8699
## F-statistic: 608.2 on 11 and 988 DF,  p-value: < 2.2e-16
```

```
print(vif(linear))
```

```
##                area            Lot.Area NeighborhoodNridgHt        Overall.Qual
##            1.627403            1.104871            1.394456            3.156057
##          Year.Built      Year.Remod.Add   Mas.Vnr.TypeStone        Exter.QualTA
##            2.175020            1.917556            1.355665            1.836752
##        BsmtFin.SF.1       Total.Bsmt.SF         Garage.Cars
##            1.424103            1.994839            1.928615
```

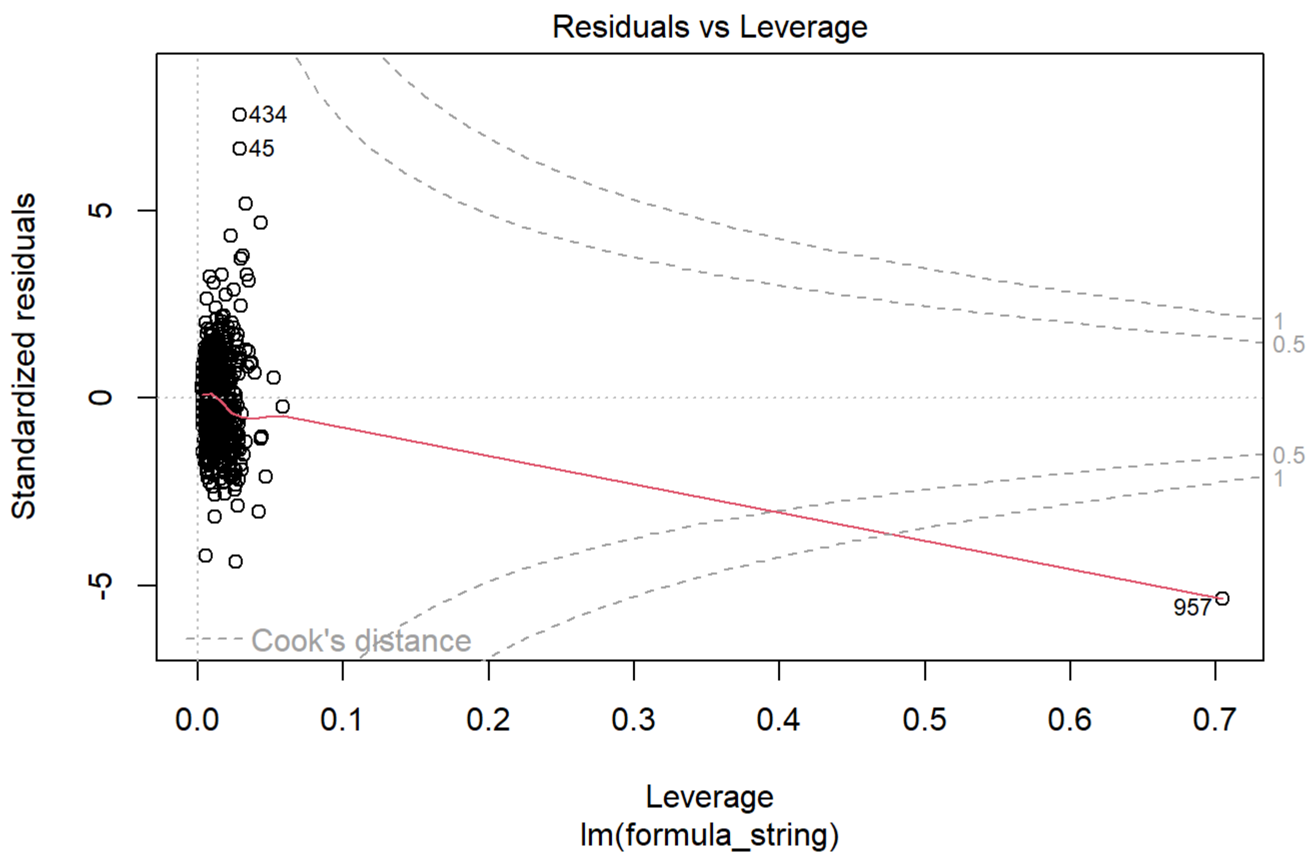
After removing `Total.Bsmt.SF` and `X1st.Flr.SF` , all the remaining predictors are statistically significant and have small variance inflation factors. This suggests we have removed multicollinearity, which has improved the interpretability of our model.

Model 1: Original

1.1 Influential observations

An observation is influential if its deletion significantly changes the fitted model. An influential observation has both a high leverage and a large standardized residual.

```
plot(linear, which=5)
```



Observation 957 is the most influential observation. We will investigate its predictor values to determine why.

```
new_houses_df[957, c("price", selected_predictors)]
```

```
##      price area Lot.Area NeighborhoodNridgHt Overall.Qual Year.Built
## 957 375000 2036   215245                   0            7       1965
##     Year.Remod.Add Mas.Vnr.TypeStone Exter.QualTA BsmtFin.SF.1 Total.Bsmt.SF
## 957           1965                 0            1         1236          2136
##     X1st.Flr.SF Garage.Cars Garage.Area
## 957        2036           2         513
```

Observation 957 has an abnormally large `Lot.Area` . It is the abnormal observation we identified in our exploratory data analysis.

```
resid(linear)[957]
```

```
##       957
## -82684.37
```

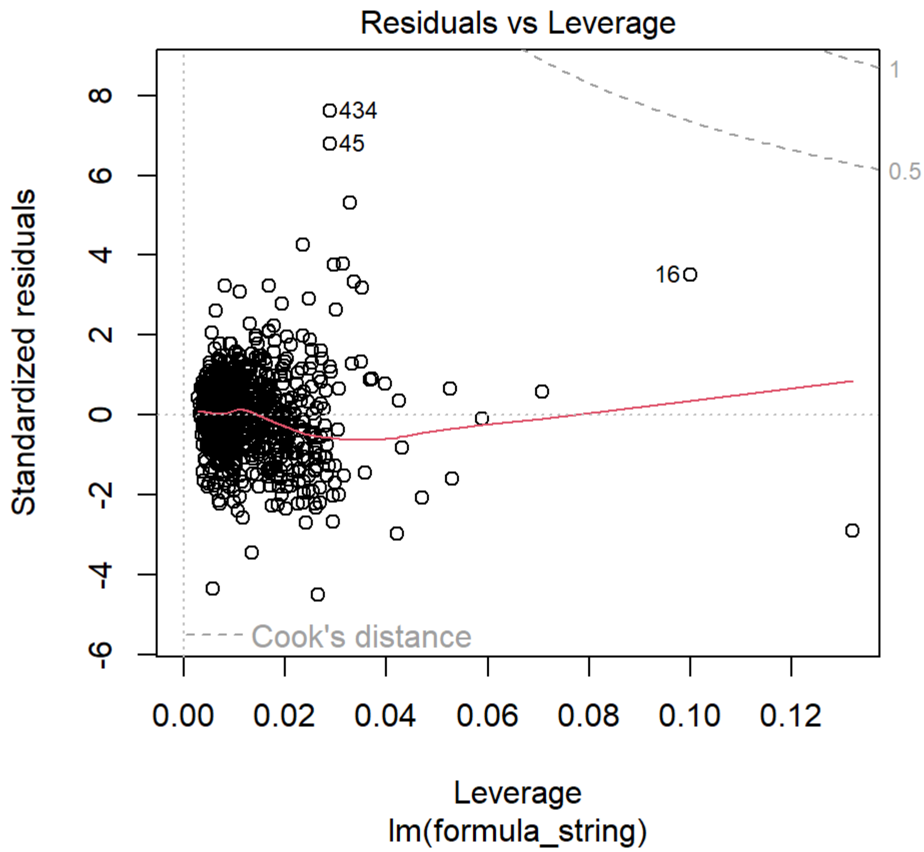
Observation 957 has a very large negative residual, indicating our model greatly overestimated its house price. Since this observation greatly changes our fitted model, we will remove it and re-train our model.

```
new_houses_df = new_houses_df[-c(957), ]
linear = lm(formula_string, data=new_houses_df)
```

```
summary(linear)
```

```
##
## Call:
## lm(formula = formula_string, data = new_houses_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -124241  -13593     347   14270  209380
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -1.036e+06  1.186e+05  -8.738  < 2e-16 ***
## area                  4.974e+01  2.481e+00  20.047  < 2e-16 ***
## Lot.Area              1.953e+00  2.136e-01   9.146  < 2e-16 ***
## NeighborhoodNridgHt   3.203e+04  4.390e+03   7.296 6.10e-13 ***
## Overall.Qual          1.381e+04  1.106e+03  12.485  < 2e-16 ***
## Year.Built            2.278e+02  4.309e+01   5.287 1.53e-07 ***
## Year.Remod.Add        2.724e+02  5.867e+01   4.643 3.90e-06 ***
## Mas.Vnr.TypeStone     1.776e+04  3.382e+03   5.251 1.85e-07 ***
## Exter.QualTA         -1.035e+04  2.448e+03  -4.227 2.59e-05 ***
## BsmtFin.SF.1          2.808e+01  2.352e+00  11.940  < 2e-16 ***
## Total.Bsmt.SF         2.360e+01  2.821e+00   8.365  < 2e-16 ***
## Garage.Cars           7.569e+03  1.643e+03   4.607 4.61e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 27920 on 987 degrees of freedom
## Multiple R-squared:  0.8743, Adjusted R-squared:  0.8729
## F-statistic: 624.1 on 11 and 987 DF,  p-value: < 2.2e-16
```

```
plot(linear, which=5)
```



Cook’s distance

An influential observation has a Cook’s distance greater than $4/n$, where n is the no. of observations.

$$D_i = \frac{1}{p}\gamma_i^2(\frac{h_{ii}}{1-h_{ii}})$$

```
influential_indices = which(cooks.distance(linear) > 4 /
                            length(cooks.distance(linear)), arr.ind=TRUE)
length(influential_indices)
```

```
## [1] 75
```

Outliers

An outlier has an abnormal target value given its predictor value. An outlier is an observation with a standardized residual whose absolute value is greater than 2.

$$\gamma_i = \frac{e_i}{\widehat{\sigma}\sqrt{1-h_{ii}}} > 2$$

```
length(which(abs(rstandard(linear)) > 2))
```

```
## [1] 45
```

High-leverage observations

A high leverage observation has a abnormal predictor values. A high leverage observation has a leverage (h_{ii}) that satisfies:

$$h_{ii} > 2\frac{1}{n}\sum_{i=1}^{n} h_{ii}$$

```
length(which(hatvalues(linear) > 2 * mean(hatvalues(linear))))
```

```
## [1] 83
```

1.2 Model evaluation

PRESS statistic

The PRESS (Prediction Error Sum of Squares) statistic measures the prediction error of a model. It is the same as the leave-one-out cross validation (LOOCV) mean squared error. We will use the square root of the PRESS statistic to get the root mean squared error, because it has the same units as the target ($USD).

$$\mathrm{PRESS} = \frac{1}{n}\sum_{i=1}^{n} e_{[i]}^2 = \frac{1}{n}\sum_{i=1}^{n}(\frac{e_i}{1-h_{ii}})^2$$

```
press = sum((resid(linear) / (1 - hatvalues(linear)))^2) / nrow(new_houses_df)
loocv_rmse = sqrt(press)
format_loocv_rmse = format(round(loocv_rmse, 2), nsmall=1, big.mark=",")

paste("LOOCV RMSE: $", format_loocv_rmse)
```

```
## [1] "LOOCV RMSE: $ 28,329.69"
```

AIC, BIC, Adjusted R-squared

These metrics are used to compare models with different no. of predictors. In other words, they balance goodness of fit and model complexity.

BIC prefers smaller models than AIC. The model with the smallest AIC or BIC is preferred.

R-squared is the proportion of variation in the target (house price) that is explained by the predictors. Since, R-squared always increases as the no. of predictors increases, adjusted R-squared is used instead. The model with the largest adjusted R-squared is preferred.

```
sprintf("AIC: %.2f", AIC(linear))
```

```
## [1] "AIC: 23302.41"
```

```
sprintf("BIC: %.2f", BIC(linear))
```

```
## [1] "BIC: 23366.19"
```

```
sprintf("Adjusted R squared: %.2f", summary(linear)$adj.r.squared)
```

```
## [1] "Adjusted R squared: 0.87"
```

1.3 Model diagnostics

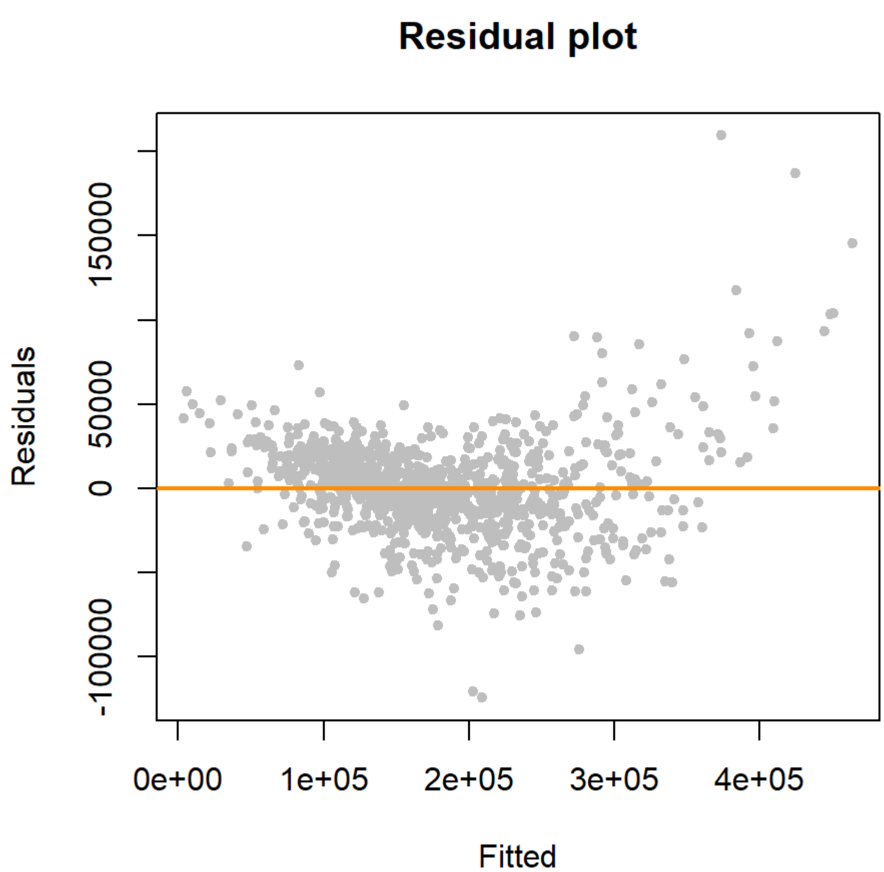
Linear regression makes 4 main assumptions about the data, called the LINE assumptions.

Linearity: y has a linear relationship with each predictor.
Independence: The observations are independent.
Normality: The residuals follow a normal distribution.
Equal Variance: The variance of the residuals is equal for all $\hat{y}$.

When these assumptions are violated, we cannot trust our model.

Residual plot, Breusch-Pagan test

```
plot(fitted(linear), resid(linear), col = "grey", pch = 20,
     xlab = "Fitted", ylab = "Residuals", main = "Residual plot")
abline(h = 0, col = "darkorange", lwd = 2)
```



The linearity assumption is violated because the residuals are not centered around 0.

The Breusch-Pagan test:
$H_0 : Var(\varepsilon)$ is constant for all $\hat{y}$
$H_a : Var(\varepsilon)$ varies depending on $\hat{y}$

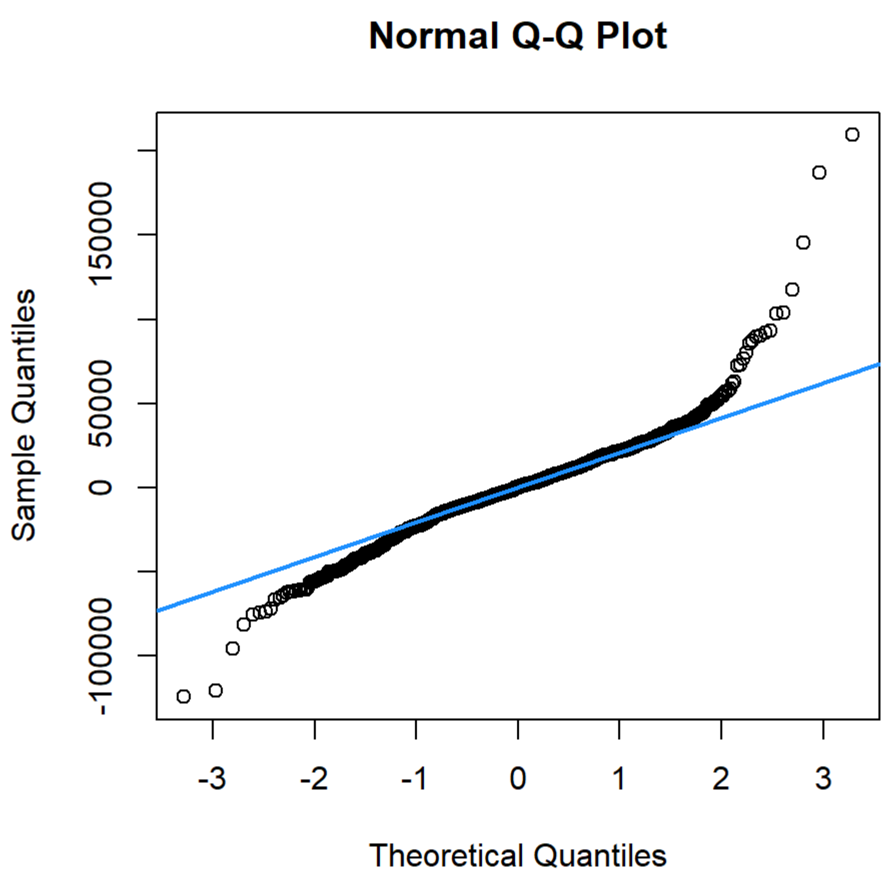
```
bptest(linear)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  linear
## BP = 180.5, df = 11, p-value < 2.2e-16
```

The constant variance assumption is violated because the variance of the residuals changes with $\hat{y}$ in the residual plot, and the p-value of the Breusch-Pagan test is smaller than 0.05.

Normal QQ Plot, Shapiro-Wilk test

```
qqnorm(resid(linear))
qqline(resid(linear), col = "dodgerblue", lwd = 2)
```



The Shapiro-Wilk test:
$H_0 : \varepsilon$ is normally distributed
$H_a : \varepsilon$ is not normally distributed

```
shapiro.test(resid(linear))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(linear)
## W = 0.92838, p-value < 2.2e-16
```

The normality assumption is violated because the normal quantile-quantile plot of the residuals does not follow a straight line, and the p-value of the Shapiro-Wilk test is smaller than 0.05.

Our model violates many of the assumptions of linear regression. We will try to fix these violations by transforming the dataset.

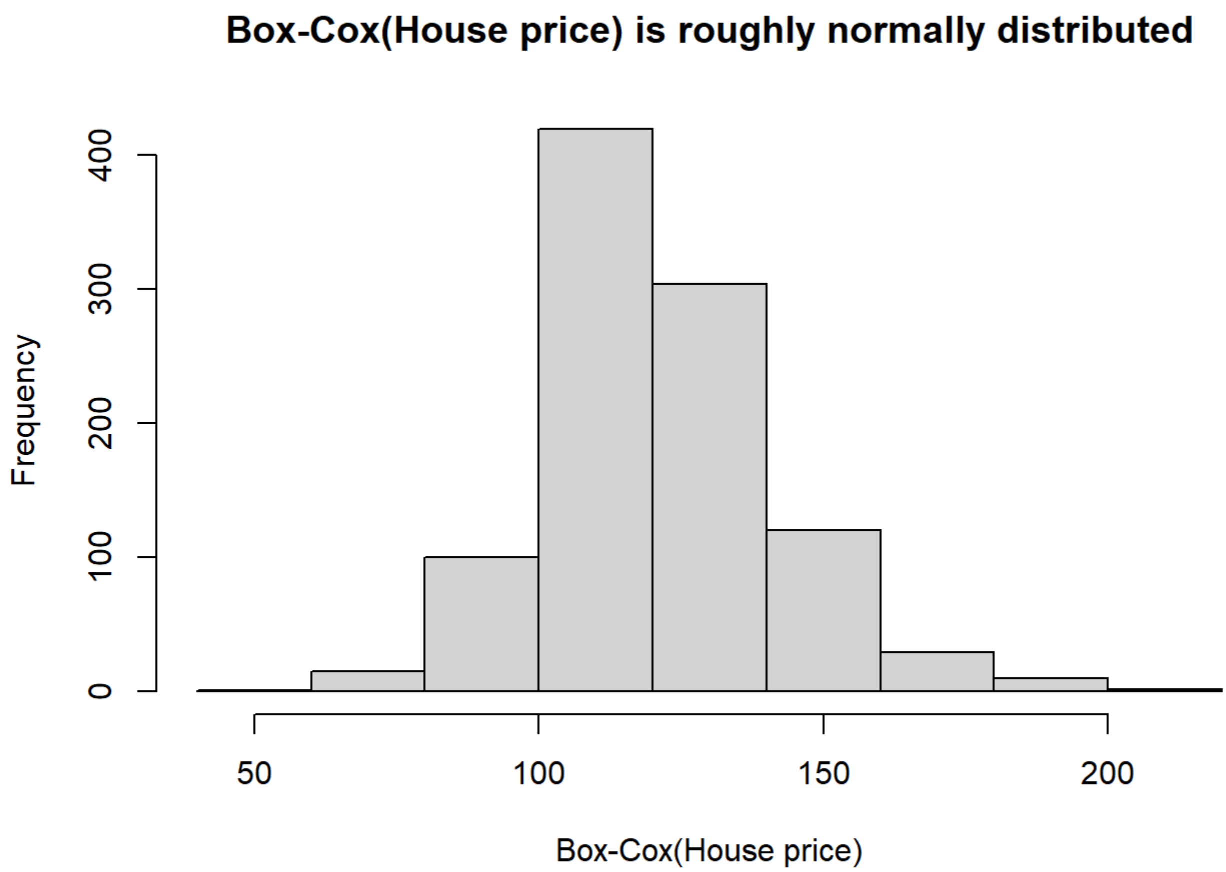
Model 2: Box-Cox transformation

To improve our model, we can transform the target variable using the Box-Cox transformation. This will transform the target variable's distribution so that it resembles a normal distribution.

$$g_\lambda(y) = \begin{cases} \frac{y^\lambda - 1}{\lambda} & \lambda \neq 0 \\ ln(y) & \lambda = 0 \end{cases}$$

```
lambdas = boxcox(linear, plot=FALSE)
best_lambda <- lambdas$x[which.max(lambdas$y)]
```

```
hist(houses_df$price ^ (best_lambda) - 1 / best_lambda, main="Box-Cox(House price) is roughly no
rmally distributed", xlab="Box-Cox(House price)")
```



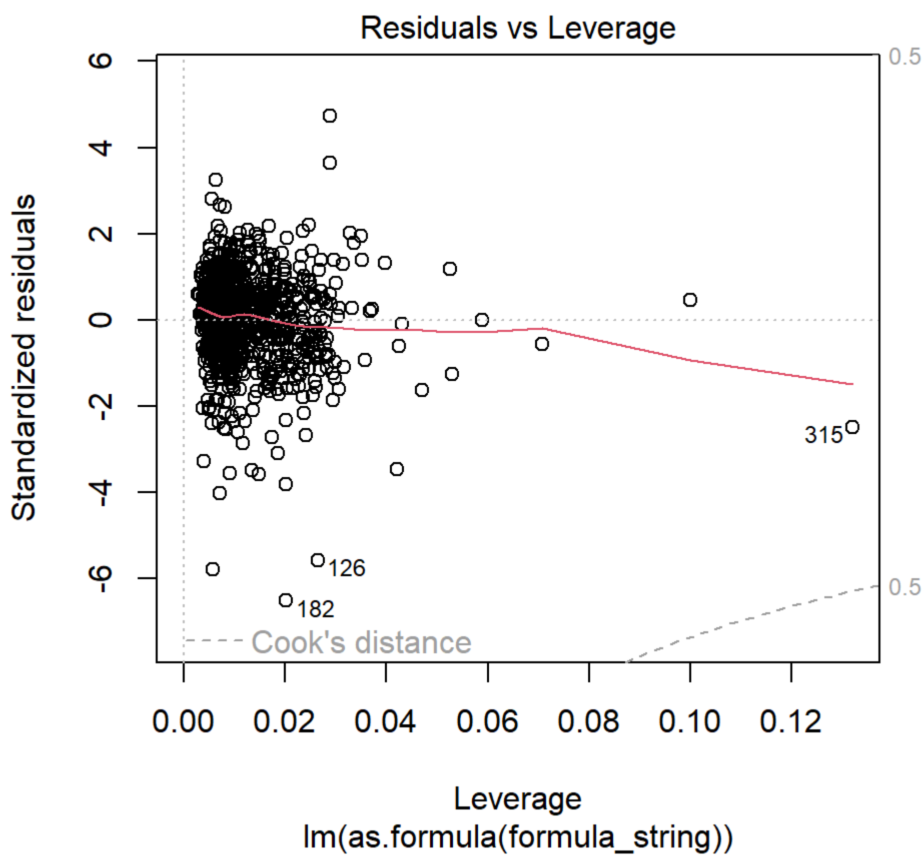

```
right_hand_side = paste(attr(linear$terms , "term.labels"), collapse="+")
formula_string = paste("((price ^ (best_lambda) - 1) / best_lambda) ~ ", right_hand_side, collap
se = "")
linear_box = lm(as.formula(formula_string), data=new_houses_df)
```

```
summary(linear_box)
```

```
##
## Call:
## lm(formula = as.formula(formula_string), data = new_houses_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -107.143   -7.787    0.804    9.896   77.476
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -8.089e+02  7.064e+01 -11.450  < 2e-16 ***
## area                 3.184e-02  1.478e-03  21.549  < 2e-16 ***
## Lot.Area             1.214e-03  1.272e-04   9.543  < 2e-16 ***
## NeighborhoodNridgHt  8.442e+00  2.614e+00   3.229 0.001282 **
## Overall.Qual         1.032e+01  6.588e-01  15.664  < 2e-16 ***
## Year.Built           2.314e-01  2.566e-02   9.016  < 2e-16 ***
## Year.Remod.Add       2.546e-01  3.495e-02   7.285 6.59e-13 ***
## Mas.Vnr.TypeStone    7.708e+00  2.014e+00   3.827 0.000138 ***
## Exter.QualTA        -3.569e+00  1.458e+00  -2.447 0.014569 *
## BsmtFin.SF.1         1.561e-02  1.401e-03  11.148  < 2e-16 ***
## Total.Bsmt.SF        1.500e-02  1.680e-03   8.928  < 2e-16 ***
## Garage.Cars          5.716e+00  9.784e-01   5.842 6.98e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.63 on 987 degrees of freedom
## Multiple R-squared:  0.8957, Adjusted R-squared:  0.8946
## F-statistic: 770.7 on 11 and 987 DF,  p-value: < 2.2e-16
```

2.1 Influential observations

```
plot(linear_box, which=5)
```




```
influential_indices = which(cooks.distance(linear_box) > 4 /
                            length(cooks.distance(linear_box)), arr.ind=TRUE)
length(influential_indices)
```

```
## [1] 62
```

The Box-Cox transformation decreased the no. of influential observations.

Outliers

```
outlier_indices = which(abs(rstandard(linear_box)) > 2)
length(outlier_indices)
```

```
## [1] 46
```

The Box-Cox transformation barely changed the no. of outliers.

High-leverage observations

```
length(which(hatvalues(linear_box) > 2 * mean(hatvalues(linear_box))))
```

```
## [1] 83
```

The Box-Cox transformation barely changed the no. of high-leverage observations.

2.2 Model evaluation

PRESS Statistic

```
y = new_houses_df$price
y_pred = (best_lambda * fitted(linear_box) + 1) ** (1/best_lambda)
loocv_rmse = sqrt(sum(((y - y_pred) / (1 - hatvalues(linear_box)))^2) / nrow(new_houses_df))
format_loocv_rmse = format(round(loocv_rmse, 2), nsmall=1, big.mark=",")

paste("LOOCV RMSE: $", format_loocv_rmse)
```

```
## [1] "LOOCV RMSE: $ 23,919.36"
```

The Box-Cox transformation decreased the model's average house price prediction error.

AIC, BIC, Adjusted R-squared

```
sprintf("AIC: %.2f", AIC(linear_box))
```

```
## [1] "AIC: 8465.29"
```

```
sprintf("BIC: %.2f", BIC(linear_box))
```

```
## [1] "BIC: 8529.08"
```

```
sprintf("Adjusted R squared: %.2f", summary(linear_box)$adj.r.squared)
```

```
## [1] "Adjusted R squared: 0.89"
```

The Box-Cox transformation dramatically decreased the AIC and BIC. Furthermore, it slightly increased the adjusted R-squared to 0.89. Since the Box-Cox transformation did not increase the no. of predictors, this indicates it increased the goodness of fit.

2.3 Model diagnostics

Residual plot, Breusch-Pagan test

```
plot(fitted(linear_box), resid(linear_box), col = "grey", pch = 20,
     xlab = "Fitted", ylab = "Residuals", main = "Residual plot")
abline(h = 0, col = "darkorange", lwd = 2)
```

The Box-Cox transformation fixed the violation of the linearity assumption (the residuals are centered around 0 for all fitted values).

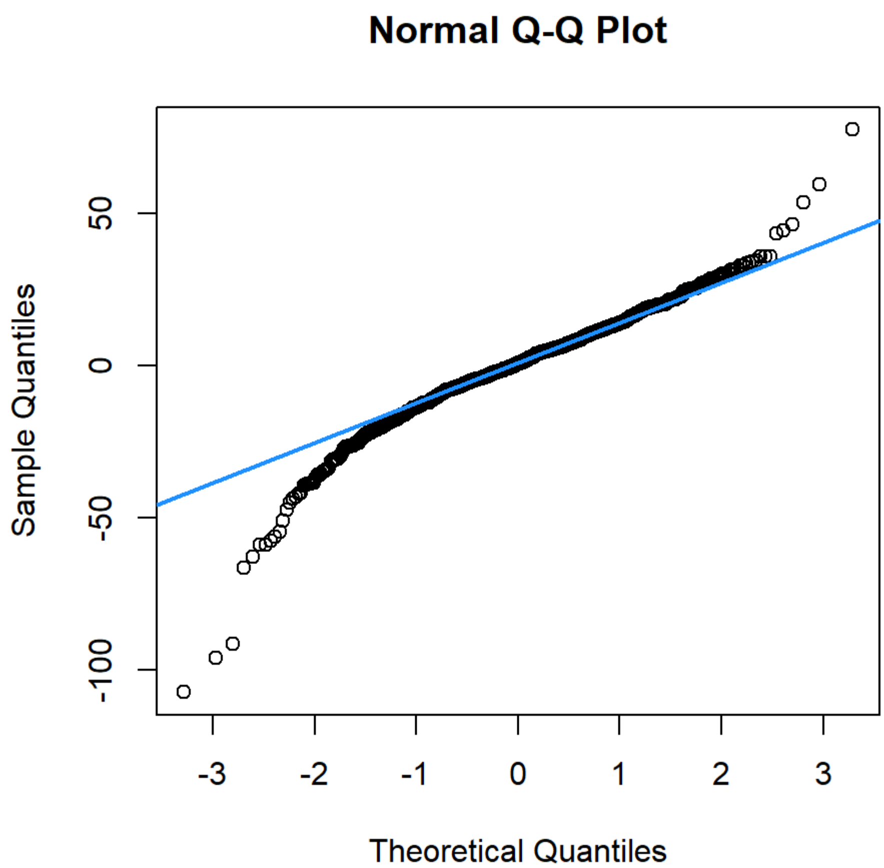
```
bptest(linear_box)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  linear_box
## BP = 48.064, df = 11, p-value = 1.391e-06
```

The Box-Cox transformation did not fix the violation of the equal variance assumption.

Normal QQ Plot, Shapiro-Wilk test

```
qqnorm(resid(linear_box))
qqline(resid(linear_box), col = "dodgerblue", lwd = 2)
```




```
shapiro.test(resid(linear_box))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(linear_box)
## W = 0.94699, p-value < 2.2e-16
```

The Box-Cox transformation did not fix the violation of the normality assumption.

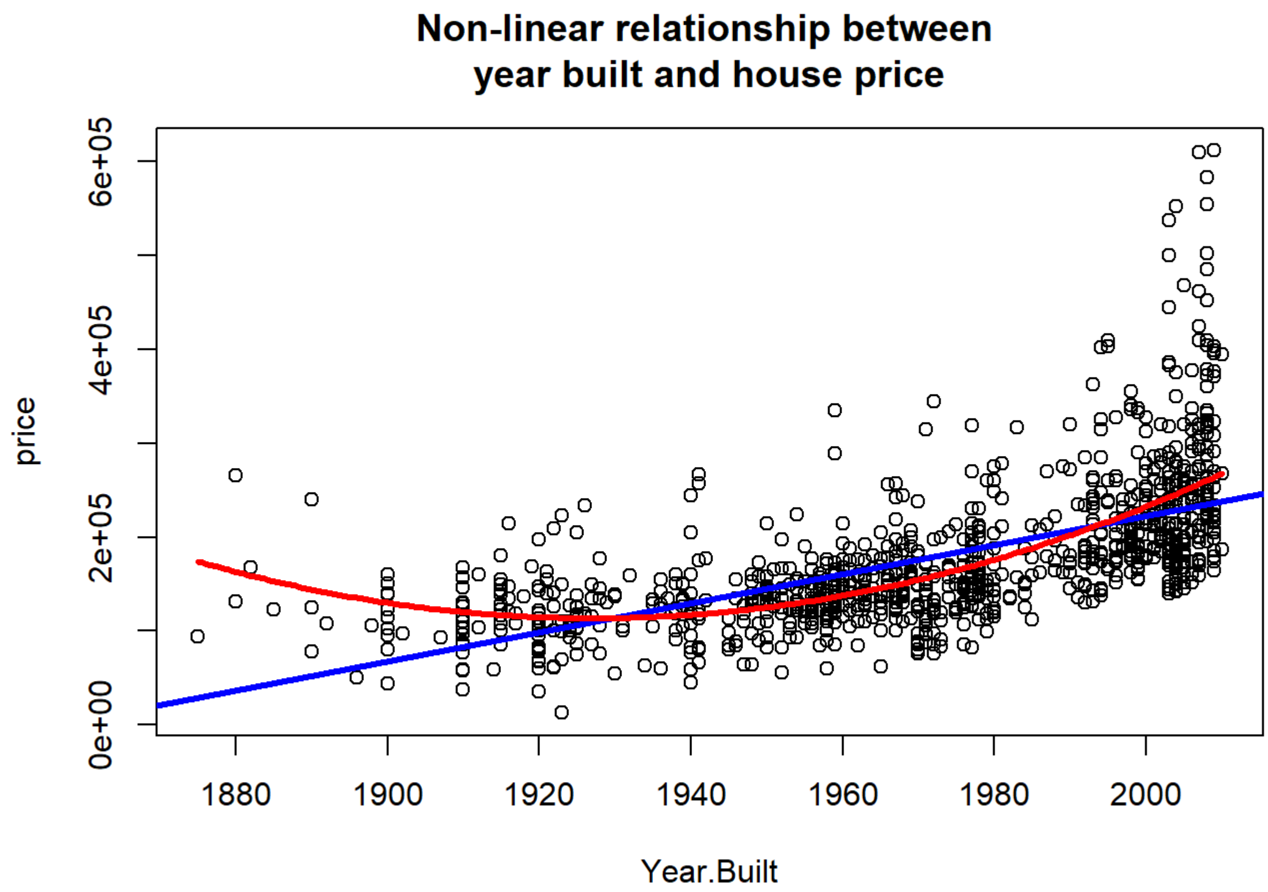
Model 3: Higher-order terms

Power terms

In our exploratory data analysis, we noticed that `Year.Built` and `Overal.Qual` had non-linear relationships with house price (we are ignoring `Garage.Cars` for the sake of time). Since linear regression assumes that each predictor is linearly related to the target, we will try to transform these predictors.

```
year_linear = lm(price ~ Year.Built, data=new_houses_df)
year_quad = lm(price ~ Year.Built + I(Year.Built^2), data=new_houses_df)

new.data <- data.frame(Year.Built = seq(from = min(new_houses_df$Year.Built),
                                    to = max(new_houses_df$Year.Built),
                                    length.out = 200))

pred_year_quad <- predict(year_quad, newdata = new.data)
```

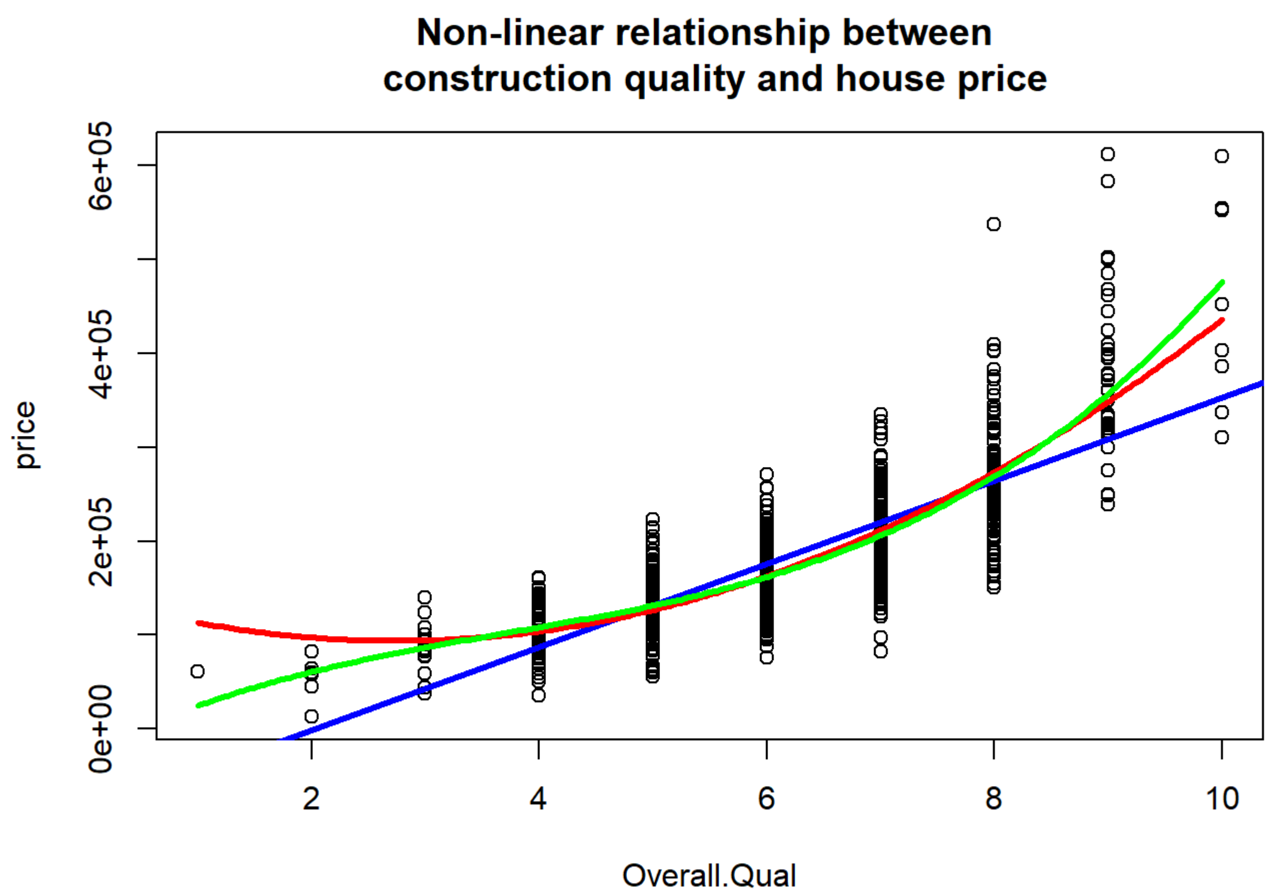
```
plot(price ~ Year.Built,
     data=new_houses_df,
     main="Non-linear relationship between \nyear built and house price")

abline(year_linear, lwd = 3, lty = 1, col = "blue")
lines(pred_year_quad ~ new.data$Year.Built, col = "red", lwd=3)
```



A quadratic polynomial is sufficient to describe the relationship between `Year.Built` and `price`.

```
quality_linear = lm(price ~ Overall.Qual, data=new_houses_df)
quality_quad = lm(price ~ Overall.Qual + I(Overall.Qual^2), data=new_houses_df)
quality_cubic = lm(price ~ Overall.Qual + I(Overall.Qual^2) + I(Overall.Qual^3), data=new_houses
_df)

new.data <- data.frame(Overall.Qual = seq(from = min(new_houses_df$Overall.Qual),
                                  to = max(new_houses_df$Overall.Qual),
                                  length.out = 200))

pred_quality_quad <- predict(quality_quad, newdata = new.data)
pred_quality_cubic <- predict(quality_cubic, newdata = new.data)
```

```
plot(price ~ Overall.Qual,
     data=new_houses_df,
     main="Non-linear relationship between \n construction quality and house price")

abline(quality_linear, lwd = 3, lty = 1, col = "blue")
lines(pred_quality_quad ~ new.data$Overall.Qual, col = "red", lwd=3)
lines(pred_quality_cubic ~ new.data$Overall.Qual, col = "green", lwd=3)
```

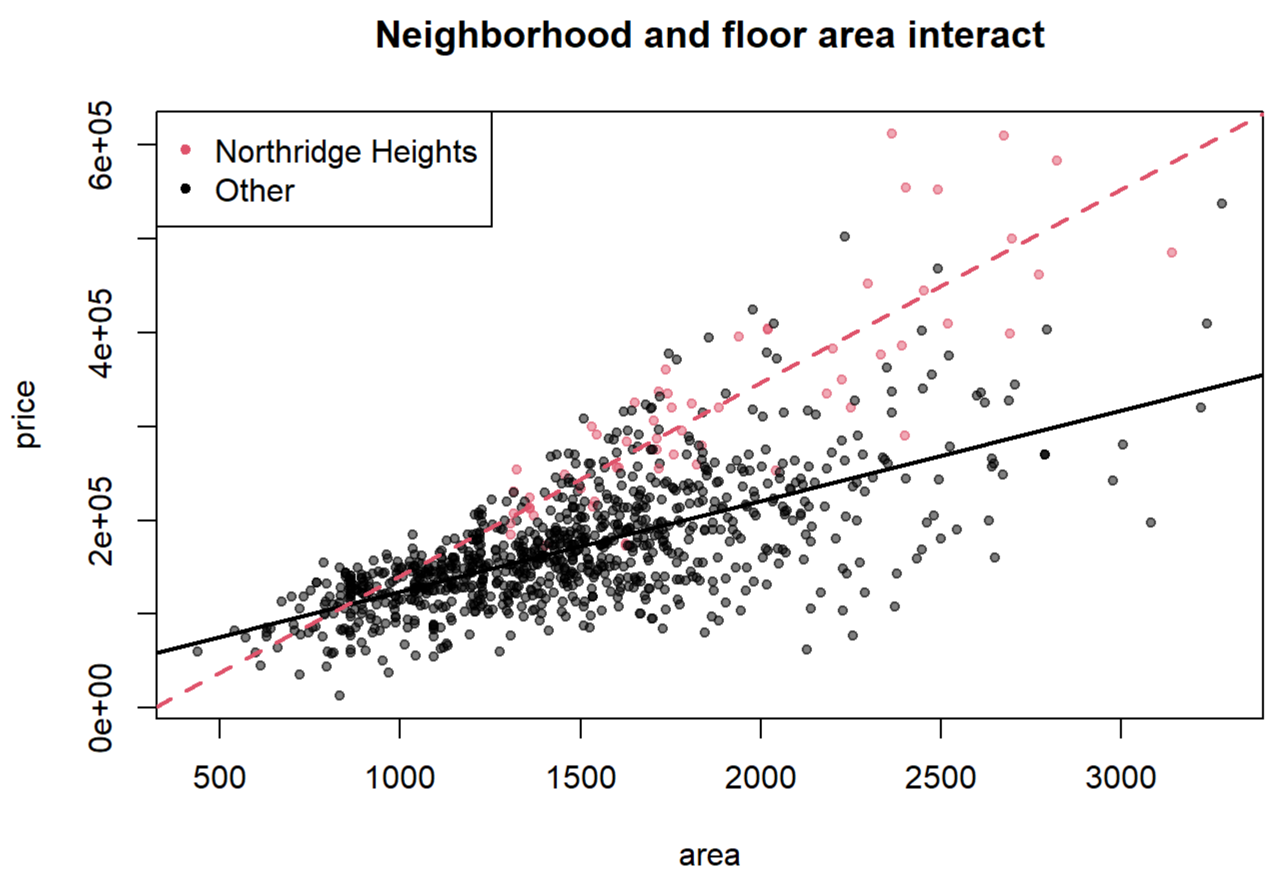


A cubic polynomial is sufficient to describe the relationship between `Overall.Qual` and `price`.

Interaction terms

We hypothesize that `area` and `NeighborhoodNridgHt` should have an interactive effect on house price. This is because the same unit increase in floor area should cost more in a fancy neighborhood compared to a modest neighborhood.

```
area_neigh = lm(price ~ area + NeighborhoodNridgHt + area:NeighborhoodNridgHt, data = new_houses
_df)

beta_hats = coef(area_neigh)
int_not_neigh = beta_hats[1]
slope_not_neigh = beta_hats[2]
int_neigh = beta_hats[1] + beta_hats[3]
slope_neigh = beta_hats[2] + beta_hats[4]


plot(price ~ area,
     data = new_houses_df,
     col = alpha(NeighborhoodNridgHt + 1, 0.5), pch=20, cex = 1,
     main="Neighborhood and floor area interact")

abline(int_not_neigh, slope_not_neigh, col = 1, lty = 1, lwd = 2)
abline(int_neigh, slope_neigh, col = 2, lty = 2, lwd = 2)
legend("topleft", c("Northridge Heights", "Other"), pch=20, col = c(2, 1))
```

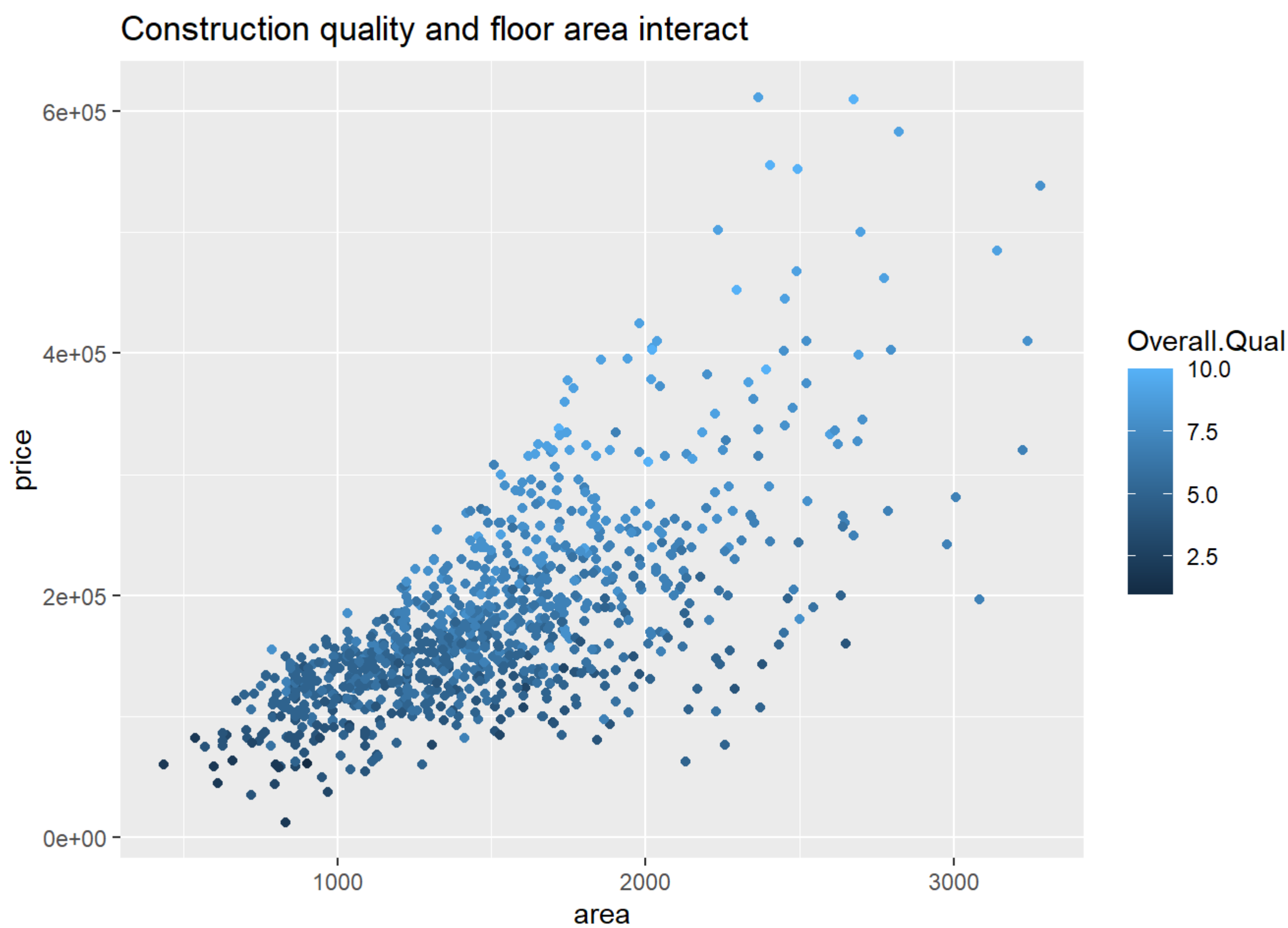


We observe that for a unit increase in floor area, the price of houses in the Northridge Heights neighborhood increase more than houses in other neighborhoods. This suggests that Northridge Heights is a luxury neighborhood.

Similarly, we hypothesize that `area` and `Overall.Qual` should have an interactive effect on house price. This is because the same unit increase in floor area should cost more when the construction quality of the house is higher.

```
qplot(area, price,
      colour=Overall.Qual,
      main="Construction quality and floor area interact",
      data = new_houses_df)
```

```
## Warning: `qplot()` was deprecated in ggplot2 3.4.0.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



We observe that for houses with the same area, the houses with a higher construction quality are more expensive.

Adding the derived predictors to the model

```
right_hand_side = paste(c(attr(linear$terms , "term.labels"),
                          "I(Year.Built^2)",
                          "I(Overall.Qual^2)",
                          "I(Overall.Qual^3)",
                          "NeighborhoodNridgHt:area",
                          "Overall.Qual:area"),
                        collapse="+")

formula_string = paste("((price ^ (best_lambda) - 1) / best_lambda) ~ ", right_hand_side, collap
se = "")
linear_interact = lm(as.formula(formula_string), data=new_houses_df)
```

```
summary(linear_interact)
```

```
##
## Call:
## lm(formula = as.formula(formula_string), data = new_houses_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -104.296   -7.587    0.735    8.909   52.210
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -5.344e+03  2.719e+03  -1.965 0.049661 *
## area                      5.715e-03  6.239e-03   0.916 0.359885
## Lot.Area                  1.130e-03  1.238e-04   9.124  < 2e-16 ***
## NeighborhoodNridgHt      -2.156e+01  1.028e+01  -2.097 0.036225 *
## Overall.Qual              3.537e+01  8.030e+00   4.404 1.18e-05 ***
## Year.Built                4.783e+00  2.762e+00   1.732 0.083607 .
## Year.Remod.Add            2.865e-01  3.610e-02   7.937 5.63e-15 ***
## Mas.Vnr.TypeStone         7.904e+00  1.989e+00   3.973 7.61e-05 ***
## Exter.QualTA             -5.320e+00  1.584e+00  -3.359 0.000812 ***
## BsmtFin.SF.1              1.365e-02  1.384e-03   9.859  < 2e-16 ***
## Total.Bsmt.SF             1.421e-02  1.634e-03   8.698  < 2e-16 ***
## Garage.Cars               5.838e+00  9.543e-01   6.118 1.37e-09 ***
## I(Year.Built^2)          -1.162e-03  7.074e-04  -1.643 0.100801
## I(Overall.Qual^2)        -5.448e+00  1.427e+00  -3.818 0.000143 ***
## I(Overall.Qual^3)         2.997e-01  7.890e-02   3.798 0.000155 ***
## area:NeighborhoodNridgHt  1.354e-02  5.535e-03   2.446 0.014603 *
## area:Overall.Qual         4.180e-03  1.008e-03   4.146 3.68e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.05 on 982 degrees of freedom
## Multiple R-squared:  0.9033, Adjusted R-squared:  0.9017
## F-statistic: 573.5 on 16 and 982 DF,  p-value: < 2.2e-16
```

Adding the higher-order predictors inflated the p-values of the original predictors. This is because the higher-order predictors are products of the original predictors, so they are correlated.

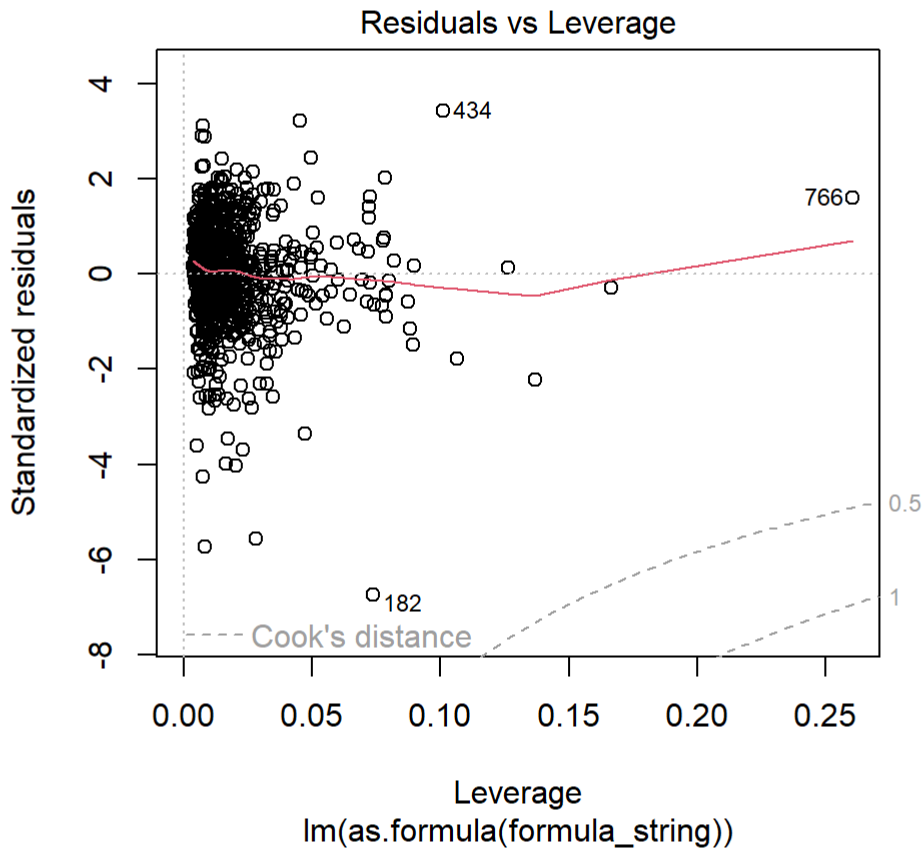
```
anova(linear_box, linear_interact)
```

```
## Analysis of Variance Table
##
## Model 1: ((price^(best_lambda) - 1)/best_lambda) ~ area + Lot.Area + NeighborhoodNridgHt +
##     Overall.Qual + Year.Built + Year.Remod.Add + Mas.Vnr.TypeStone +
##     Exter.QualTA + BsmtFin.SF.1 + Total.Bsmt.SF + Garage.Cars
## Model 2: ((price^(best_lambda) - 1)/best_lambda) ~ area + Lot.Area + NeighborhoodNridgHt +
##     Overall.Qual + Year.Built + Year.Remod.Add + Mas.Vnr.TypeStone +
##     Exter.QualTA + BsmtFin.SF.1 + Total.Bsmt.SF + Garage.Cars +
##     I(Year.Built^2) + I(Overall.Qual^2) + I(Overall.Qual^3) +
##     NeighborhoodNridgHt:area + Overall.Qual:area
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    987 272833
## 2    982 252935  5     19899 15.451 1.195e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The ANOVA test has a p-value smaller than 0.05. Hence, at least one of the higher-order predictors is linearly related to house price, given that all the original predictors are used in the model.

3.1 Influential observations

```
plot(linear_interact, which=5)
```




```
influential_indices = which(cooks.distance(linear_interact) > 4 /
                            length(cooks.distance(linear_interact)), arr.ind=TRUE)
length(influential_indices)
```

```
## [1] 60
```

The higher-order predictors barely changed the no. of influential observations.

Outliers

```
length(which(abs(rstandard(linear_interact)) > 2))
```

```
## [1] 50
```

The higher-order predictors barely changed the no. of outliers.

High-leverage observations

```
length(which(hatvalues(linear_interact) > 2 * mean(hatvalues(linear_interact))))
```

```
## [1] 90
```

The higher-order predictors increased the no. of high-leverage observations. This is because adding more predictors increased the dimensionality of the predictor space, so the observations are further apart.

3.2 Model evaluation

PRESS Statistic

```
y = new_houses_df$price
y_pred = (best_lambda * fitted(linear_interact) + 1) ** (1/best_lambda)

loocv_rmse = sqrt(sum(((y - y_pred) / (1 - hatvalues(linear_interact)))^2) / nrow(new_houses_d
f))

format_loocv_rmse = format(round(loocv_rmse, 2), nsmall=1, big.mark=",")
paste("LOOCV RMSE: $", format_loocv_rmse)
```

```
## [1] "LOOCV RMSE: $ 22,316.35"
```

The higher-order predictors decreased the model's average house price prediction error.

AIC, BIC, Adjusted R-squared

```
sprintf("AIC: %.2f", AIC(linear_interact))
```

```
## [1] "AIC: 8399.64"
```

```
sprintf("BIC: %.2f", BIC(linear_interact))
```

```
## [1] "BIC: 8487.96"
```

```
sprintf("Adjusted R squared: %.2f", summary(linear_interact)$adj.r.squared)
```

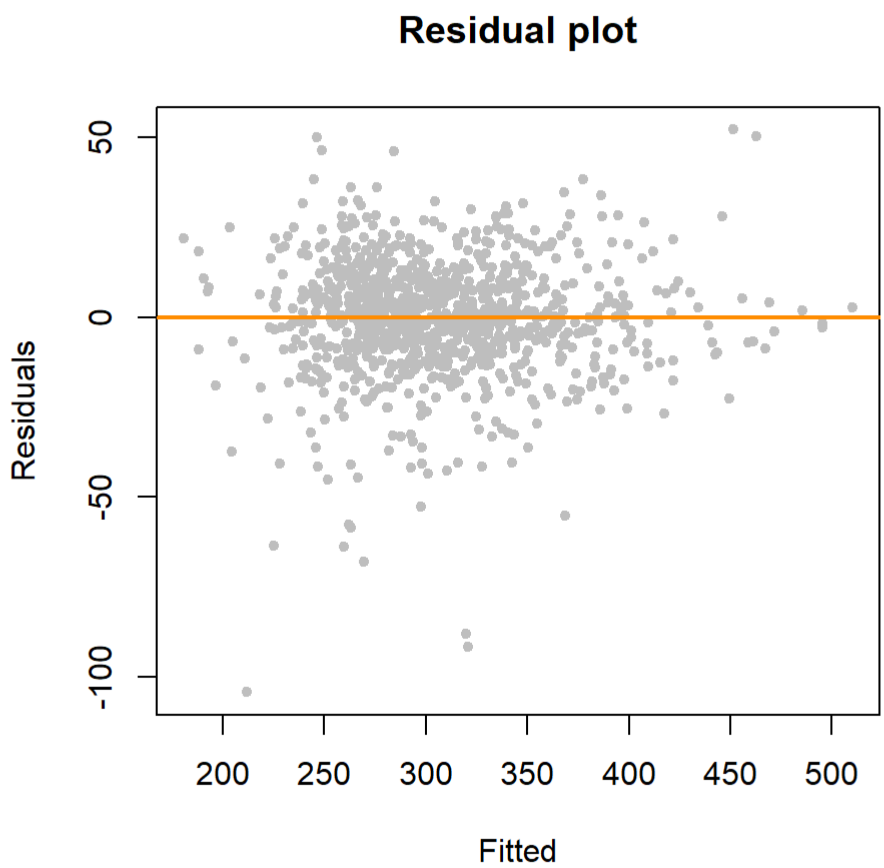
```
## [1] "Adjusted R squared: 0.90"
```

The higher-order predictors decreased the AIC and BIC slightly. The higher-order predictors also increased the adjusted R-squared slightly to 0.90. This indicates that the higher-order predictors significantly increased the goodness of fit (enough to overcome the effect of increasing the no. of predictors).

3.3 Model diagnostics

Residual plot, Breusch-Pagan test

```
plot(fitted(linear_interact), resid(linear_interact), col = "grey", pch = 20,
     xlab = "Fitted", ylab = "Residuals", main = "Residual plot")
abline(h = 0, col = "darkorange", lwd = 2)
```



The higher-order predictors did not affect the linearity assumption (it is still satisfied).

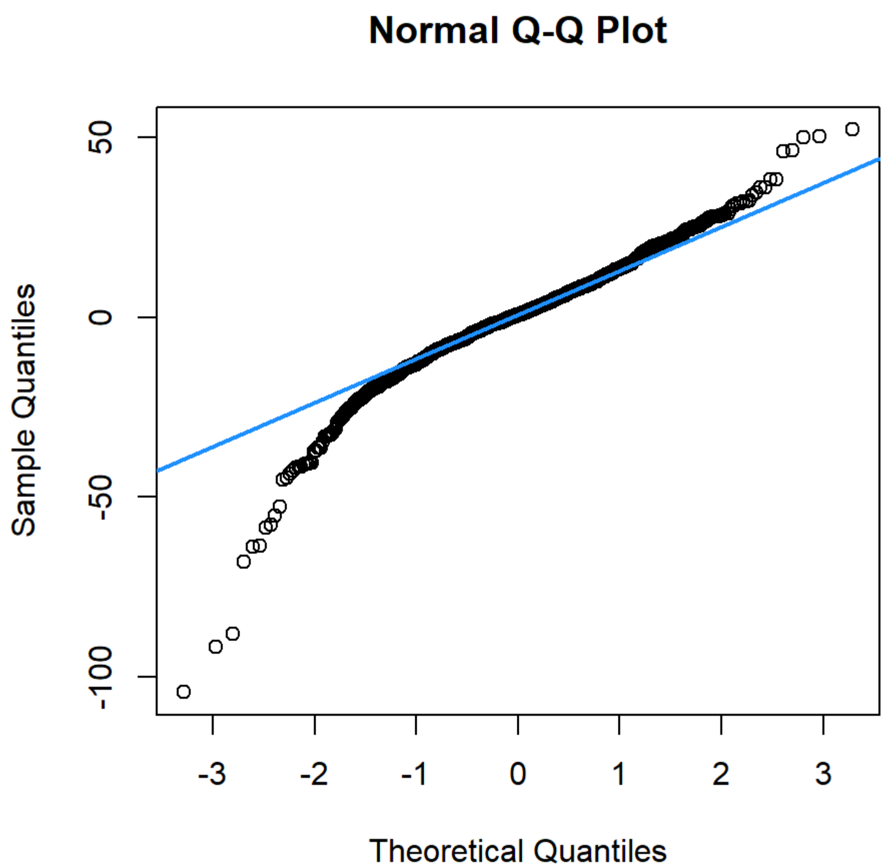
```
bptest(linear_interact)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  linear_interact
## BP = 59.504, df = 16, p-value = 6.343e-07
```

The higher-order predictors did not fix the violation of the equal variance assumption.

Normal QQ Plot, Shapiro-Wilk test

```
qqnorm(resid(linear_interact))
qqline(resid(linear_interact), col = "dodgerblue", lwd = 2)
```




```
shapiro.test(resid(linear_interact))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(linear_interact)
## W = 0.94238, p-value < 2.2e-16
```

The higher-order predictors did not fix the violation of the normality assumption.

Mean absolute error of final model

Training set error

```
y = new_houses_df$price
y_pred = (best_lambda * fitted(linear_interact) + 1) ** (1/best_lambda)
(sum(abs(y - y_pred)))/nrow(new_houses_df)
```

```
## [1] 15441.52
```

LOOCV error

```r
e_cv_linear = numeric(nrow(new_houses_df))
for (i in 1:nrow(new_houses_df))
{
  # Remove the ith observation
  training_data <- new_houses_df[-i, ]

  # Fit models using training_data
  cv_linear <- lm(as.formula(formula_string), data=training_data)

  y_pred = (best_lambda * predict(cv_linear, newdata = new_houses_df[i, ]) + 1) ** (1/best_lambd
a)

  # Prediction for the ith observation and obtain the residual
  e_cv_linear[i] <- new_houses_df[i, "price"] - y_pred
}

sum(abs(e_cv_linear))/nrow(new_houses_df)
```

```
## [1] 15769.48
```