

Data Analysis - Ames House Prices

Random85

2023-07-20

Introduction:

In this study we will attempt to find the best model to predict Housing Prices in Ames Iowa. Our group has a potential home buyer, and we would like to create a model to help predict home prices, and determine which attributes most contribute to housing prices.

This data set was acquired from Kraggle. It has 79 predictors describing 2,930 houses that were sold in Ames, Iowa between 2006-2010. The `SalePrice` column is the amount the house sold for and will be our response variable.

We will attempt to eliminate collinearity between predictors, build multiple models, and use various evaluators such as RMSE and Adjusted R^2 to determine the model with the best prediction ability.

While prediction is the main goal, we will also aim to minimize collinearity to gain insight on which predictors have large impact on the price of homes.

The Dataset

```
library(faraway)
library(dplyr)
library(psych)
library(corrplot)
library(ggplot2)
library(corrplot)
library(laers)
library(reshape2)
library(scales)
library(Hmisc)

housing_data = read.csv("dataset/AmesHousing.csv")

nrow(housing_data)
```

```
## [1] 2930
```

```
ncol(housing_data)
```

```
## [1] 82
```

```
# Coercing categorical predictors into factor variables
```

```
housing_data[is.na(housing_data)] = 1
```

```
for (i in 1:ncol(housing_data)) {
  if (typeof(housing_data[, i]) == "character") {
    if (length(unique(housing_data[, i])) >= 2) {
      housing_data[, i] = as.factor(housing_data[, i])
    }
  }
}
```

```
housing_data$Yr.Sold = as.factor(housing_data$Yr.Sold)
```

```
str(housing_data)
```

```
## 'data.frame':    2930 obs. of  82 variables:
## $ Order          : int  1 2 3 4 5 6 7 8 9 10 ...
## $ PID            : int  526301100 526350040 526351010 526353030 527105010 527105030 527127150 52714...
## $ MS.SubClass     : int  20 20 20 20 60 60 120 120 120 60 ...
## $ MS.Zoning       : Factor w/ 7 levels "A (agr)","C (all)",...: 6 5 6 6 6 6 6 6 6 ...
## $ Lot.Frontage    : num  141 80 81 93 74 78 41 43 39 60 ...
## $ Lot.Area        : int  31770 11622 14267 11160 13830 9978 4920 5005 5389 7500 ...
## $ Street          : Factor w/ 2 levels "Grvl","Pave": 2 2 2 2 2 2 2 2 2 ...
## $ Alley           : Factor w/ 3 levels "1","Grvl","Pave": 1 1 1 1 1 1 1 1 1 ...
## $ Lot.Shape       : Factor w/ 4 levels "IR1","IR2","IR3",...: 1 4 1 4 1 1 4 1 1 4 ...
## $ Land.Contour     : Factor w/ 4 levels "Bnk","HLS","Low",...: 4 4 4 4 4 4 4 2 4 4 ...
## $ Utilities       : Factor w/ 3 levels "AllPub","NoSeWa",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ Lot.Config      : Factor w/ 5 levels "Corner","CulDSac",...: 1 5 1 1 5 5 5 5 5 5 ...
## $ Land.Slope      : Factor w/ 3 levels "Gtl","Mod","Sev": 1 1 1 1 1 1 1 1 1 1 ...
## $ Neighborhood    : Factor w/ 28 levels "Blmngtn","Blueste",...: 16 16 16 16 9 9 25 25 25 9 ...
## $ Condition.1     : Factor w/ 9 levels "Artery","Feedr",...: 3 2 3 3 3 3 3 3 3 ...
## $ Condition.2     : Factor w/ 8 levels "Artery","Feedr",...: 3 3 3 3 3 3 3 3 3 ...
## $ Bldg.Type       : Factor w/ 5 levels "1Fam","2fmCon",...: 1 1 1 1 1 1 5 5 5 1 ...
## $ House.Style     : Factor w/ 8 levels "1.5Fin","1.5Unf",...: 3 3 3 3 6 6 3 3 3 6 ...
## $ Overall.Qual     : int  6 5 6 7 5 6 8 8 8 7 ...
## $ Overall.Cond    : int  5 6 6 5 5 6 5 5 5 5 ...
## $ Year.Built      : int  1960 1961 1958 1968 1997 1998 2001 1992 1995 1999 ...
## $ Year.Remod.Add  : int  1960 1961 1958 1968 1998 1998 2001 1992 1996 1999 ...
## $ Roof.Style      : Factor w/ 6 levels "Flat","Gable",...: 4 2 4 4 2 2 2 2 2 2 ...
## $ Roof.Matl       : Factor w/ 8 levels "ClyTile","CompShg",...: 2 2 2 2 2 2 2 2 2 2 ...
## $ Exterior.1st    : Factor w/ 16 levels "AsbShng","AsphShn",...: 4 14 15 4 14 14 6 7 6 14 ...
## $ Exterior.2nd    : Factor w/ 17 levels "AsbShng","AsphShn",...: 11 15 16 4 15 15 6 7 6 15 ...
## $ Mas.Vnr.Type    : Factor w/ 6 levels "", "BrkCmn", "BrkFace",...: 6 5 3 5 5 3 5 5 5 5 ...
## $ Mas.Vnr.Area    : num  112 0 108 0 0 20 0 0 0 0 ...
## $ Exter.Qual      : Factor w/ 4 levels "Ex","Fa","Gd",...: 4 4 4 3 4 4 3 3 3 4 ...
## $ Exter.Cond      : Factor w/ 5 levels "Ex","Fa","Gd",...: 5 5 5 5 5 5 5 5 5 5 ...
## $ Foundation      : Factor w/ 6 levels "BrkTil","CBlock",...: 2 2 2 2 3 3 3 3 3 3 ...
## $ Bsmt.Qual       : Factor w/ 7 levels "", "1", "Ex", "Fa",...: 7 7 7 7 5 7 5 5 5 7 ...
## $ Bsmt.Cond       : Factor w/ 7 levels "", "1", "Ex", "Fa",...: 5 7 7 7 7 7 7 7 7 7 ...
## $ Bsmt.Exposure   : Factor w/ 6 levels "", "1", "Av", "Gd",...: 4 6 6 6 6 6 5 6 6 6 ...
## $ BsmtFin.Type.1  : Factor w/ 8 levels "", "1", "ALQ", "BLQ",...: 4 7 3 3 5 5 5 3 5 8 ...
```

```

## $ BsmtFin.SF.1 : num 639 468 923 1065 791 ...
## $ BsmtFin.Type.2 : Factor w/ 8 levels "", "1", "ALQ", "BLQ", ...: 8 6 8 8 8 8 8 8 8 ...
## $ BsmtFin.SF.2 : num 0 144 0 0 0 0 0 0 0 ...
## $ Bsmt.Unf.SF : num 441 270 406 1045 137 ...
## $ Total.Bsmt.SF : num 1080 882 1329 2110 928 ...
## $ Heating : Factor w/ 6 levels "Floor", "GasA", ...: 2 2 2 2 2 2 2 2 2 ...
## $ Heating.QC : Factor w/ 5 levels "Ex", "Fa", "Gd", ...: 2 5 5 1 3 1 1 1 3 ...
## $ Central.Air : Factor w/ 2 levels "N", "Y": 2 2 2 2 2 2 2 2 2 ...
## $ Electrical : Factor w/ 6 levels "", "FuseA", "FuseF", ...: 6 6 6 6 6 6 6 6 6 ...
## $ X1st.Flr.SF : int 1656 896 1329 2110 928 926 1338 1280 1616 1028 ...
## $ X2nd.Flr.SF : int 0 0 0 0 701 678 0 0 0 776 ...
## $ Low.Qual.Fin.SF: int 0 0 0 0 0 0 0 0 0 0 ...
## $ Gr.Liv.Area : int 1656 896 1329 2110 1629 1604 1338 1280 1616 1804 ...
## $ Bsmt.Full.Bath : num 1 0 0 1 0 0 1 0 1 0 ...
## $ Bsmt.Half.Bath : num 0 0 0 0 0 0 0 0 0 0 ...
## $ Full.Bath : int 1 1 1 2 2 2 2 2 2 2 ...
## $ Half.Bath : int 0 0 1 1 1 1 0 0 0 1 ...
## $ Bedroom.AbvGr : int 3 2 3 3 3 3 2 2 2 3 ...
## $ Kitchen.AbvGr : int 1 1 1 1 1 1 1 1 1 1 ...
## $ Kitchen.Qual : Factor w/ 5 levels "Ex", "Fa", "Gd", ...: 5 5 3 1 5 3 3 3 3 ...
## $ TotRms.AbvGrd : int 7 5 6 8 6 7 6 5 5 7 ...
## $ Functional : Factor w/ 8 levels "Maj1", "Maj2", ...: 8 8 8 8 8 8 8 8 8 ...
## $ Fireplaces : int 2 0 0 2 1 1 0 0 1 1 ...
## $ Fireplace.Qu : Factor w/ 6 levels "1", "Ex", "Fa", ...: 4 1 1 6 6 4 1 1 6 6 ...
## $ Garage.Type : Factor w/ 7 levels "1", "2Types", "Attchd", ...: 3 3 3 3 3 3 3 3 3 3 ...
## $ Garage.Yr.Blt : num 1960 1961 1958 1968 1997 ...
## $ Garage.Finish : Factor w/ 5 levels "", "1", "Fin", "RFn", ...: 3 5 5 3 3 3 3 4 4 3 ...
## $ Garage.Cars : num 2 1 1 2 2 2 2 2 2 2 ...
## $ Garage.Area : num 528 730 312 522 482 470 582 506 608 442 ...
## $ Garage.Qual : Factor w/ 7 levels "", "1", "Ex", "Fa", ...: 7 7 7 7 7 7 7 7 7 7 ...
## $ Garage.Cond : Factor w/ 7 levels "", "1", "Ex", "Fa", ...: 7 7 7 7 7 7 7 7 7 7 ...
## $ Paved.Drive : Factor w/ 3 levels "N", "P", "Y": 2 3 3 3 3 3 3 3 3 3 ...
## $ Wood.Deck.SF : int 210 140 393 0 212 360 0 0 237 140 ...
## $ Open.Porch.SF : int 62 0 36 0 34 36 0 82 152 60 ...
## $ Enclosed.Porch : int 0 0 0 0 0 0 170 0 0 0 ...
## $ X3Ssn.Porch : int 0 0 0 0 0 0 0 0 0 0 ...
## $ Screen.Porch : int 0 120 0 0 0 0 0 144 0 0 ...
## $ Pool.Area : int 0 0 0 0 0 0 0 0 0 0 ...
## $ Pool.QC : Factor w/ 5 levels "1", "Ex", "Fa", ...: 1 1 1 1 1 1 1 1 1 1 ...
## $ Fence : Factor w/ 5 levels "1", "GdPrv", "GdWo", ...: 1 4 1 1 4 1 1 1 1 1 ...
## $ Misc.Feature : Factor w/ 6 levels "1", "Elev", "Gar2", ...: 1 1 3 1 1 1 1 1 1 1 ...
## $ Misc.Val : int 0 0 12500 0 0 0 0 0 0 0 ...
## $ Mo.Sold : int 5 6 6 4 3 6 4 1 3 6 ...
## $ Yr.Sold : Factor w/ 5 levels "2006", "2007", ...: 5 5 5 5 5 5 5 5 5 5 ...
## $ Sale.Type : Factor w/ 10 levels "COD", "Con", "ConLD", ...: 10 10 10 10 10 10 10 10 10 10 ...
## $ Sale.Condition : Factor w/ 6 levels "Abnorml", "AdjLand", ...: 5 5 5 5 5 5 5 5 5 5 ...
## $ SalePrice : int 215000 105000 172000 244000 189900 195500 213500 191500 236500 189000 ...

```

First few examples:

```
housing_data$SalePrice[1:10]
```

```
## [1] 215000 105000 172000 244000 189900 195500 213500 191500 236500 189000
```

```
housing_data$Lot.Area[1:10]
```

```
## [1] 31770 11622 14267 11160 13830 9978 4920 5005 5389 7500
```

```
housing_data$Utilities[1:10]
```

```
## [1] AllPub AllPub AllPub AllPub AllPub AllPub AllPub AllPub AllPub AllPub  
## Levels: AllPub NoSeWa NoSewr
```

Dropping some of the columns that are not useful as a predictor for sale price.

```
# Dropping the order and PID, looks like this is just for record keeping
```

```
housing_data = subset(housing_data, select = -c(Order, PID))
```

```
# Removing lot fraontage area, since we have lot area. Frontage is measurement of the house start to th
```

```
housing_data = subset(housing_data, select = -c(Lot.Frontage))
```

```
# Removing alley because we have street data
```

```
housing_data = subset(housing_data, select = -c(Alley))
```

```
# removing one condition column and exterior
```

```
housing_data = subset(housing_data, select = -c(Condition.2, Exterior.2nd))
```

```
# updating NA's for some numeric variables to 0 where it make sense for calculation purpose
```

```
# housing_data$Garage.Area = ifelse(is.na(housing_data$Garage.Area), 0, housing_data$Garage.Area)
```

```
# housing_data$Fence = ifelse(is.na(housing_data$Fence), 0, housing_data$Fence)
```

Location of the house could play a big role in house price, but for the dataset that is used, it has total 28 neighbors as follow:

```
length(unique(housing_data[, "Neighborhood"]))
```

```
## [1] 28
```

We don't need this big list of dummy variable, so creating new variable as location based on some important factor

```
# first converting the some exterior variables to numeric and then using it:
```

```
levels(housing_data[, "Exter.Cond"])
```

```
## [1] "Ex" "Fa" "Gd" "Po" "TA"
```

```
levels(housing_data[, "Exter.Qual"])
```

```
## [1] "Ex" "Fa" "Gd" "TA"
```

```
levels(housing_data[, "Functional"])
```

```
## [1] "Maj1" "Maj2" "Min1" "Min2" "Mod" "Sal" "Sev" "Typ"
```

```
housing_data$Exter.Cond.Num = 5 - as.numeric(housing_data[, "Exter.Cond"])
housing_data$Exter.Qual.Num = 4 - as.numeric(housing_data[, "Exter.Qual"])
housing_data$Functional.Num = ifelse(housing_data[, "Functional"] == "Maj1", 8,
                                     ifelse(housing_data[, "Functional"] == "Maj2", 7,
                                             ifelse(housing_data[, "Functional"] == "Min1", 6,
                                                     ifelse(housing_data[, "Functional"] == "Min2", 5,
                                                             ifelse(housing_data[, "Functional"] == "Mod", 4,
                                                                     ifelse(housing_data[, "Functional"] == "Sal", 3,
                                                                             ifelse(housing_data[, "Functional"] == "Sev", 2,
                                                                                     ifelse(housing_data[, "Functional"] == "Typ", 1, 0))))))))
housing_data$Location = (housing_data[, "Overall.Qual"] / mean(housing_data[, "Overall.Qual"])) +
  (housing_data[, "Overall.Cond"] / mean(housing_data[, "Overall.Cond"])) +
  (housing_data[, "Exter.Cond.Num"] / mean(housing_data[, "Exter.Cond.Num"])) +
  (housing_data[, "Exter.Qual.Num"] / mean(housing_data[, "Exter.Qual.Num"])) +
  (housing_data[, "Functional.Num"] / mean(housing_data[, "Functional.Num"]))

summary(housing_data[, "Location"])
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  1.631   2.846   3.613   5.000   5.281   24.363
```

Converting ranked columns from factor to numeric

```
#Function to convert Qual/Cond values into a numeric scale
convert_rank= function(col) {
  col=replace(col,col=="Ex",values=5)
  col=replace(col,col=="Gd",values=4 )
  col=replace(col,col=="TA",values=3 )
  col=replace(col,col=="Fa",values=2 )
  col=replace(col,col=="Po",values=1 )
  col=replace(col,col=="",values=0 )
  as.numeric(col)
}
```

```
#Create Numeric Columns for the rank
#Convert Bsmt Columns
housing_data$Bsmt.Qual.Num=convert_rank(as.character(housing_data$Bsmt.Qual))
housing_data$Bsmt.Cond.Num=convert_rank(as.character(housing_data$Bsmt.Cond))
#Convert Garage Columns
housing_data$Garage.Cond.Num=convert_rank(as.character(housing_data$Garage.Cond))
```

```
#housing_data$Garage.Qual.Num=convert_rank(as.character(housing_data$Garage.Qual))
#Convert Exterior Columns
housing_data$Exter.Cond.Num=convert_rank(as.character(housing_data$Exter.Cond))
housing_data$Exter.Qual.Num=convert_rank(as.character(housing_data$Exter.Qual))
#Test equality after rank conversion
Ex=housing_data$Exter.Cond=="Gd"
test=housing_data$Exter.Cond.Num==4
#all.equal(Ex,test)
```

```
#Now that data cleaning is done, we split Test and Train data
set.seed(720)
ames_trn_idx = sample(nrow(housing_data), size = trunc(0.80 * nrow(housing_data)))
ames_trn_data = housing_data[ames_trn_idx, ]
ames_tst_data = housing_data[-ames_trn_idx, ]
```

```
View(housing_data)
```

Collinearity and correlation analysis

Subsetting all the numeric elements of the dataset for collinearity and correlation analysis:

```
n_idx = unlist(lapply(housing_data, is.numeric))
all_numeric_housing_data = housing_data[, n_idx]
numeric_housing_data = subset(all_numeric_housing_data, select = -c(SalePrice))
```

```
#str(numeric_housing_data)
```

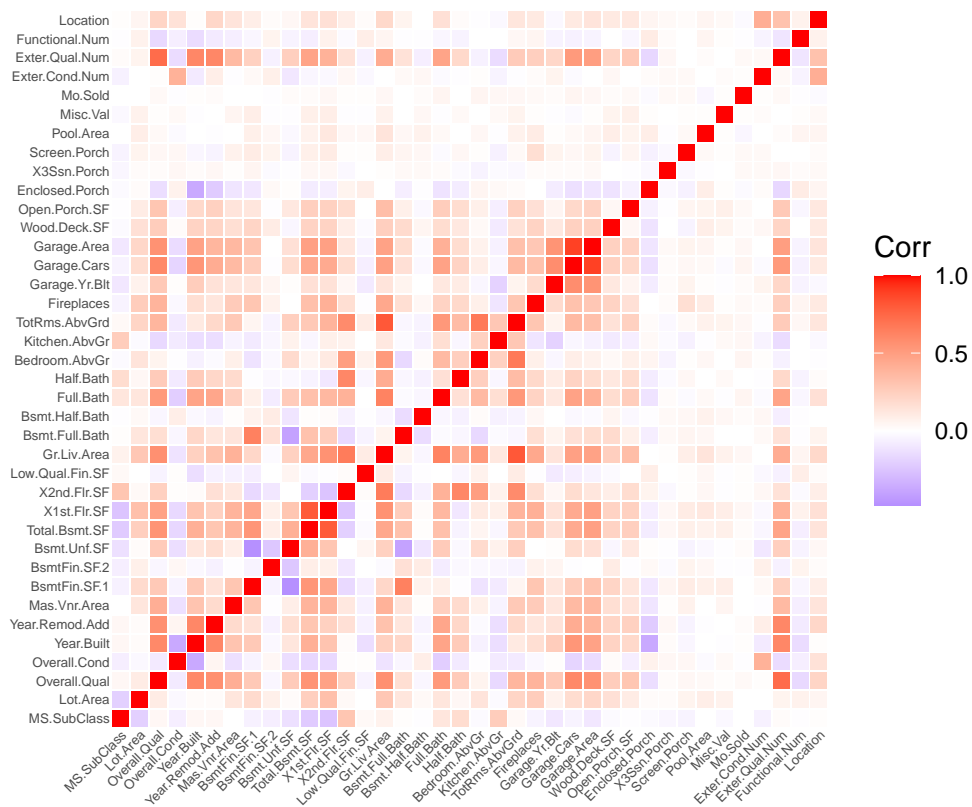
```
# MUST specify use = "complete.obs" argument to ignore NA's in dataset
corrs = round(cor(numeric_housing_data, use="complete.obs"), 2)
```

```
# some possible correlation plots?
```

```
#corrplot(corrs, method="number")
#ggcorrplot(corrs, lab_size = 0.1)
```

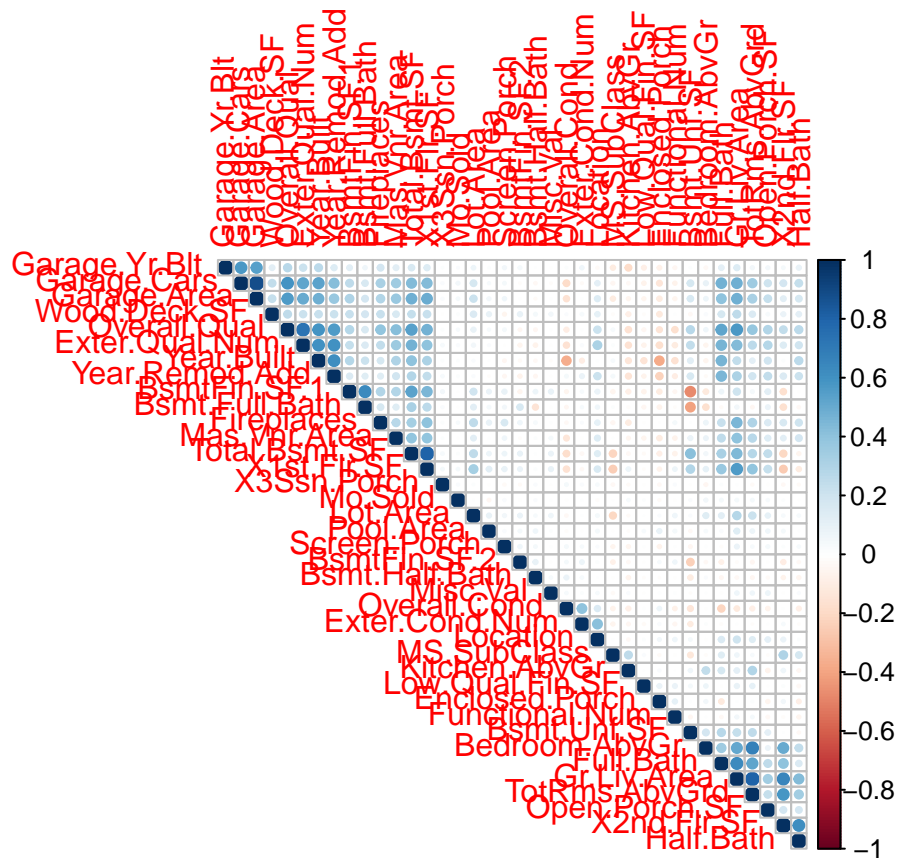
```
#corrs
```

```
ggplot(melt(corrs), aes(Var1, Var2, fill=value)) +
  geom_tile(height=0.9, width=0.9) +
  scale_fill_gradient2(low="blue", mid="white", high="red") +
  theme_minimal() +
  coord_equal() +
  labs(x="", y="", fill="Corr") +
  theme(axis.text.x=element_text(size=5, angle=45, vjust=1, hjust=1,
                                margin=margin(-3,0,0,0)),
        axis.text.y=element_text(size=5, margin=margin(0,-3,0,0)),
        panel.grid.major=element_blank())
```



Taking a closer look at some of the most correlated predictors:

```
res = rcorr(as.matrix(numeric_housing_data));
corrplot(res$r, type="upper", order="hclust", sig.level = 0.5, insig = "blank");
```



```
#cor.plot(numeric_housing_data)

# pairs(numeric_housing_data, col = "dodgerblue")
```

While some of these high correlation measures are to be expected, such as house year built along garage year built, we can also see some non-trivial patterns start to emerge from the more continuous numeric predictors.

Models (VP)

```
#Functions to evaluate models
library(lmtest)

get_bp_decision = function(model, alpha) {
  decide = unname(bptest(model)$p.value < alpha)
  ifelse(decide, "Reject", "Fail to Reject")
}

get_sw_decision = function(model, alpha) {
  decide = unname(shapiro.test(resid(model))$p.value < alpha)
  ifelse(decide, "Reject", "Fail to Reject")
}

get_num_params = function(model) {
```



```

length(coef(model))
}

get_loocv_rmse = function(model) {
  sqrt(mean((resid(model) / (1 - hatvalues(model))) ^ 2, na.rm=TRUE))
}

get_adj_r2 = function(model) {
  summary(model)$adj.r.squared
}

compute_rmse = function(model) {
  sqrt(mean(resid(model)^2))
}

get_avg_per_error = function(model, newdata=ames_tst_data) {
  predicted = predict(model, newdata = newdata)
  n = nrow(newdata)
  error = abs(newdata$SalePrice - predicted)
  avg_per_error = ((1 / n) * (sum(error / predicted))) * 100
  avg_per_error
}

pred_graph= function(model, main="Predicted Vs. Actual") {
  predicted = predict(model, newdata = ames_tst_data)
  plot(
    ames_tst_data$SalePrice,
    predicted,
    ylab = "Predicted Price ($)",
    xlab = "Actual Price ($)",
    col = "blue",
    main = main
  )
  abline(a = 0, b = 1, lwd = 2)
}

```

Make sure that location is correlated to saleprice and positive:

```
cor(housing_data[,c("Location", "SalePrice")])
```

```
##           Location SalePrice
## Location  1.0000000 0.2533568
## SalePrice 0.2533568 1.0000000
```

It has weak but a positive relation of with saleprice.

Taking most relevant predictors that are logically be useful to build saleprice model with existing knowledge of real estate.

```
# this model removes some of the qualities variable because it has some condition of those variables pr
# Such as, keeping Bsmt.Cond and removing Bsmt.Qual (because both factor has almost same levels)
```

```
saleprice_full_model_selected = lm(SalePrice ~ MS.Zoning + Lot.Area + Street + Lot.Shape + Land.Contour
```

```
saleprice_additive_model = lm(SalePrice ~ ., data = ames_trn_data)
```

```
saleprice_selected_backward_aic = step(saleprice_full_model_selected, direction = "backward", trace = 0)
```

```
# check the number of predictors in the model
```

```
length(coef(saleprice_full_model_selected)) - 1
```

```
## [1] 112
```

```
length(coef(saleprice_selected_backward_aic)) - 1
```

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

Forward BIC Model - Full Additive Scope

```
#Create BIC Forward model with Scope as full Additive
```

```
saleprice_additive_model = lm(SalePrice ~ ., data = ames_trn_data)
```

```
form_add=update(formula(saleprice_additive_model), ~.-Misc.Feature)
```

```
start=lm(SalePrice ~ 1, data=ames_trn_data)
```

```
n=nrow(ames_trn_data)
```

```
bic_for_mod=step(start, direction = "forward", scope = form_add ,k=log(n), trace=0)
```

Taking most relevant predictors that are logically be useful to build saleprice model with existing knowledge of real estate.

```
# this model removes some of the qualities variable because it has some condition of those variables pr
# Such as, keeping Bsmt.Cond and removing Bsmt.Qual (because both factor has almost same levels)
```

```
saleprice_full_model_selected = lm(SalePrice ~ MS.Zoning + Lot.Area + Street + Lot.Shape + Land.Contour
```

```
saleprice_selected_backward_aic = step(saleprice_full_model_selected, direction = "backward", trace = 0)
```

```
# check the number of predictors in the model
```

```
length(coef(saleprice_full_model_selected)) - 1
```

```
## [1] 112
```

```
length(coef(saleprice_selected_backward_aic)) - 1
```

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

Testing the built model

```
get_loocv_rmse(saleprice_additive_model)
```

```
## [1] Inf
```

```
get_loocv_rmse(saleprice_full_model_selected)
```

```
## [1] Inf
```

```
get_loocv_rmse(saleprice_selected_backward_aic)
```

```
## [1] Inf
```

Even though removing so many variables, the model is still very huge, so, removing some of the variables that are may be corelated. Also, removing the variables that seems related based on the common real estate knowledge.

```
saleprice_full_model_selected_reduced = lm(SalePrice ~ MS.Zoning + Lot.Area + Street + Lot.Shape + Util.
```

```
saleprice_selected_reduced_backward_aic = step(saleprice_full_model_selected_reduced, direction = "back
```

```
summary(saleprice_selected_reduced_backward_aic)
```

```
##
```

```
## Call:
```

```
## lm(formula = SalePrice ~ MS.Zoning + Lot.Area + Street + Lot.Shape +  
##   Utilities + Land.Slope + House.Style + Year.Built + Foundation +  
##   Location + Central.Air + Gr.Liv.Area + Bedroom.AbvGr + Kitchen.AbvGr +  
##   Kitchen.Qual + TotRms.AbvGrd + Fireplaces + Garage.Area +  
##   Wood.Deck.SF + Pool.Area + Misc.Val, data = ames_trn_data)
```

```
##
```

```
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max  
## -421605  -17180   -1053   14176  243038
```

```
##
```

```
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)  
## (Intercept)   -9.694e+05  9.261e+04 -10.467  < 2e-16 ***  
## MS.ZoningC (all)  6.768e+04  2.603e+04   2.600  0.009383 **  
## MS.ZoningFV      8.741e+04  2.500e+04   3.496  0.000481 ***  
## MS.ZoningI (all)  1.349e+05  5.641e+04   2.391  0.016884 *  
## MS.ZoningRH      7.669e+04  2.579e+04   2.973  0.002979 **  
## MS.ZoningRL      8.386e+04  2.472e+04   3.392  0.000704 ***  
## MS.ZoningRM      7.832e+04  2.482e+04   3.155  0.001624 **  
## Lot.Area        4.988e-01  1.407e-01   3.544  0.000403 ***  
## StreetPave      2.716e+04  1.306e+04   2.080  0.037669 *  
## Lot.ShapeIR2     3.120e+03  4.707e+03   0.663  0.507509  
## Lot.ShapeIR3    -4.821e+04  9.434e+03  -5.110  3.49e-07 ***  
## Lot.ShapeReg    -4.875e+03  1.651e+03  -2.953  0.003184 **  
## UtilitiesNoSeWa -2.717e+04  3.459e+04  -0.785  0.432301
```

```
## UtilitiesNoSewr -6.830e+04 3.452e+04 -1.978 0.048014 *
## Land.SlopeMod 1.223e+04 3.616e+03 3.381 0.000734 ***
## Land.SlopeSev -1.079e+04 1.215e+04 -0.888 0.374795
## House.Style1.5Unf 2.419e+04 8.506e+03 2.843 0.004503 **
## House.Style1Story 1.529e+04 2.808e+03 5.444 5.75e-08 ***
## House.Style2.5Fin -2.182e+04 1.461e+04 -1.493 0.135610
## House.Style2.5Unf 1.829e+04 9.071e+03 2.016 0.043891 *
## House.Style2Story -4.223e+03 2.809e+03 -1.503 0.132860
## House.StyleSFoyer 1.586e+04 5.190e+03 3.056 0.002268 **
## House.StyleSLvl 2.428e+03 4.369e+03 0.556 0.578539
## Year.Built 4.945e+02 4.499e+01 10.990 < 2e-16 ***
## FoundationCBlock -7.432e+03 2.981e+03 -2.493 0.012738 *
## FoundationPConc 4.393e+03 3.521e+03 1.248 0.212278
## FoundationSlab -3.177e+04 6.050e+03 -5.252 1.65e-07 ***
## FoundationStone 2.067e+04 1.146e+04 1.803 0.071454 .
## FoundationWood -1.340e+04 1.589e+04 -0.843 0.399075
## Location 1.447e+03 2.476e+02 5.842 5.90e-09 ***
## Central.AirY 5.839e+03 3.401e+03 1.717 0.086146 .
## Gr.Liv.Area 7.258e+01 3.163e+00 22.947 < 2e-16 ***
## Bedroom.AbvGr -4.733e+03 1.302e+03 -3.635 0.000284 ***
## Kitchen.AbvGr -2.437e+04 3.837e+03 -6.351 2.57e-10 ***
## Kitchen.QualFa -7.703e+04 5.821e+03 -13.233 < 2e-16 ***
## Kitchen.QualGd -6.211e+04 3.187e+03 -19.492 < 2e-16 ***
## Kitchen.QualPo -5.865e+04 3.482e+04 -1.685 0.092215 .
## Kitchen.QualTA -7.438e+04 3.654e+03 -20.355 < 2e-16 ***
## TotRms.AbvGrd 1.470e+03 9.458e+02 1.555 0.120144
## Fireplaces 1.076e+04 1.329e+03 8.096 9.13e-16 ***
## Garage.Area 4.683e+01 4.433e+00 10.564 < 2e-16 ***
## Wood.Deck.SF 2.113e+01 6.004e+00 3.519 0.000442 ***
## Pool.Area -1.013e+02 2.015e+01 -5.029 5.32e-07 ***
## Misc.Val -8.072e+00 1.153e+00 -6.998 3.38e-12 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 34320 on 2300 degrees of freedom
## Multiple R-squared: 0.8127, Adjusted R-squared: 0.8092
## F-statistic: 232.1 on 43 and 2300 DF, p-value: < 2.2e-16
```

```
# temp
length(coef(saleprice_full_model_selected_reduced)) - 1
```

```
## [1] 65
```

```
length(coef(saleprice_selected_reduced_backward_aic)) - 1
```

```
## [1] 43
```

Testing the built model

```
get_loocv_rmse(saleprice_additive_model)
```

```
## [1] Inf

get_loocv_rmse(saleprice_full_model_selected_reduced)

## [1] Inf

get_loocv_rmse(saleprice_selected_reduced_backward_aic)

## [1] Inf

anova(saleprice_full_model_selected_reduced, saleprice_selected_reduced_backward_aic)

## Analysis of Variance Table
##
## Model 1: SalePrice ~ MS.Zoning + Lot.Area + Street + Lot.Shape + Utilities +
##   Land.Slope + House.Style + Year.Built + Foundation + Location +
##   Heating + Central.Air + Electrical + Gr.Liv.Area + Full.Bath +
##   Bedroom.AbvGr + Kitchen.AbvGr + Kitchen.Qual + TotRms.AbvGrd +
##   Fireplaces + Garage.Area + Paved.Drive + Wood.Deck.SF + Open.Porch.SF +
##   Pool.Area + Fence + Misc.Val + Yr.Sold
## Model 2: SalePrice ~ MS.Zoning + Lot.Area + Street + Lot.Shape + Utilities +
##   Land.Slope + House.Style + Year.Built + Foundation + Location +
##   Central.Air + Gr.Liv.Area + Bedroom.AbvGr + Kitchen.AbvGr +
##   Kitchen.Qual + TotRms.AbvGrd + Fireplaces + Garage.Area +
##   Wood.Deck.SF + Pool.Area + Misc.Val
##   Res.Df      RSS   Df Sum of Sq    F Pr(>F)
## 1    2278 2.6864e+12
## 2    2300 2.7090e+12 -22 -2.2603e+10 0.8712 0.6348
```

Variable transformations

For this section, we'll begin by visualizing the relationships between the house sale prices and some of our predictors. By doing this, we're looking to gain some insight into potential variable transformations we could implement in order to improve the performance of our models.

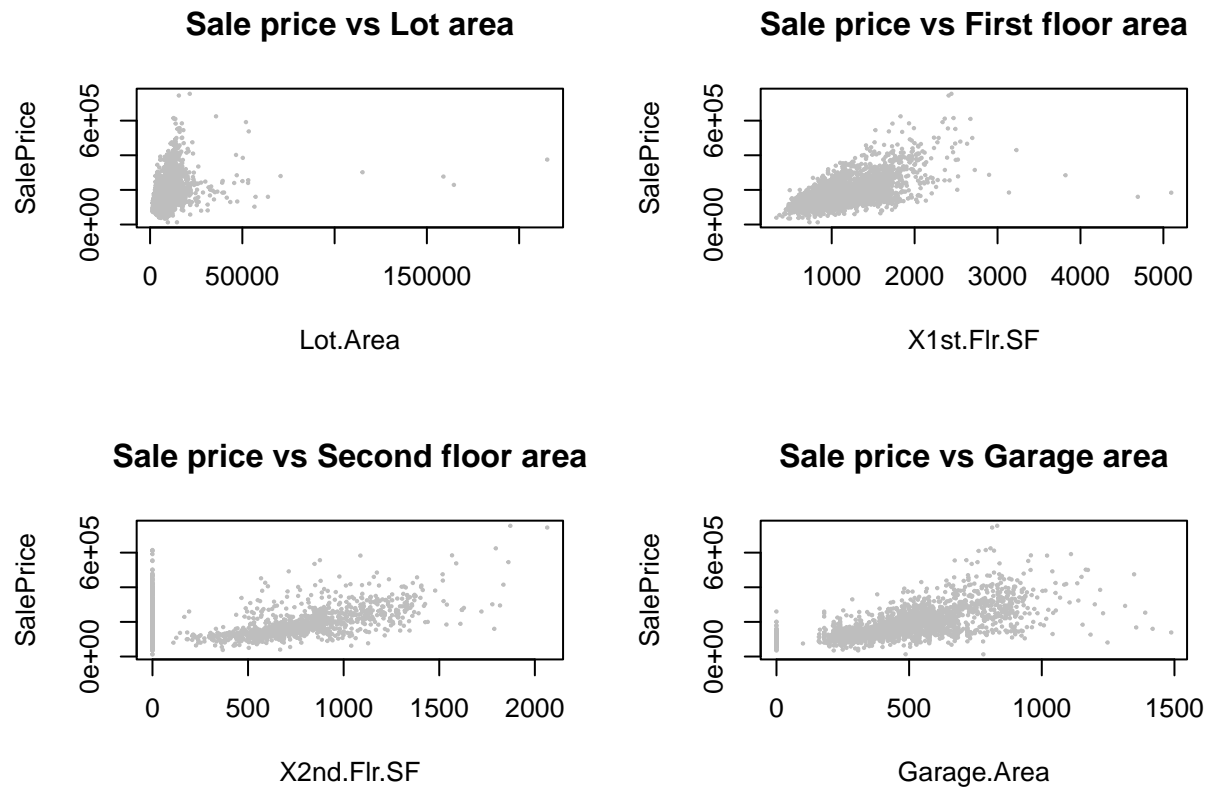
```
par(mfrow = c(2, 2))

plot(SalePrice ~ Lot.Area, data = housing_data, col = "grey", pch = 20, cex = 0.3,
     main = "Sale price vs Lot area")

plot(SalePrice ~ X1st.Flr.SF, data = housing_data, col = "grey", pch = 20, cex = 0.3,
     main = "Sale price vs First floor area")

plot(SalePrice ~ X2nd.Flr.SF, data = housing_data, col = "grey", pch = 20, cex = 0.3,
     main = "Sale price vs Second floor area")

plot(SalePrice ~ Garage.Area, data = housing_data, col = "grey", pch = 20, cex = 0.3,
     main = "Sale price vs Garage area")
```



```
par(mfrow = c(2, 2))

plot(SalePrice ~ Total.Bsmt.SF, data = housing_data, col = "grey", pch = 20, cex = 0.3,
     main = "Sale price vs Basement area")

plot(SalePrice ~ Overall.Qual, data = housing_data, col = "grey", pch = 20, cex = 0.3,
     main = "Sale price vs Overall quality")

plot(SalePrice ~ Year.Built, data = housing_data, col = "grey", pch = 20, cex = 0.3,
     main = "Sale price vs Year built")

plot(SalePrice ~ Overall.Cond, data = housing_data, col = "grey", pch = 20, cex = 0.3,
     main = "Sale price vs Overall condition")
```



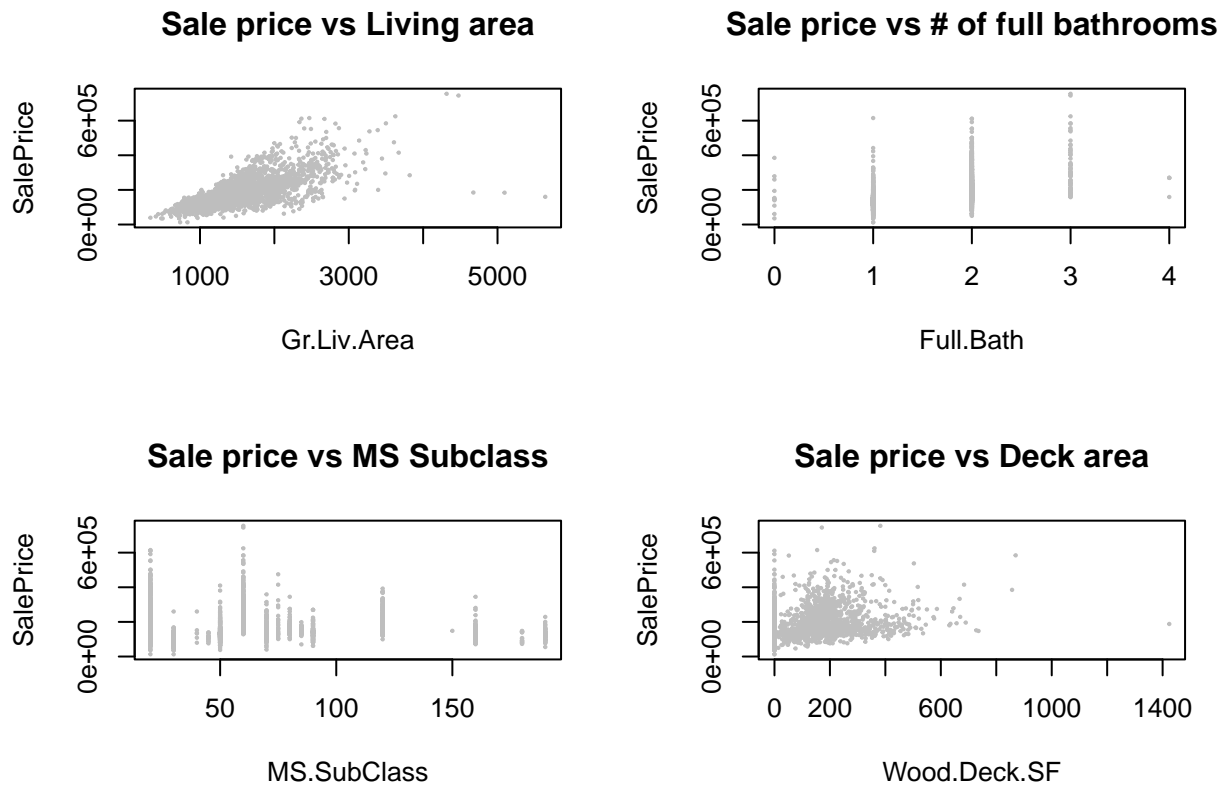
```
par(mfrow = c(2, 2))

plot(SalePrice ~ Gr.Liv.Area, data = housing_data, col = "grey", pch = 20, cex = 0.3,
     main = "Sale price vs Living area")

plot(SalePrice ~ Full.Bath, data = housing_data, col = "grey", pch = 20, cex = 0.3,
     main = "Sale price vs # of full bathrooms")

plot(SalePrice ~ MS.SubClass, data = housing_data, col = "grey", pch = 20, cex = 0.3,
     main = "Sale price vs MS Subclass")

plot(SalePrice ~ Wood.Deck.SF, data = housing_data, col = "grey", pch = 20, cex = 0.3,
     main = "Sale price vs Deck area")
```



```
par(mfrow = c(2, 2))

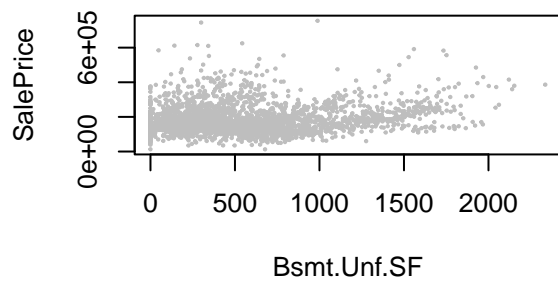
plot(SalePrice ~ Bsmt.Unf.SF, data = housing_data, col = "grey", pch = 20, cex = 0.3,
     main = "Sale price vs Basement unfinished area")

plot(SalePrice ~ BsmtFin.SF.1, data = housing_data, col = "grey", pch = 20, cex = 0.3,
     main = "Sale price vs Basement finished area")

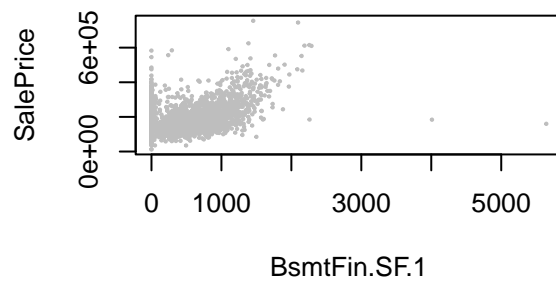
plot(SalePrice ~ Mo.Sold, data = housing_data, col = "grey", pch = 20, cex = 0.3,
     main = "Sale price vs Month sold")

plot(SalePrice ~ TotRms.AbvGrd, data = housing_data, col = "grey", pch = 20, cex = 0.3,
     main = "Sale price vs Total rooms above ground")
```

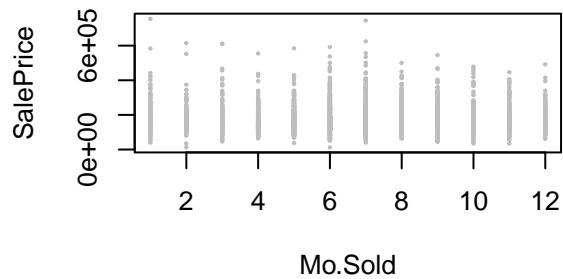

Sale price vs Basement unfinished area



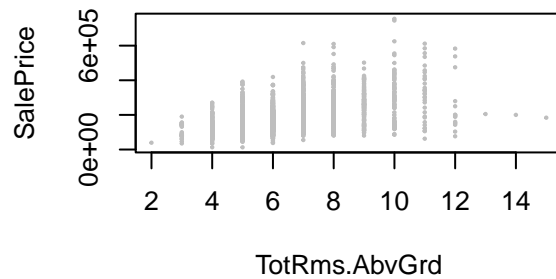
Sale price vs Basement finished area



Sale price vs Month sold

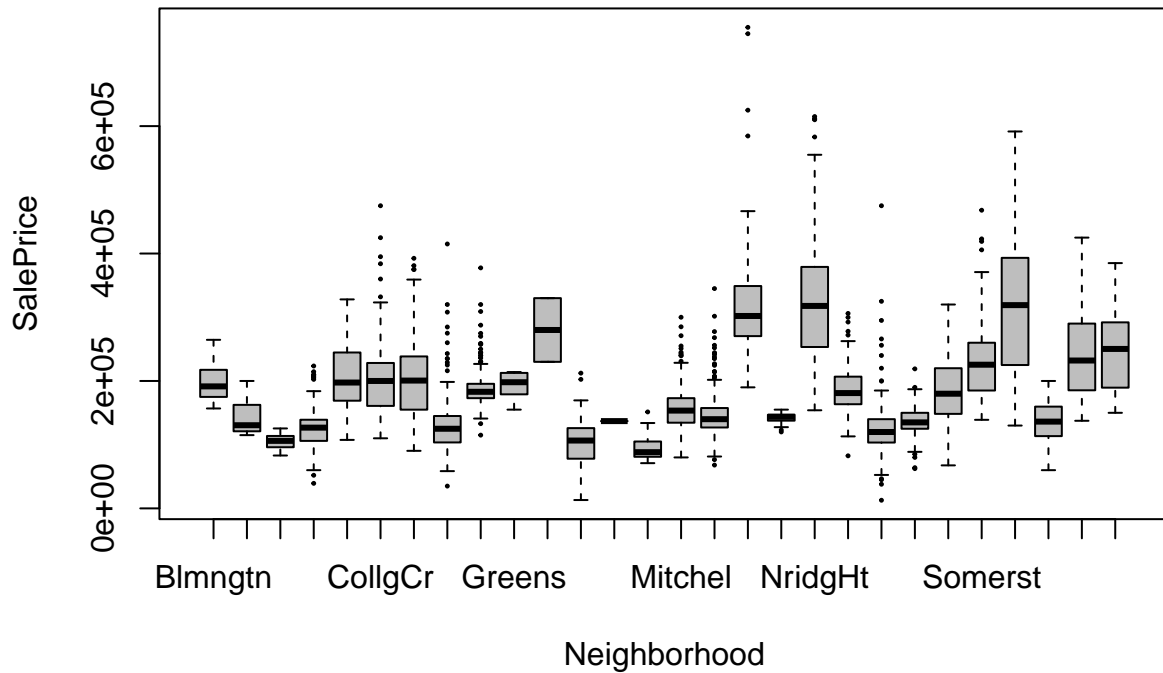


Sale price vs Total rooms above ground



```
plot(SalePrice ~ Neighborhood, data = housing_data, col = "grey", pch = 20, cex = 0.3,
     main = "Sale price vs Neighborhood")
```

Sale price vs Neighborhood



```
transformed_model_old = lm(SalePrice ~ MS.Zoning + log(Lot.Area) + Street + Lot.Shape + Land.Slope + Hor
```

```
transformed_model = lm(SalePrice ~ MS.Zoning + log(Lot.Area) + Street + I(Year.Built^2) + Kitchen.AbvGr
```

```
get_loocv_rmse(saleprice_selected_reduced_backward_aic)
```

```
## [1] Inf
```

```
get_loocv_rmse(transformed_model_old)
```

```
## [1] Inf
```

```
get_loocv_rmse(transformed_model)
```

```
## [1] 30360.39
```

```
sqrt(mean((ames_trn_data$SalePrice - fitted(saleprice_selected_reduced_backward_aic)) ^ 2))
```

```
## [1] 33995.86
```

```
sqrt(mean((ames_trn_data$SalePrice - fitted(transformed_model_old)) ^ 2))
```

```
## [1] 29437.04
```

```
sqrt(mean((ames_trn_data$SalePrice - fitted(transformed_model)) ^ 2))
```

```
## [1] 28706.19
```

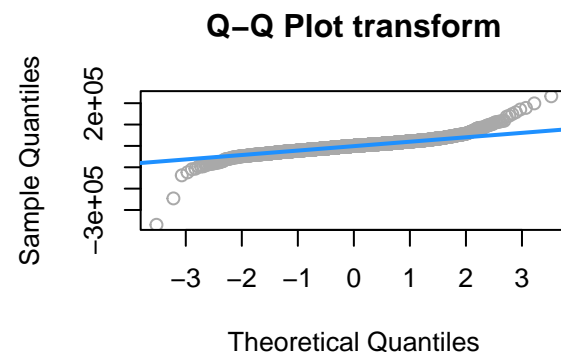
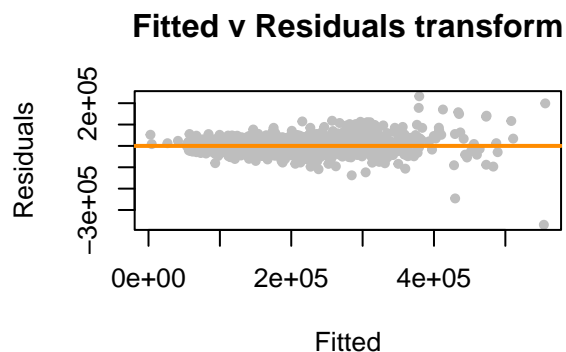
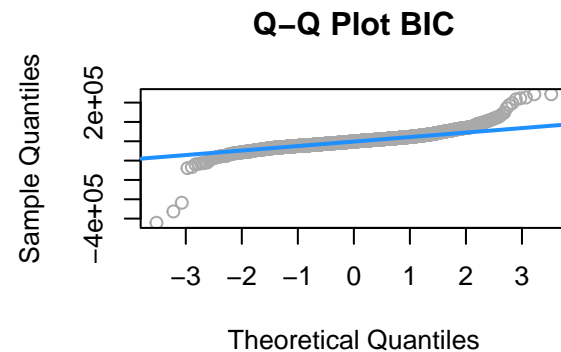
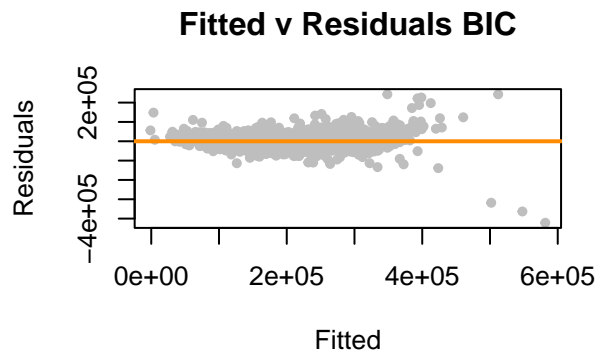
```
par(mfrow = c(2,2))
```

```
plot(fitted(saleprice_selected_reduced_backward_aic), resid(saleprice_selected_reduced_backward_aic), col = "grey",  
      xlab = "Fitted", ylab = "Residuals", main = "Fitted v Residuals BIC")  
abline(h = 0, col = "darkorange", lwd = 2)
```

```
qqnorm(resid(saleprice_selected_reduced_backward_aic), main = "Q-Q Plot BIC", col = "darkgrey")  
qqline(resid(saleprice_selected_reduced_backward_aic), col = "dodgerblue", lwd = 2)
```

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

```
qqnorm(resid(transformed_model), main = "Q-Q Plot transform", col = "darkgrey")  
qqline(resid(transformed_model), col = "dodgerblue", lwd = 2)
```



After adding some polynomial and logarithmic kernel transformations to applicable predictor variables, we can see some noticeable improvement in both RMSE and LOOCV-RMSE metrics when comparing to the reduced backward-BIC model from earlier.

```
formula(bic_for_mod)

## SalePrice ~ Overall.Qual + Gr.Liv.Area + Neighborhood + Bsmt.Qual +
##   BsmtFin.SF.1 + Roof.Mat1 + MS.SubClass + Bsmt.Exposure +
##   Kitchen.Qual + Overall.Cond + Garage.Area + Misc.Val + Year.Built +
##   Exter.Qual + Lot.Area + Screen.Porch + Fireplaces + Sale.Condition +
##   Functional.Num + Bsmt.Full.Bath + Total.Bsmt.SF + Bldg.Type +
##   Full.Bath + Mas.Vnr.Area + X2nd.Flr.SF

transformed_bic_mod=lm(SalePrice ~ Overall.Qual + Gr.Liv.Area + Neighborhood + Bsmt.Qual +
  BsmtFin.SF.1 + Roof.Mat1 + MS.SubClass + Bsmt.Exposure +
  Kitchen.Qual + Overall.Cond + I(Garage.Area^2) + sqrt(Misc.Val) + I(Year.Built^2) +
  Exter.Qual + log(Lot.Area) + Screen.Porch + (Fireplaces) + Sale.Condition +
  Functional.Num + Bsmt.Full.Bath + Total.Bsmt.SF + Bldg.Type +
  Full.Bath + Mas.Vnr.Area + X2nd.Flr.SF,data=ames_trn_data)
```

Model Evaluations

```
library(knitr)
models=list(transformed_model,saleprice_full_model_selected,saleprice_full_model_selected_reduced,saleprice_selected_backward_aic,saleprice_selected_reduced_backward_aic,bic_for_mod,transformed_bic_mod)

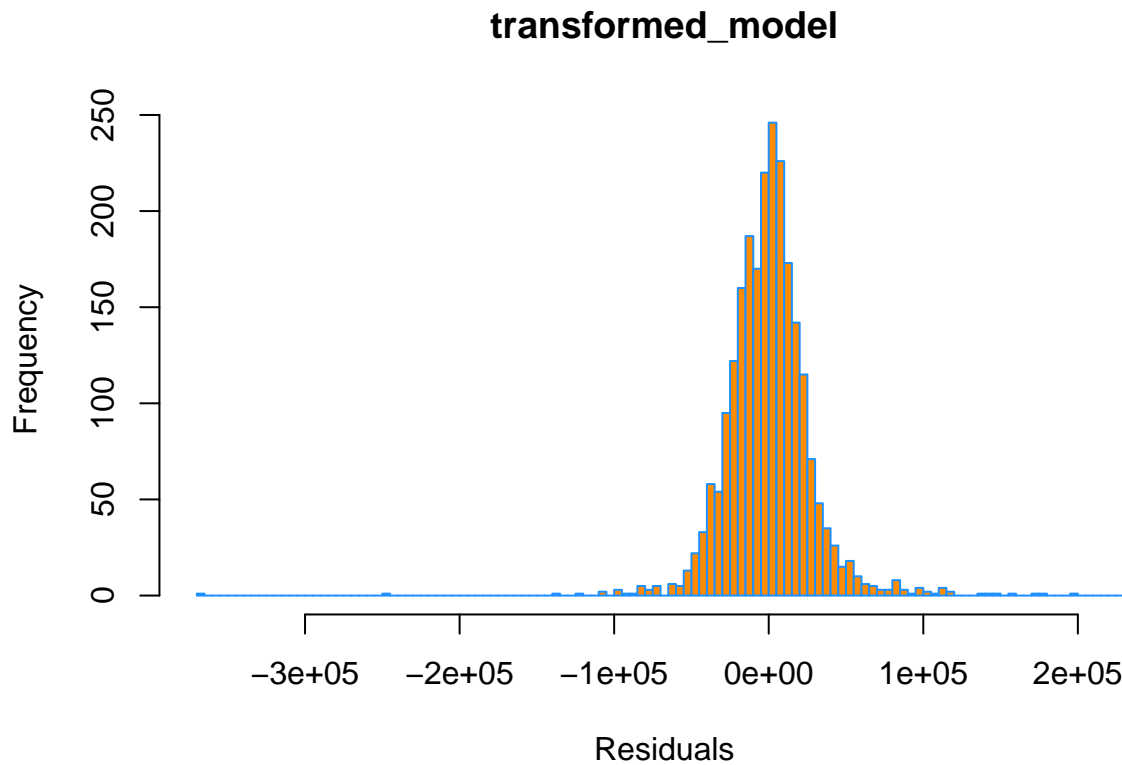
RMSE =lapply(lapply(models, compute_rmse), round, 2)
LOOCV_RMSE= lapply(lapply(models,get_loocv_rmse), round, 2)
AVG_ERROR= lapply(lapply(models,get_avg_per_error), round, 2)
ADJ_R2= lapply(lapply(models,get_adj_r2), round, 2)
BP_TEST= lapply(models,get_bp_decision,alpha=.90)
SW_TEST= lapply(models,get_sw_decision,alpha=.90)

Models=c("transformed_model","saleprice_full_model_selected","saleprice_full_model_selected_reduced","saleprice_selected_backward_aic","saleprice_selected_reduced_backward_aic","bic_for_mod","transformed_bic_mod")
evaluation=cbind(Models,RMSE,LOOCV_RMSE,AVG_ERROR,ADJ_R2,BP_TEST,SW_TEST)
kable(evaluation)
```

Models	RMSE	LOOCV_RMSE	AVG_ERROR	ADJ_R2	BP_TEST	SW_TEST
transformed_model	28706.19	30360.39	11.72	0.86	Reject	Reject
saleprice_full_model_selected	31795.88	Inf	18.34	0.83	Reject	Reject
saleprice_full_model_selected_reduced	33853.74	Inf	14	0.81	Reject	Reject
saleprice_selected_backward_aic	31964.79	Inf	13.85	0.83	Reject	Reject
saleprice_selected_reduced_backward_aic	33995.86	Inf	13.42	0.81	Reject	Reject
bic_for_mod	24031.77	Inf	9.68	0.9	Reject	Reject
transformed_bic_mod	24021.84	Inf	9.51	0.9	Reject	Reject

```
hist(resid(transformed_model),
     xlab = "Residuals",
     main = "transformed_model",
```

```
col    = "darkorange",
border = "dodgerblue",
breaks = 100)
```



```
shapiro.test(resid(transformed_model))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(transformed_model)
## W = 0.87315, p-value < 2.2e-16
```

```
ks.test(resid(transformed_model), "pnorm")
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  resid(transformed_model)
## D = 0.50128, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

```
shapiro.test(rnorm(499, mean=0, sd=1000))
```

```
##
## Shapiro-Wilk normality test
##
## data:  rnorm(499, mean = 0, sd = 1000)
## W = 0.99835, p-value = 0.9218

par(mfrow = c(1,2))
pred_graph(transformed_model,main="Transformed")
pred_graph(transformed_bic_mod,main="Transformed BIC")
```

