An examination of trends in Ames, Iowa housing prices

Team: Random85

2023-08-04

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 Kaggle. 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 elimate 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.

Methods:

The Dataset

```
library(faraway)
library(dplyr)
library(psych)
library(corrplot)
library(ggplot2)
library(corrplot)
library(lares)
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 52712715
0 527145080 527146030 527162130 ...
##  $ 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 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 2 ...
##  $ Alley          : Factor w/ 3 levels "1","Grvl","Pave": 1 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 3 ...
##  $ Condition.2    : Factor w/ 8 levels "Artery","Feedr",..: 3 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 8 ...
##  $ BsmtFin.SF.2   : num  0 144 0 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 2 ...
##  $ Heating.QC     : Factor w/ 5 levels "Ex","Fa","Gd",..: 2 5 5 1 3 1 1 1 1 3 ...
##  $ Central.Air    : Factor w/ 2 levels "N","Y": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Electrical     : Factor w/ 6 levels "","FuseA","FuseF",..: 6 6 6 6 6 6 6 6 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 ...
##  $ 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 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 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 ...
##  $ 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 18900
0 ...
```

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
```

Data Cleaning

```
# 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 star
t to the street. It has way more null values
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)
```

Creating Dummy variable

```
# 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 s
ome important factor
```

```
#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)
}

housing_data$Functional.Num = ifelse(housing_data[,"Functional"] == "Typ", 8,
                                     ifelse(housing_data[,"Functional"] == "Min1", 7,
                                            ifelse(housing_data[,"Functional"] == "Min2", 6,
                                                   ifelse(housing_data[,"Functional"] == "Mod",
5,
                                                          ifelse(housing_data[,"Functional"] ==
"Maj1", 4,
                                                                 ifelse(housing_data[,"Functiona
l"] == "Maj2", 3,
                                                                        ifelse(housing_data[,"Fu
nctional"] == "Sev", 2,
                                                                               ifelse(housing_da
ta[,"Functional"] == "Sal", 1, 0)))))))

#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
```

```
# creating new Location variable
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.Nu
m"])) +
                        (housing_data[,"Exter.Qual.Num"] / mean(housing_data[,"Exter.Qual.Nu
m"])) +
                        (housing_data[,"Functional.Num"] / mean(housing_data[,"Functional.Nu
m"]))

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

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.090   4.758   5.052   5.000   5.282   7.370
```

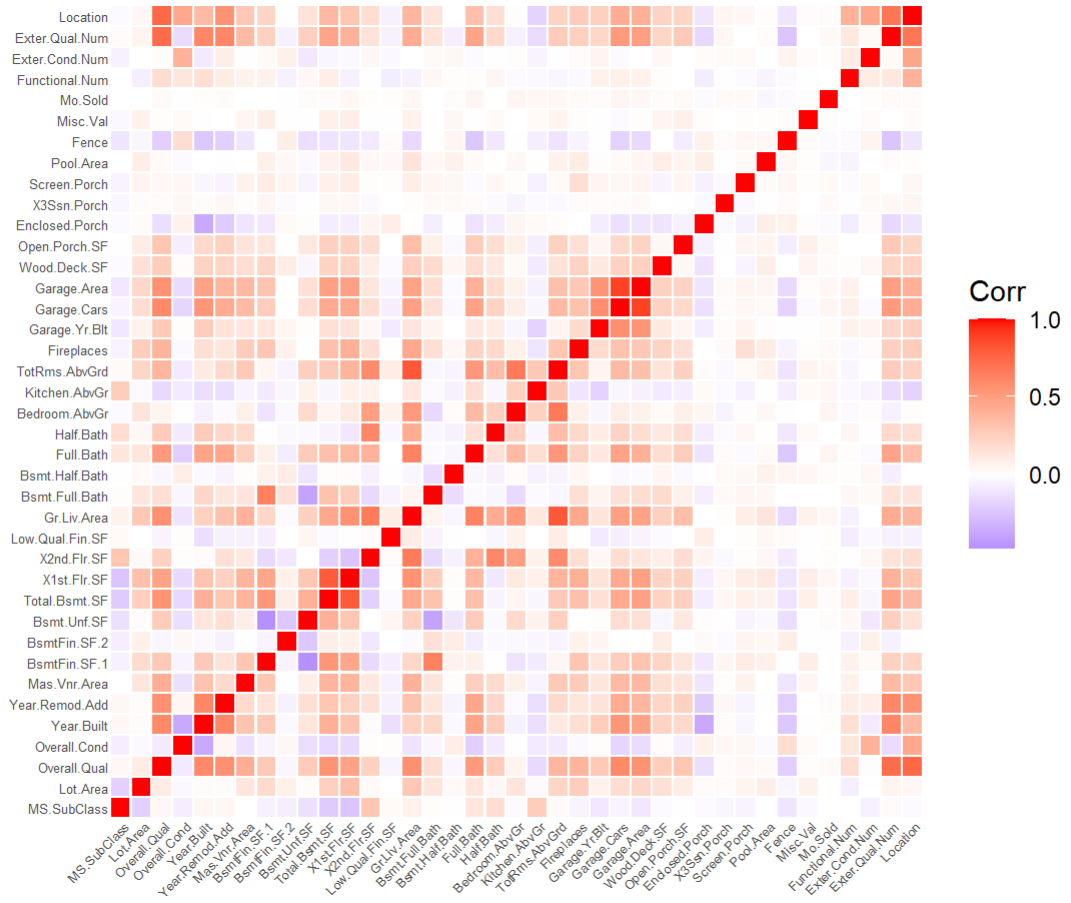
```
# Make sure that location is corelated to saleprice and positive:
cor(housing_data[,c("Location","SalePrice")])
```

```
##            Location SalePrice
## Location  1.0000000 0.6309229
## SalePrice 0.6309229 1.0000000
```

```
#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, ]
```

Studying Collinearity and correlation between variable to come up with starting model

```
# Subsetting all the numeric elements of the dataset for collinearity and correlation analysis:
n_idxs = unlist(lapply(housing_data, is.numeric))
all_numeric_housing_data = housing_data[, n_idxs]
numeric_housing_data = subset(all_numeric_housing_data, select = -c(SalePrice))

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

#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())
```



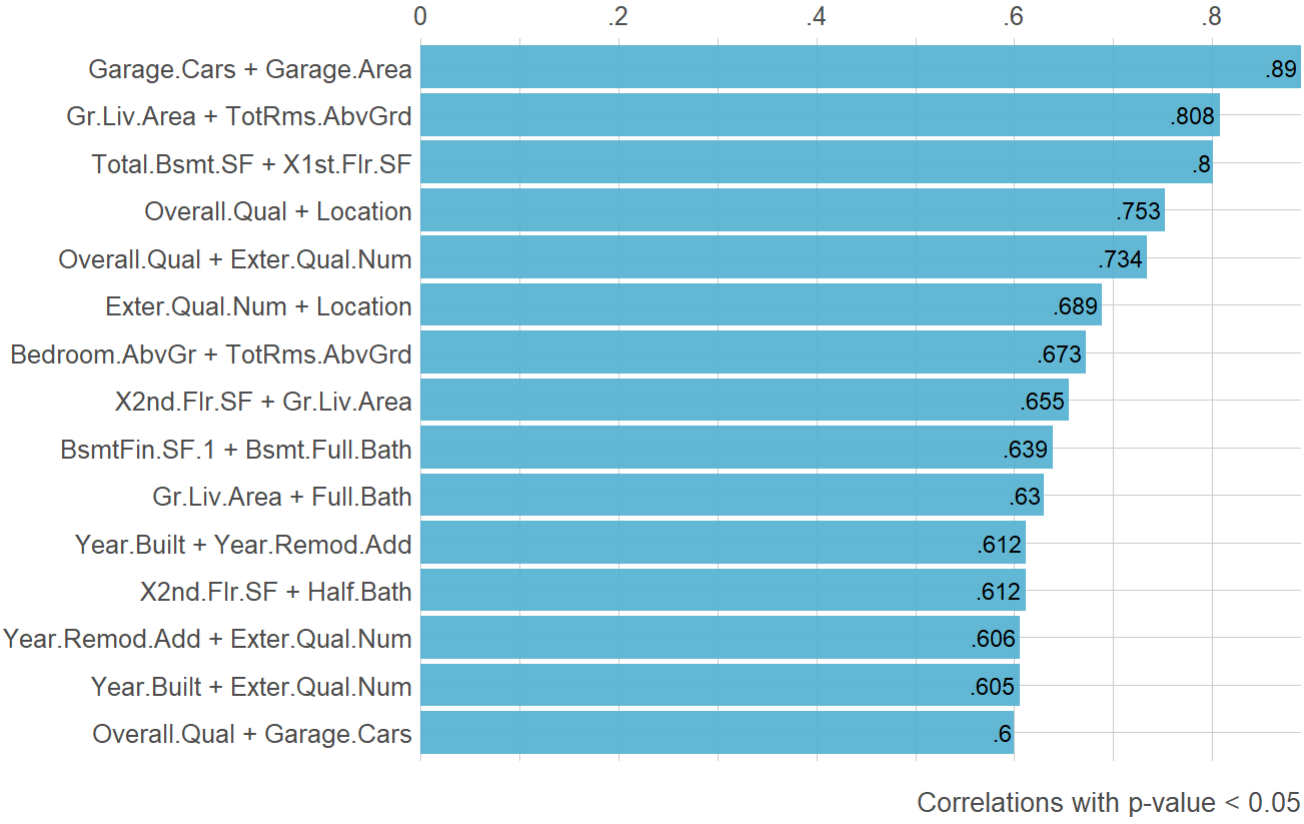
While some of these high correlation measures are to be expected, such as house year built along garage year built, we can also see some non-trivial patterns start to emerge from the more continuous numeric predictors.

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

```
corr_cross(numeric_housing_data,
  max_pvalue = 0.05,
  top = 15
)
```

Ranked Cross-Correlations

15 most relevant



```
#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)
}
```

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 varia
bles present
# 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 + Utilities + Lot.Config + Land.Slope + Condition.1  + House.Style + Year.Built + Founda
tion + Location + Roof.Style + Exterior.1st + Total.Bsmt.SF + Bsmt.Cond + Bsmt.Full.Bath + Heati
ng + Central.Air + Electrical + X1st.Flr.SF + X2nd.Flr.SF + Gr.Liv.Area + Full.Bath + Half.Bath
+ Bedroom.AbvGr + Kitchen.AbvGr + Kitchen.Qual + TotRms.AbvGrd + Fireplaces + Garage.Area + Pave
d.Drive + Wood.Deck.SF  + Open.Porch.SF + Pool.Area + Fence + Misc.Val + Mo.Sold + Yr.Sold, data
= ames_trn_data)

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

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

```
# check the number of predictors in the model
length(coef(saleprice_full_model_selected)) - 1
```

```
## [1] 109
```

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

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

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)
saleprice_forward_bic=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 varia
bles present
# 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 + Utilities + Lot.Config + Land.Slope + Condition.1  + House.Style + Year.Built + Founda
tion + Location + Roof.Style + Exterior.1st + Total.Bsmt.SF + Bsmt.Cond + Bsmt.Full.Bath + Heati
ng + Central.Air + Electrical + X1st.Flr.SF + X2nd.Flr.SF + Gr.Liv.Area + Full.Bath + Half.Bath
+ Bedroom.AbvGr + Kitchen.AbvGr + Kitchen.Qual + TotRms.AbvGrd + Fireplaces + Garage.Area + Pave
d.Drive + Wood.Deck.SF  + Open.Porch.SF + Pool.Area + Fence + Misc.Val + Mo.Sold + Yr.Sold, data
= ames_trn_data)
saleprice_selected_backward_aic = step(saleprice_full_model_selected, direction = "backward", tr
ace = 0)
```

Testing the built model

```
anova(saleprice_full_model_selected, saleprice_selected_backward_aic)
```

```
## Analysis of Variance Table
##
## Model 1: SalePrice ~ MS.Zoning + Lot.Area + Street + Lot.Shape + Land.Contour +
##     Utilities + Lot.Config + Land.Slope + Condition.1 + House.Style +
##     Year.Built + Foundation + Location + Roof.Style + Exterior.1st +
##     Total.Bsmt.SF + Bsmt.Cond + Bsmt.Full.Bath + Heating + Central.Air +
##     Electrical + X1st.Flr.SF + X2nd.Flr.SF + Gr.Liv.Area + Full.Bath +
##     Half.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 +
##     Mo.Sold + Yr.Sold
## Model 2: SalePrice ~ MS.Zoning + Lot.Area + Street + Lot.Shape + Land.Contour +
##     Lot.Config + Land.Slope + Condition.1 + House.Style + Year.Built +
##     Foundation + Location + Roof.Style + Exterior.1st + Total.Bsmt.SF +
##     Bsmt.Full.Bath + Central.Air + X2nd.Flr.SF + Gr.Liv.Area +
##     Full.Bath + Half.Bath + Bedroom.AbvGr + Kitchen.AbvGr + Kitchen.Qual +
##     TotRms.AbvGrd + Fireplaces + Garage.Area + Wood.Deck.SF +
##     Open.Porch.SF + Pool.Area + Misc.Val
##   Res.Df        RSS  Df   Sum of Sq      F Pr(>F)
## 1   2234 2.0742e+12
## 2   2261 2.0911e+12 -27 -1.6838e+10 0.6717 0.8985
```

```
anova(saleprice_selected_backward_aic, saleprice_forward_bic)
```

```
## Analysis of Variance Table
##
## Model 1: SalePrice ~ MS.Zoning + Lot.Area + Street + Lot.Shape + Land.Contour +
##     Lot.Config + Land.Slope + Condition.1 + House.Style + Year.Built +
##     Foundation + Location + Roof.Style + Exterior.1st + Total.Bsmt.SF +
##     Bsmt.Full.Bath + Central.Air + X2nd.Flr.SF + Gr.Liv.Area +
##     Full.Bath + Half.Bath + Bedroom.AbvGr + Kitchen.AbvGr + Kitchen.Qual +
##     TotRms.AbvGrd + Fireplaces + Garage.Area + Wood.Deck.SF +
##     Open.Porch.SF + Pool.Area + Misc.Val
## Model 2: SalePrice ~ Overall.Qual + Gr.Liv.Area + Neighborhood + Bsmt.Qual +
##     BsmtFin.SF.1 + Roof.Matl + MS.SubClass + Location + Bsmt.Exposure +
##     Kitchen.Qual + Garage.Area + Misc.Val + Year.Built + Lot.Area +
##     Screen.Porch + Sale.Condition + Fireplaces + Exter.Cond.Num +
##     Exter.Qual + Bsmt.Full.Bath + Total.Bsmt.SF + Bldg.Type +
##     Full.Bath + Mas.Vnr.Area + X2nd.Flr.SF
##   Res.Df        RSS Df  Sum of Sq F Pr(>F)
## 1   2261 2.0911e+12
## 2   2266 1.3489e+12 -5 7.4212e+11
```

Based on the anova F-tests, the forward BIC model is preferred.

In order to find an even better model, some trial and error testing was performed. Some variables were added and removed manually, and we tested each model by checking and comparing the adjusted r squared values and other metrics. One of the models that resulted from this process is below:

```
# Even more reduced model by several regression attempts:
saleprice_full_model_selected_reduced = lm(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 + TotR
ms.AbvGrd + Fireplaces + Garage.Area + Paved.Drive + Wood.Deck.SF  + Open.Porch.SF + Pool.Area +
Fence + Misc.Val + Yr.Sold  , data = ames_trn_data)

saleprice_selected_reduced_backward_aic = step(saleprice_full_model_selected_reduced, direction
= "backward", trace = 0)

formula(saleprice_selected_reduced_backward_aic)
```

```
## SalePrice ~ MS.Zoning + Lot.Area + Street + Lot.Shape + 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 +
##     Open.Porch.SF + Pool.Area + Misc.Val
```

```
# how much model is decreased from aic backward method
length(coef(saleprice_full_model_selected_reduced)) - 1
```

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

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

```
## [1] 42
```

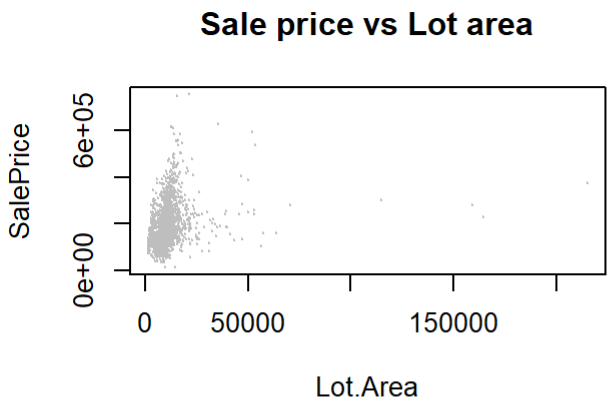
```
anova(saleprice_full_model_selected_reduced, saleprice_selected_reduced_backward_aic)
```

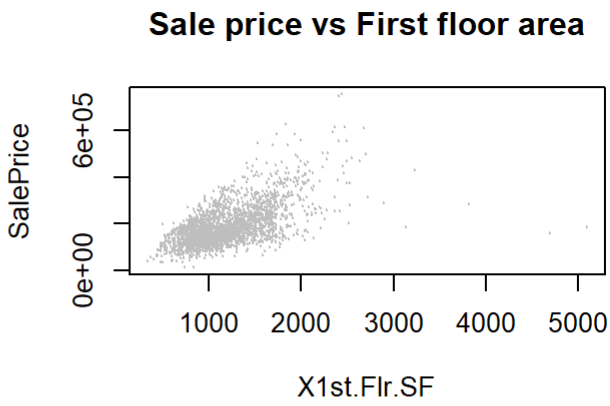
```
## Analysis of Variance Table
##
## Model 1: SalePrice ~ MS.Zoning + Lot.Area + Street + Lot.Shape + Utilities +
##     Land.Slope + House.Style + Year.Built + Foundation + Location +
##     Heating + Central.Air + Electrical + Gr.Liv.Area + Full.Bath +
##     Bedroom.AbvGr + Kitchen.AbvGr + Kitchen.Qual + 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 + 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 +
##     Open.Porch.SF + Pool.Area + Misc.Val
##   Res.Df        RSS  Df   Sum of Sq      F Pr(>F)
## 1   2281 2.3376e+12
## 2   2301 2.3516e+12 -20 -1.3981e+10 0.6821 0.8475
```

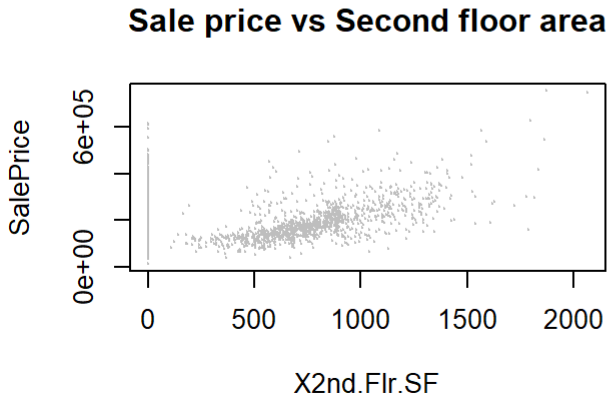
Variable transformations

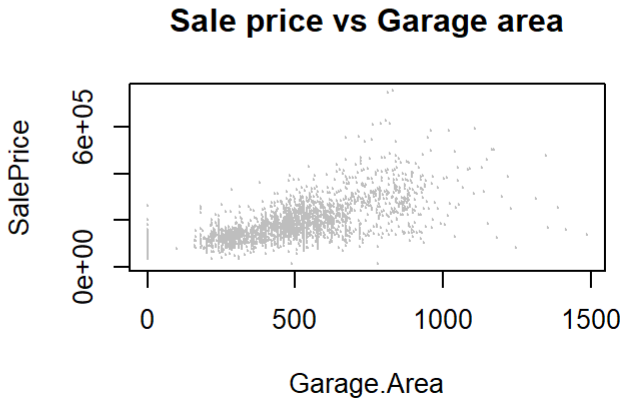
For this section, we'll begin by visualizing the relationships between the house sale prices and some of our predictors. By doing this, we're looking to gain some insight into potential variable transformations we could implement in order to improve the performance of our models.

```
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")
```

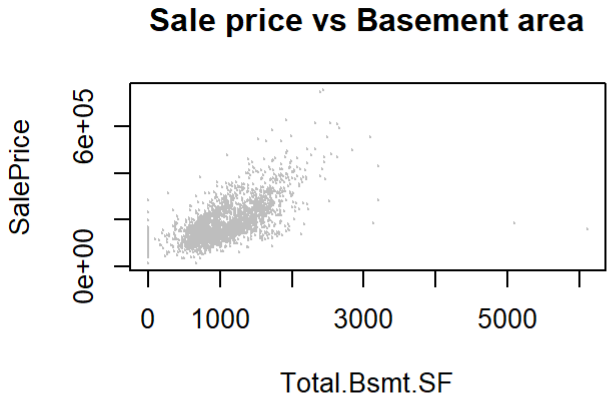


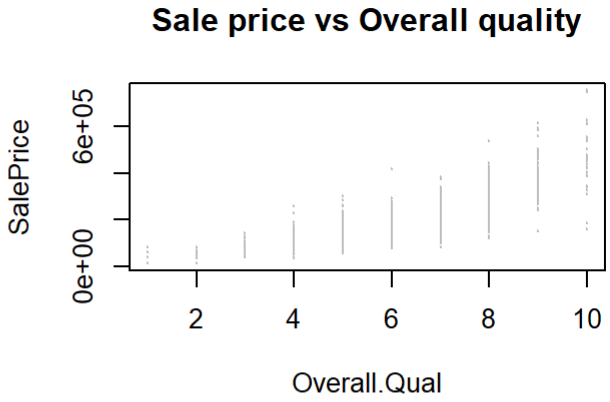


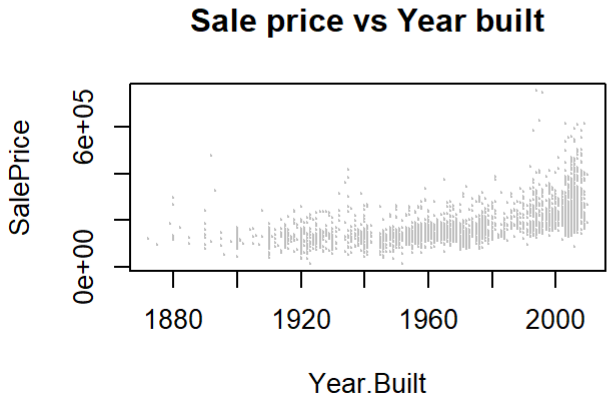


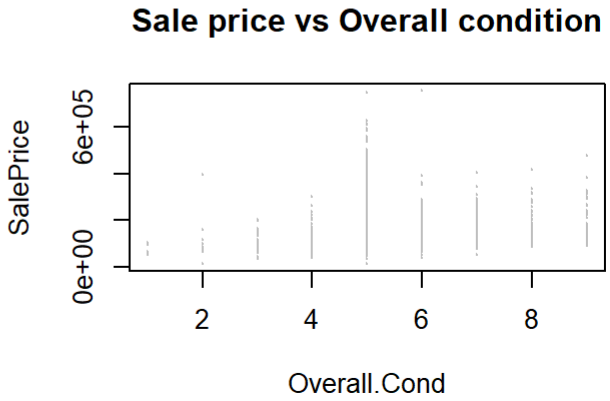



```
par(mfrow = c(2, 2))

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")
```

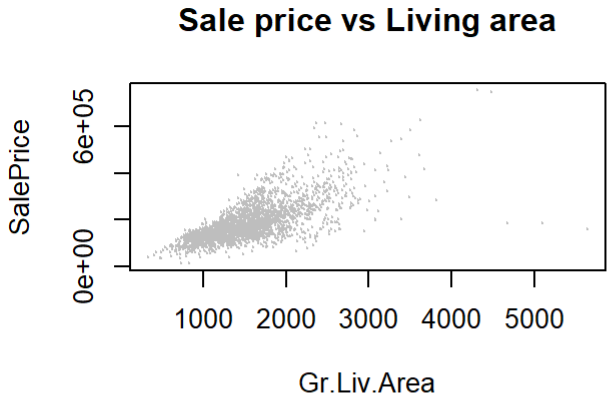


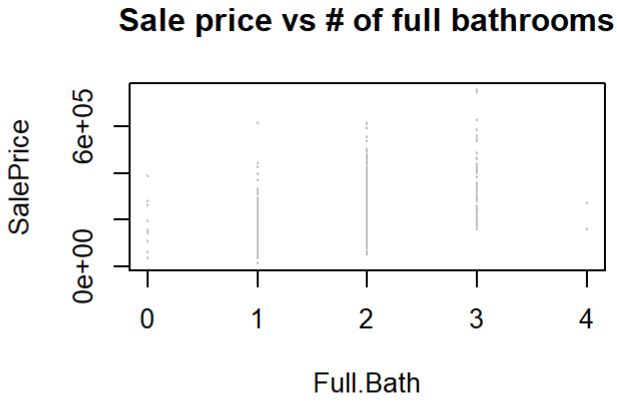


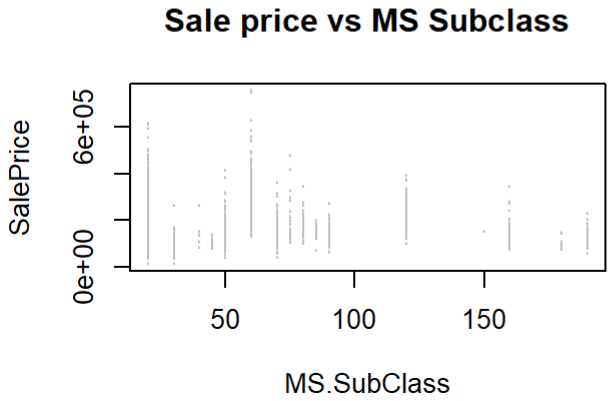


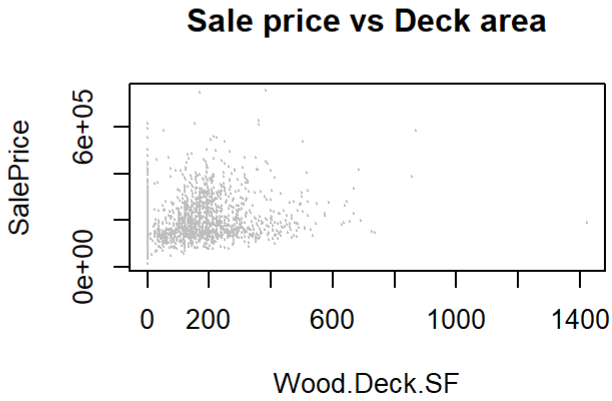



```
par(mfrow = c(2, 2))

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")
```

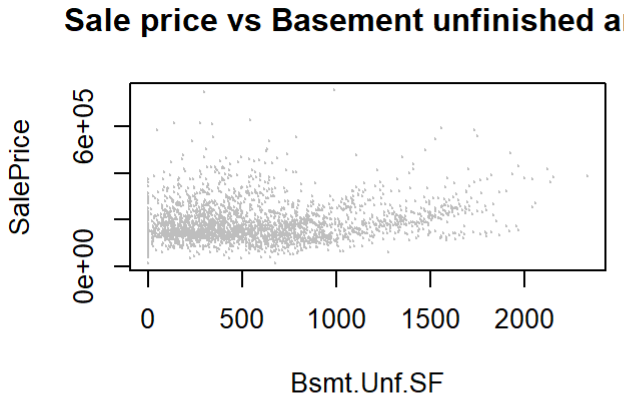


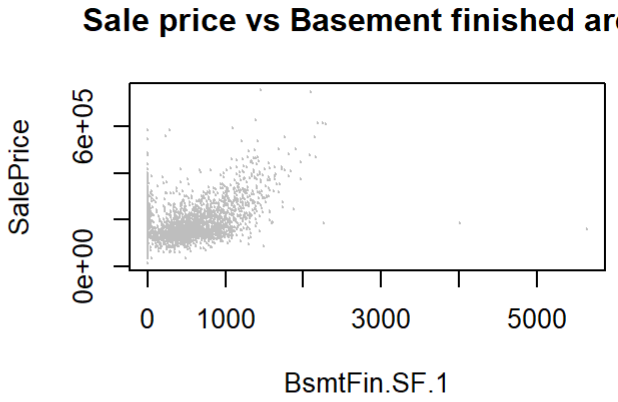


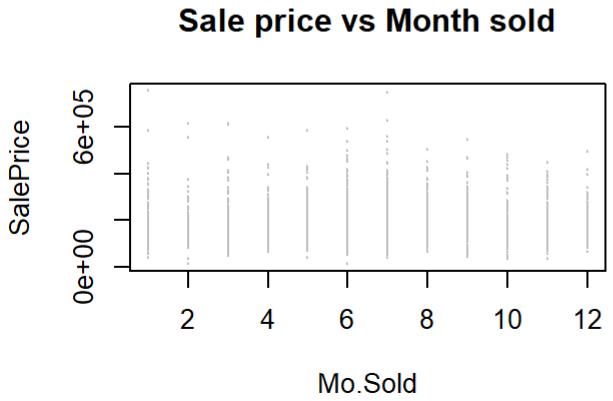


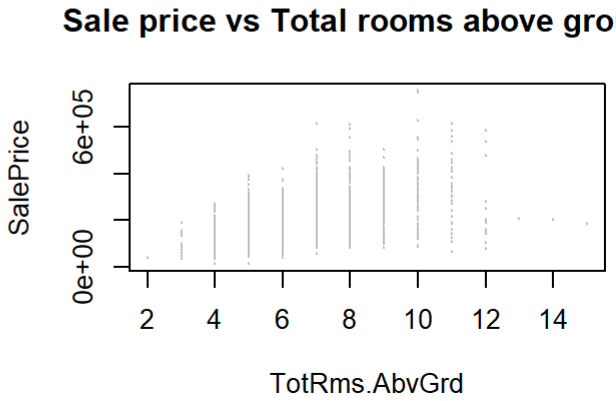



```
par(mfrow = c(2, 2))

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")
```

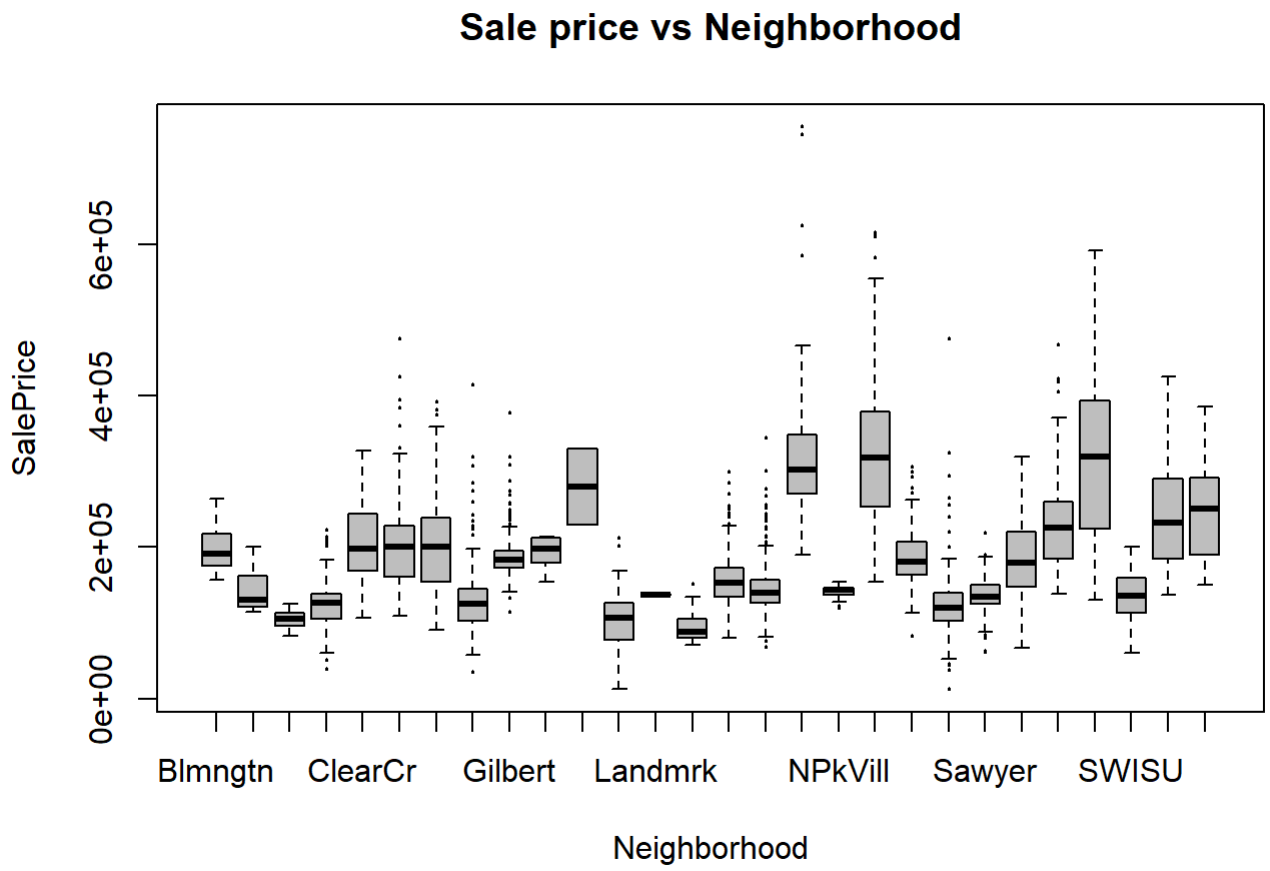









```
plot(SalePrice ~ Neighborhood, data = housing_data, col = "grey", pch = 20, cex = 0.3,
     main = "Sale price vs Neighborhood")
```



Based on the above visualizations of variable relationships and intuition on the dataset, we can start applying logarithmic and polynomial kernel transformations in order to improve model performance. After some testing, we present the following models that exhibit significant improvements in terms of error metrics:

```
transformed_model_old = lm(SalePrice ~ MS.Zoning + log(Lot.Area) + Street + Lot.Shape + Land.Slo
pe + House.Style + I(Year.Built^2) + Foundation + Location + Central.Air + Gr.Liv.Area + Bedroo
m.AbvGr + Kitchen.AbvGr + Kitchen.Qual + TotRms.AbvGrd + (Fireplaces) + I(Garage.Area^2) + (Woo
d.Deck.SF) + I(Pool.Area^2) + sqrt(Misc.Val) + Exter.Qual + log(TotRms.AbvGrd) + I(BsmtFin.SF.1^
3) + Overall.Qual, data = ames_trn_data)


transformed_model = lm(SalePrice ~ MS.Zoning + log(Lot.Area) + Street + I(Year.Built^2) + Kitche
n.AbvGr + Bedroom.AbvGr + Gr.Liv.Area + log(TotRms.AbvGrd) + (Fireplaces) + I(Garage.Area^2) + W
ood.Deck.SF + I(Pool.Area^2) + I(BsmtFin.SF.1^3) + sqrt(Misc.Val) + I(Overall.Qual^5) + I(Gr.Li
v.Area^2) + House.Style + Lot.Shape + Land.Slope + Foundation + Exter.Qual + Exter.Cond, data =
ames_trn_data)

# checking errors after fitting the train dataset.
sqrt(mean((ames_trn_data$SalePrice - fitted(saleprice_selected_reduced_backward_aic)) ^ 2))
```

```
## [1] 31673.89
```

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

```
## [1] 28755.43
```

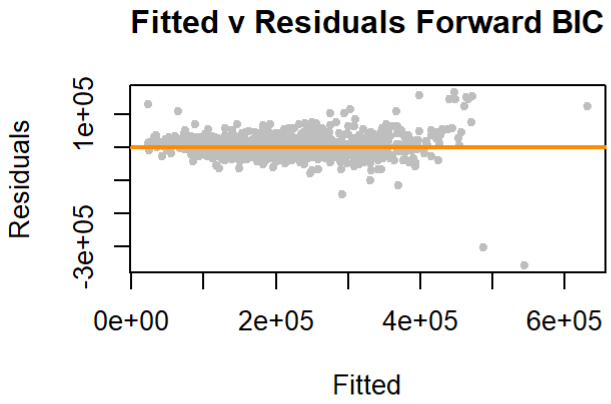
```
sqrt(mean((ames_trn_data$SalePrice - fitted(transformed_model)) ^ 2))
```

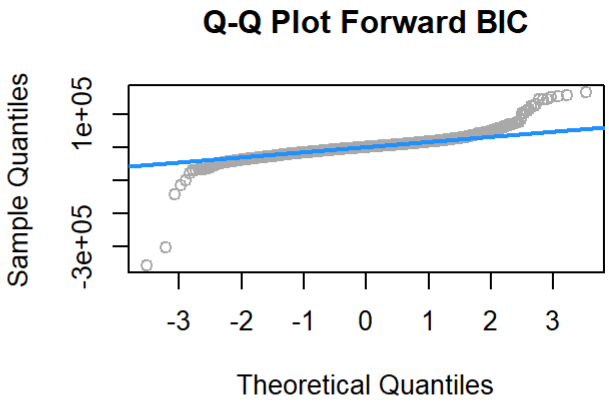
```
## [1] 28706.19
```

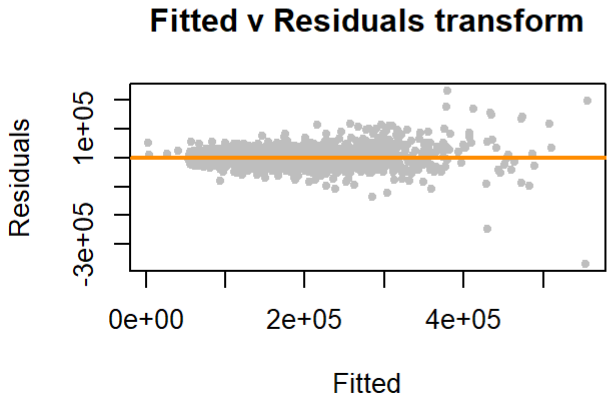
Results:

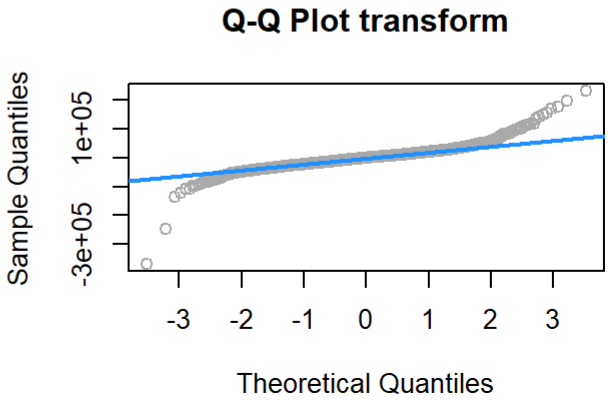
Now let's analyze how our our models follow the normality and homoscedasticity assumptions of linear regression, using the Fitted Vs. Residuals and Q-Q plots.

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

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

qqnorm(resid(saleprice_forward_bic), main = "Q-Q Plot Forward BIC", col = "darkgrey")
qqline(resid(saleprice_forward_bic), 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)
```

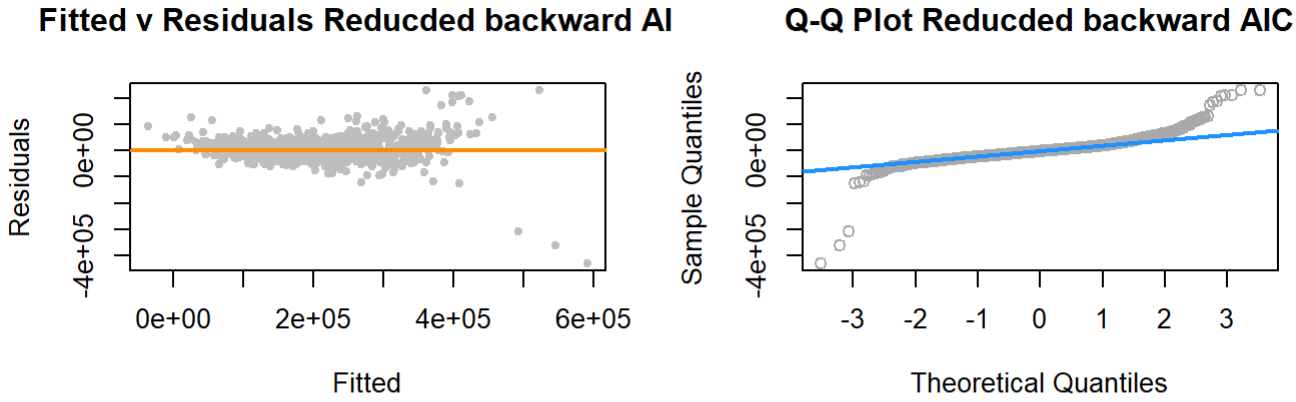









```
plot(fitted(saleprice_selected_reduced_backward_aic), resid(saleprice_selected_reduced_backward_
aic), col = "grey", pch = 20,
     xlab = "Fitted", ylab = "Residuals", main = "Fitted v Residuals Reducded backward AIC")
abline(h = 0, col = "darkorange", lwd = 2)

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



Based on a visually assuring performance of the forward BIC, we will take it a step further and apply some similar variable transformations that improved performance on previous model iterations.

```
formula(saleprice_forward_bic)
```

```
## SalePrice ~ Overall.Qual + Gr.Liv.Area + Neighborhood + Bsmt.Qual +
##     BsmtFin.SF.1 + Roof.Matl + MS.SubClass + Location + Bsmt.Exposure +
##     Kitchen.Qual + Garage.Area + Misc.Val + Year.Built + Lot.Area +
##     Screen.Porch + Sale.Condition + Fireplaces + Exter.Cond.Num +
##     Exter.Qual + 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.Matl + 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)
log_transformed_bic_mod=lm(log(SalePrice) ~ Overall.Qual + Gr.Liv.Area + Neighborhood + Bsmt.Qua
l +
    BsmtFin.SF.1 + Roof.Matl + 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

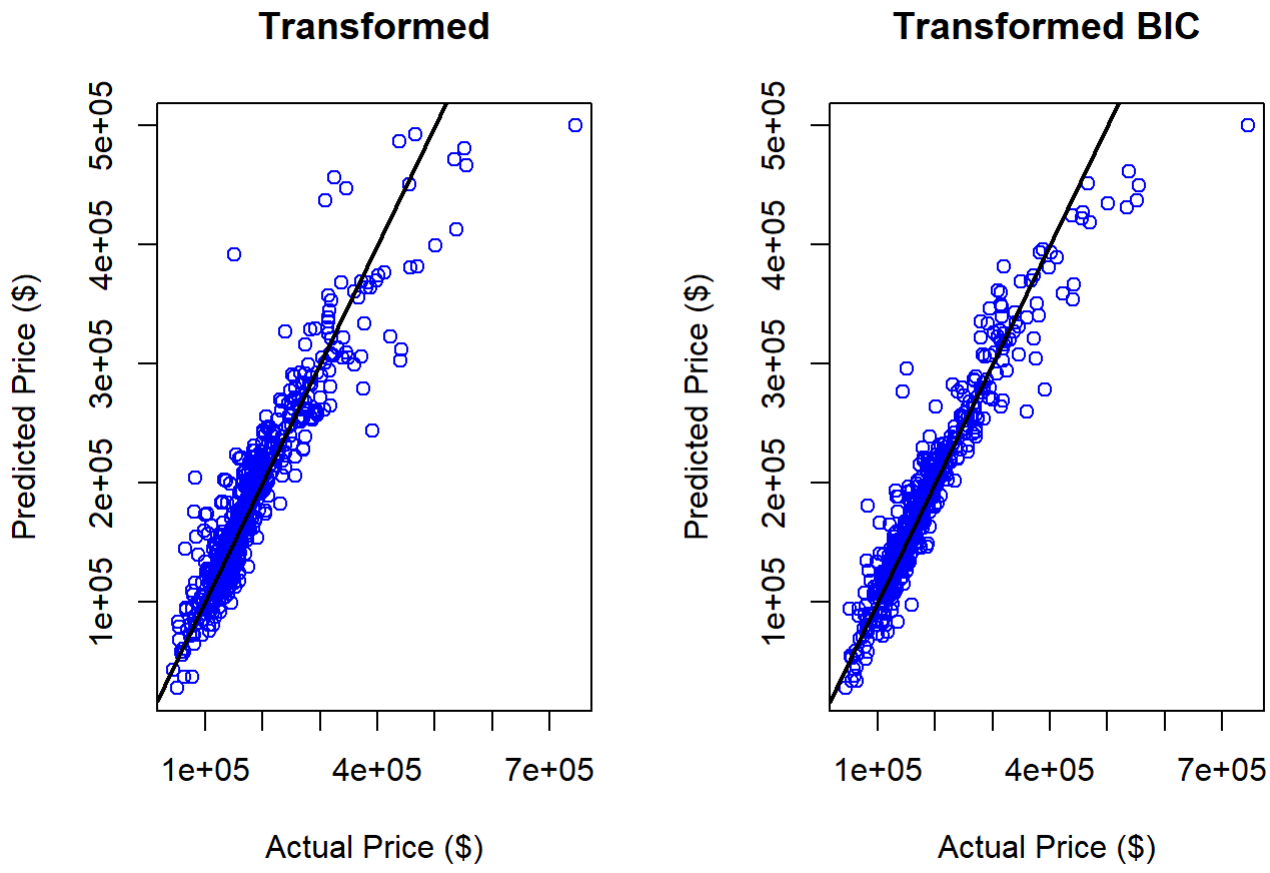
```
library(knitr)
models=list(transformed_model,saleprice_full_model_selected,saleprice_full_model_selected_reduce
d,saleprice_selected_backward_aic,saleprice_selected_reduced_backward_aic,saleprice_forward_bic,
transformed_bic_mod)

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)

#Compute the values for the Log Response BIC model
RMSE[8]=round(sqrt(mean((ames_trn_data$SalePrice - exp(fitted(log_transformed_bic_mod))) ^ 2)),
2)
ADJ_R2[8]=round(get_adj_r2(log_transformed_bic_mod),2)

#Average Prediction error for Log BIC Mode
 predicted = exp(predict(log_transformed_bic_mod, newdata = ames_tst_data))
  n = nrow(ames_tst_data)
  error = abs(ames_tst_data$SalePrice - predicted)
  avg_per_error = ((1 / n) * (sum(error / predicted))) * 100
  AVG_ERROR[8]= round(avg_per_error,2)

Models=c("transformed_model","saleprice_full_model_selected","saleprice_full_model_selected_redu
ced","saleprice_selected_backward_aic","saleprice_selected_reduced_backward_aic","bic_for_mo
d","transformed_bic_mod","log_transformed_bic_mod")
evaluation=cbind(Models,RMSE,AVG_ERROR,ADJ_R2)
kable(evaluation)
```

Models	RMSE	AVG_ERROR	ADJ_R2
transformed_model	28706.19	11.72	0.86

Models	RMSE	AVG_ERROR	ADJ_R2
saleprice_full_model_selected	29747.32	76.92	0.85
saleprice_full_model_selected_reduced	31579.59	8.71	0.83
saleprice_selected_backward_aic	29867.82	18.44	0.85
saleprice_selected_reduced_backward_aic	31673.89	8.99	0.83
bic_for_mod	23989.2	9.54	0.9
transformed_bic_mod	23979.21	9.38	0.9
log_transformed_bic_mod	27533.58	8.15	0.91

Now, seeing the result of Predicted VS. Actual plot for our two best performing models:

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



Discussion:

We have used several methods to determine the best model for predicting housing data using the Ames Iowa Housing Data from Kaggle.com.

Model Fit Analysis

The model created using Forward BIC, a log transform of SalePrice, and some transformed variables performed the best preddictions. It has an RMSE of 2.753358^{4}, an Adj R^2 of 0.91 and an average error of 8.15 percent when used on the test data. With our error under 10%, this model could be a very useful tool in predicting house prices in Ames, Iowa. Looking at our Predicted Vs. Actual graph, we do see the accuracy fall off as the houses get more expensive. If used in real life, we would need to be more cautious for high-end homes. That is not a problem for our team-member looking to buy.

Broader Applicability/Limitations

If we wanted to expand this model to areas outside of Ames, Iowa, we would need additional training data from other areas. Location is a crucial component of housing prices, as is partially confirmed with the inclusion of the 'Neighborhood' predictor in our BIC Forward Model. A further limitation of the model is that the Year Sold data only goes up until 2010. While the useful predictors may be the same, we would need to include newer data to ensure our predictions are still accurate.

Predictor Insights

While prediction accuracy was favored over Interpretability of predictor influence, we can still glean some insight from the included predictors in our models. In particular, our transformed_model has fairly low vif values, and may be the best model for gleaning insight on the impact of individual predictors on SalePrice. The Year Built and Gr.Liv.Area both have a polynomial influence on Sales Price which is not surprising while Roof Style and Land Slope are a bit more surprising.

Appendix:

Team Members: Ian Vetter - ivette2 Thomas Richter - thomasr8 Vivek Patel - vivekdp2

Some of the models that were created during the methods section. It include full additive model and then backward AIC and BIC of that additive model. We tried to make the interaction model but since, there are so many variables it took huge computation and have very large model over 100 predictors. so those were ignored.

```
saleprice_additive_model = lm(SalePrice ~ ., data = ames_trn_data)
n = length(resid(saleprice_additive_model))
saleprice_additive_model_bic = step(saleprice_additive_model, direction = "backward", k = log
(n), trace = 0)
```

Some model while doing the variable selection that were created but not later used are as follow:

```
SalePrice_regression_1 = lm(SalePrice ~ Lot.Area + Street + Lot.Shape + Utilities + Lot.Config +
Land.Slope + Condition.1  + House.Style + Year.Built + Foundation + Location + Roof.Style + Exte
rior.1st + Heating + Central.Air + Electrical + Gr.Liv.Area + Full.Bath + Half.Bath + Bedroom.Ab
vGr + Kitchen.AbvGr + Kitchen.Qual + TotRms.AbvGrd + Fireplaces + Garage.Area + Paved.Drive + Wo
od.Deck.SF  + Open.Porch.SF + Pool.Area + Fence + Misc.Val + Mo.Sold + Yr.Sold, data = ames_trn_
data)

SalePrice_regression_2 = lm(SalePrice ~ MS.Zoning + Lot.Area + Street + Lot.Config + Year.Built
+ Foundation + Location + Heating + Central.Air + Electrical + Gr.Liv.Area + Full.Bath + Bedroo
m.AbvGr + Kitchen.AbvGr + Kitchen.Qual + Fireplaces + Garage.Area + Paved.Drive + Misc.Val + Yr.
Sold, data = ames_trn_data)
```