HARVARD EXTENSION SCHOOL

EXT CSCI E-106 Model Data Class Special Project Template

Nilay Sundarkar Christopher Craddock Seraphim Eilken Simon Carandang

05 May 2024

Contents

Data

Refer to the **Housing prices in Ames, Iowa**

```
2930 observations, 82 variables
```

```
#Step 0: Data Preparations
#install.packages("visdat")
library(readr)
library(visdat)
library(tidyr)
library(MASS)
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
library(olsrr)
```

```
##
## Attaching package: 'olsrr'
```

```
## The following object is masked from 'package:MASS':
##
##     cement
```

```
## The following object is masked from 'package:datasets':
##
##     rivers
```

```
library(ggplot2)
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
```

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

```
library(olsrr)
library(car)
```

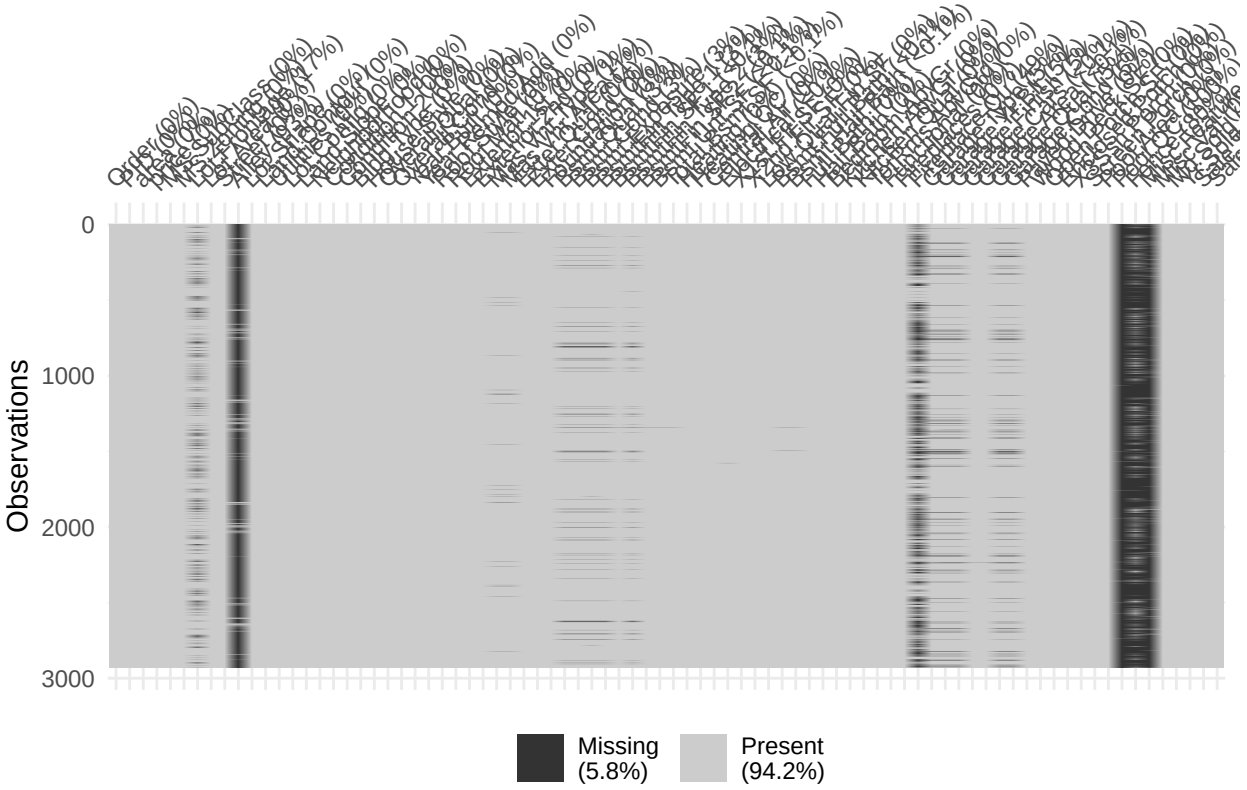
```
## Loading required package: carData
```

```
library(rpart)
library(rpart.plot)
# Checking for NA, or missing data using graphics
ames_data <- read_csv("ames.csv")
```

```
## Rows: 2930 Columns: 82
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (43): MS.Zoning, Street, Alley, Lot.Shape, Land.Contour, Utilities, Lot....
## dbl (39): Order, PID, area, price, MS.SubClass, Lot.Frontage, Lot.Area, Over...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
vis_miss(ames_data)
```




```
str(ames_data)
```

```
## spc_tbl_ [2,930 x 82] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Order          : num [1:2930] 1 2 3 4 5 6 7 8 9 10 ...
##  $ PID            : num [1:2930] 5.26e+08 5.26e+08 5.26e+08 5.26e+08 5.27e+08 ...
##  $ area           : num [1:2930] 1656 896 1329 2110 1629 ...
##  $ price          : num [1:2930] 215000 105000 172000 244000 189900 ...
##  $ MS.SubClass    : num [1:2930] 20 20 20 20 60 60 120 120 120 60 ...
##  $ MS.Zoning      : chr [1:2930] "RL" "RH" "RL" "RL" ...
```

```
##  $ Lot.Frontage   : num [1:2930] 141 80 81 93 74 78 41 43 39 60 ...
##  $ Lot.Area       : num [1:2930] 31770 11622 14267 11160 13830 ...
##  $ Street         : chr [1:2930] "Pave" "Pave" "Pave" "Pave" ...
##  $ Alley          : chr [1:2930] NA NA NA NA ...
##  $ Lot.Shape      : chr [1:2930] "IR1" "Reg" "IR1" "Reg" ...
##  $ Land.Contour   : chr [1:2930] "Lvl" "Lvl" "Lvl" "Lvl" ...
##  $ Utilities      : chr [1:2930] "AllPub" "AllPub" "AllPub" "AllPub" ...
##  $ Lot.Config     : chr [1:2930] "Corner" "Inside" "Corner" "Corner" ...
##  $ Land.Slope     : chr [1:2930] "Gtl" "Gtl" "Gtl" "Gtl" ...
##  $ Neighborhood   : chr [1:2930] "NAmes" "NAmes" "NAmes" "NAmes" ...
##  $ Condition.1    : chr [1:2930] "Norm" "Feedr" "Norm" "Norm" ...
##  $ Condition.2    : chr [1:2930] "Norm" "Norm" "Norm" "Norm" ...
##  $ Bldg.Type      : chr [1:2930] "1Fam" "1Fam" "1Fam" "1Fam" ...
##  $ House.Style    : chr [1:2930] "1Story" "1Story" "1Story" "1Story" ...
##  $ Overall.Qual   : num [1:2930] 6 5 6 7 5 6 8 8 8 7 ...
##  $ Overall.Cond   : num [1:2930] 5 6 6 5 5 6 5 5 5 5 ...
##  $ Year.Built     : num [1:2930] 1960 1961 1958 1968 1997 ...
##  $ Year.Remod.Add : num [1:2930] 1960 1961 1958 1968 1998 ...
##  $ Roof.Style     : chr [1:2930] "Hip" "Gable" "Hip" "Hip" ...
##  $ Roof.Matl      : chr [1:2930] "CompShg" "CompShg" "CompShg" "CompShg" ...
##  $ Exterior.1st   : chr [1:2930] "BrkFace" "VinylSd" "Wd Sdng" "BrkFace" ...
##  $ Exterior.2nd   : chr [1:2930] "Plywood" "VinylSd" "Wd Sdng" "BrkFace" ...
##  $ Mas.Vnr.Type   : chr [1:2930] "Stone" "None" "BrkFace" "None" ...
##  $ Mas.Vnr.Area   : num [1:2930] 112 0 108 0 0 20 0 0 0 0 ...
##  $ Exter.Qual     : chr [1:2930] "TA" "TA" "TA" "Gd" ...
##  $ Exter.Cond     : chr [1:2930] "TA" "TA" "TA" "TA" ...
##  $ Foundation     : chr [1:2930] "CBlock" "CBlock" "CBlock" "CBlock" ...
##  $ Bsmt.Qual      : chr [1:2930] "TA" "TA" "TA" "TA" ...
##  $ Bsmt.Cond      : chr [1:2930] "Gd" "TA" "TA" "TA" ...
##  $ Bsmt.Exposure  : chr [1:2930] "Gd" "No" "No" "No" ...
##  $ BsmtFin.Type.1 : chr [1:2930] "BLQ" "Rec" "ALQ" "ALQ" ...
##  $ BsmtFin.SF.1   : num [1:2930] 639 468 923 1065 791 ...
##  $ BsmtFin.Type.2 : chr [1:2930] "Unf" "LwQ" "Unf" "Unf" ...
##  $ BsmtFin.SF.2   : num [1:2930] 0 144 0 0 0 0 0 0 0 0 ...
##  $ Bsmt.Unf.SF    : num [1:2930] 441 270 406 1045 137 ...
##  $ Total.Bsmt.SF  : num [1:2930] 1080 882 1329 2110 928 ...
##  $ Heating        : chr [1:2930] "GasA" "GasA" "GasA" "GasA" ...
##  $ Heating.QC     : chr [1:2930] "Fa" "TA" "TA" "Ex" ...
##  $ Central.Air    : chr [1:2930] "Y" "Y" "Y" "Y" ...
##  $ Electrical     : chr [1:2930] "SBrkr" "SBrkr" "SBrkr" "SBrkr" ...
##  $ X1st.Flr.SF    : num [1:2930] 1656 896 1329 2110 928 ...
##  $ X2nd.Flr.SF    : num [1:2930] 0 0 0 0 701 678 0 0 0 776 ...
##  $ Low.Qual.Fin.SF: num [1:2930] 0 0 0 0 0 0 0 0 0 0 ...
##  $ Bsmt.Full.Bath : num [1:2930] 1 0 0 1 0 0 1 0 1 0 ...
##  $ Bsmt.Half.Bath : num [1:2930] 0 0 0 0 0 0 0 0 0 0 ...
##  $ Full.Bath      : num [1:2930] 1 1 1 2 2 2 2 2 2 2 ...
##  $ Half.Bath      : num [1:2930] 0 0 1 1 1 1 0 0 0 1 ...
##  $ Bedroom.AbvGr  : num [1:2930] 3 2 3 3 3 3 2 2 2 3 ...
##  $ Kitchen.AbvGr  : num [1:2930] 1 1 1 1 1 1 1 1 1 1 ...
##  $ Kitchen.Qual   : chr [1:2930] "TA" "TA" "Gd" "Ex" ...
##  $ TotRms.AbvGrd  : num [1:2930] 7 5 6 8 6 7 6 5 5 7 ...
##  $ Functional     : chr [1:2930] "Typ" "Typ" "Typ" "Typ" ...
##  $ Fireplaces     : num [1:2930] 2 0 0 2 1 1 0 0 1 1 ...
##  $ Fireplace.Qu   : chr [1:2930] "Gd" NA NA "TA" ...
##  $ Garage.Type    : chr [1:2930] "Attchd" "Attchd" "Attchd" "Attchd" ...
##  $ Garage.Yr.Blt  : num [1:2930] 1960 1961 1958 1968 1997 ...
##  $ Garage.Finish  : chr [1:2930] "Fin" "Unf" "Unf" "Fin" ...
##  $ Garage.Cars    : num [1:2930] 2 1 1 2 2 2 2 2 2 2 ...
##  $ Garage.Area    : num [1:2930] 528 730 312 522 482 470 582 506 608 442 ...
##  $ Garage.Qual    : chr [1:2930] "TA" "TA" "TA" "TA" ...
```

```
##  $ Garage.Cond    : chr [1:2930] "TA" "TA" "TA" "TA" ...
##  $ Paved.Drive    : chr [1:2930] "P" "Y" "Y" "Y" ...
##  $ Wood.Deck.SF   : num [1:2930] 210 140 393 0 212 360 0 0 237 140 ...
##  $ Open.Porch.SF  : num [1:2930] 62 0 36 0 34 36 0 82 152 60 ...
##  $ Enclosed.Porch : num [1:2930] 0 0 0 0 0 0 170 0 0 0 ...
##  $ X3Ssn.Porch    : num [1:2930] 0 0 0 0 0 0 0 0 0 0 ...
##  $ Screen.Porch   : num [1:2930] 0 120 0 0 0 0 0 144 0 0 ...
##  $ Pool.Area      : num [1:2930] 0 0 0 0 0 0 0 0 0 0 ...
##  $ Pool.QC        : chr [1:2930] NA NA NA NA ...
##  $ Fence          : chr [1:2930] NA "MnPrv" NA NA ...
##  $ Misc.Feature   : chr [1:2930] NA NA "Gar2" NA ...
##  $ Misc.Val       : num [1:2930] 0 0 12500 0 0 0 0 0 0 0 ...
##  $ Mo.Sold        : num [1:2930] 5 6 6 4 3 6 4 1 3 6 ...
##  $ Yr.Sold        : num [1:2930] 2010 2010 2010 2010 2010 2010 2010 2010 2010 2010 ...
##  $ Sale.Type      : chr [1:2930] "WD" "WD" "WD" "WD" ...
##  $ Sale.Condition : chr [1:2930] "Normal" "Normal" "Normal" "Normal" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Order = col_double(),
##   ..   PID = col_double(),
##   ..   area = col_double(),
##   ..   price = col_double(),
##   ..   MS.SubClass = col_double(),
##   ..   MS.Zoning = col_character(),
##   ..   Lot.Frontage = col_double(),
##   ..   Lot.Area = col_double(),
##   ..   Street = col_character(),
##   ..   Alley = col_character(),
##   ..   Lot.Shape = col_character(),
##   ..   Land.Contour = col_character(),
##   ..   Utilities = col_character(),
##   ..   Lot.Config = col_character(),
##   ..   Land.Slope = col_character(),
##   ..   Neighborhood = col_character(),
##   ..   Condition.1 = col_character(),
##   ..   Condition.2 = col_character(),
##   ..   Bldg.Type = col_character(),
##   ..   House.Style = col_character(),
##   ..   Overall.Qual = col_double(),
##   ..   Overall.Cond = col_double(),
##   ..   Year.Built = col_double(),
##   ..   Year.Remod.Add = col_double(),
##   ..   Roof.Style = col_character(),
##   ..   Roof.Matl = col_character(),
##   ..   Exterior.1st = col_character(),
##   ..   Exterior.2nd = col_character(),
##   ..   Mas.Vnr.Type = col_character(),
##   ..   Mas.Vnr.Area = col_double(),
##   ..   Exter.Qual = col_character(),
##   ..   Exter.Cond = col_character(),
##   ..   Foundation = col_character(),
##   ..   Bsmt.Qual = col_character(),
##   ..   Bsmt.Cond = col_character(),
##   ..   Bsmt.Exposure = col_character(),
##   ..   BsmtFin.Type.1 = col_character(),
##   ..   BsmtFin.SF.1 = col_double(),
##   ..   BsmtFin.Type.2 = col_character(),
##   ..   BsmtFin.SF.2 = col_double(),
##   ..   Bsmt.Unf.SF = col_double(),
##   ..   Total.Bsmt.SF = col_double(),
```

```
##   ..   Heating = col_character(),
##   ..   Heating.QC = col_character(),
##   ..   Central.Air = col_character(),
##   ..   Electrical = col_character(),
##   ..   X1st.Flr.SF = col_double(),
##   ..   X2nd.Flr.SF = col_double(),
##   ..   Low.Qual.Fin.SF = col_double(),
##   ..   Bsmt.Full.Bath = col_double(),
##   ..   Bsmt.Half.Bath = col_double(),
##   ..   Full.Bath = col_double(),
##   ..   Half.Bath = col_double(),
##   ..   Bedroom.AbvGr = col_double(),
##   ..   Kitchen.AbvGr = col_double(),
##   ..   Kitchen.Qual = col_character(),
##   ..   TotRms.AbvGrd = col_double(),
##   ..   Functional = col_character(),
##   ..   Fireplaces = col_double(),
##   ..   Fireplace.Qu = col_character(),
##   ..   Garage.Type = col_character(),
##   ..   Garage.Yr.Blt = col_double(),
##   ..   Garage.Finish = col_character(),
##   ..   Garage.Cars = col_double(),
##   ..   Garage.Area = col_double(),
##   ..   Garage.Qual = col_character(),
##   ..   Garage.Cond = col_character(),
##   ..   Paved.Drive = col_character(),
##   ..   Wood.Deck.SF = col_double(),
##   ..   Open.Porch.SF = col_double(),
##   ..   Enclosed.Porch = col_double(),
##   ..   X3Ssn.Porch = col_double(),
##   ..   Screen.Porch = col_double(),
##   ..   Pool.Area = col_double(),
##   ..   Pool.QC = col_character(),
##   ..   Fence = col_character(),
##   ..   Misc.Feature = col_character(),
##   ..   Misc.Val = col_double(),
##   ..   Mo.Sold = col_double(),
##   ..   Yr.Sold = col_double(),
##   ..   Sale.Type = col_character(),
##   ..   Sale.Condition = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

Description

Data set contains information from the Ames Assessor's Office used in computing assessed values for individual residential properties sold in Ames, IA from 2006 to 2010. See here for detailed variable descriptions.

Objective

Using the data build a prediction model using explanatory variables or predictors to allow a typical buyer or real estate agent to sit down and estimate the selling price of a house "SalePrice" (It is a continuous variable) is the response variable.

Due Date: May 6, 2024 at 11:59 pm EST

Instructions:

1	Join a team with your fellow students with appropriate size (at most four students total). You may post an advertising in ED. Once you are set, send to rafael_gomeztagle@g.harvard.edu the name of the team members and their emails.
2	Review the dataset named "ames'csv, report on preliminary findings (missing data, type of variables, distributions).
3	Create the train data set which contains 70% of the data and use set.seed (1023). The remaining 30% will be your test data set.
4	Investigate the data and combine the level of categorical variables if needed and drop variables as needed. For example, you may drop id, variables with too many missing observations, etc.
5	Create scatter plots and a correlation matrix for the train data set. Interpret the possible relationship between the response and the covariates.
6	Build several multiple linear models by using the stepwise selection methods. Compare the performance of the best two linear models.
7	Make sure that model assumption(s) are checked for the final model. Apply remedy measures (transformation, etc.) that helps satisfy the linear model assumptions.
8	Investigate unequal variances and multicollinearity. If necessary, apply remedial methods (WLS, Ridge, Elastic Net, Lasso, etc.).
9	Build an alternative model based on one of the following approaches to predict price: regression tree, NN, or SVM. Then check again the applicable model assumptions.
10	Use the test data set to assess the model performances from above.
11	Based on the performances on both train and test data sets, determine your primary (champion) model and the other model which would be your benchmark model.
12	Create a model development document that describes the model following this template, input the name of the authors, Harvard IDs, the name of the Group, all of your code and calculations, etc.: be sure you populate all the sections of this template.
13	Each student must submit the files on Canvas to get the full credit.

Notes: No typographical errors, grammar mistakes, or misspelled words, use English language All tables need to be numbered and describe their content in the body of the document All figures/graphs need to be numbered and describe their content All results must be accurate and clearly explained for a casual reviewer to fully understand their purpose and impact Submit both the RMD markdown file and PDF with the sections with appropriate explanations. A more formal document in Word can be used in place of the pdf file but must include all appropriate explanations.

1. Send email details - done by Simon
2. Review the dataset named "ames'csv, report on preliminary findings (missing data, type of variables, distributions).

   Data set contains information from the Ames Assessor's Office used in computing assessed values for individual residential properties sold in Ames, IA from 2006 to 2010.

   The data has 82 columns which include 23 nominal, 23 ordinal, 14 discrete, and 20 continuous variables (and 2 additional observation identifiers).
3. Create the train data set which contains 70% of the data and use set.seed (1023). The remaining 30% will be your test data set.

```
ames.smp_size <- floor(0.70 * nrow(ames_data))
set.seed(1023)
ames.train_index <- sample(seq_len(nrow(ames_data)), size = ames.smp_size)
ames.train_data <- ames_data[ames.train_index, ]
ames.test_data <- ames_data[-ames.train_index, ]
```

4. Investigate the data and combine the level of categorical variables if needed and drop variables as needed. For example, you may drop id, variables with too many missing observations, etc.

Dropping Order, PID as they are just an identifier for the observations.

Dropping "Pool.QC", "Misc.Feature", "Alley","Fence", "Fireplace.Qu", "Lot.Frontage","Garage.Yr.Blt", "Garage.Finish", "Garage.Qual","Garage.Cond", "Garage.Type" as they have high number of missing values.

Cleaning NA rows for the rest of the data.

```
ames_data.df <- data.frame(ames_data)
# Domain analysis to clean up data
# We rely on descriptions/comments provided at https://jse.amstat.org/v19n3/decock/DataDocumentation.txt, stat
# The document mentions below for outliers -
#SPECIAL NOTES:
#There are 5 observations that an instructor may wish to remove from the data set before giving it to students
ames.known_outliers <- ames_data.df[ames_data.df$area > 4000,]
ames_data.df <- ames_data.df[!(ames_data.df$PID %in% ames.known_outliers$PID),]
ames.train_data.df <- data.frame(ames.train_data)
ames.train_data.df <- ames.train_data.df[!(ames.train_data.df$PID %in% ames.known_outliers$PID),]
ames.test_data.df <- data.frame(ames.test_data)
ames.test_data.df <- ames.test_data.df[!(ames.test_data.df$PID %in% ames.known_outliers$PID),]

# check which columns have NA values and how many per column
na_count <-sapply(ames_data.df, function(y) sum(length(which(is.na(y)))))
na_count <- data.frame(na_count)
na_count$name<-rownames(na_count)
na_count <- na_count[na_count$na_count>0,]
na_count <- data.frame(na_count)
na_count <- na_count[order(na_count$na_count, decreasing = TRUE), ]
na_count
```

```
##                 na_count           name
## Pool.QC             2914        Pool.QC
## Misc.Feature        2820   Misc.Feature
## Alley               2727          Alley
## Fence               2354          Fence
## Fireplace.Qu        1422   Fireplace.Qu
## Lot.Frontage         490   Lot.Frontage
## Garage.Yr.Blt        159  Garage.Yr.Blt
## Garage.Finish        159  Garage.Finish
## Garage.Qual          159    Garage.Qual
## Garage.Cond          159    Garage.Cond
## Garage.Type          157    Garage.Type
## Bsmt.Exposure         83  Bsmt.Exposure
## BsmtFin.Type.2        81 BsmtFin.Type.2
## Bsmt.Qual             80      Bsmt.Qual
## Bsmt.Cond             80      Bsmt.Cond
## BsmtFin.Type.1        80 BsmtFin.Type.1
## Mas.Vnr.Type          23   Mas.Vnr.Type
## Mas.Vnr.Area          23   Mas.Vnr.Area
## Bsmt.Full.Bath         2 Bsmt.Full.Bath
## Bsmt.Half.Bath         2 Bsmt.Half.Bath
## BsmtFin.SF.1           1   BsmtFin.SF.1
## BsmtFin.SF.2           1   BsmtFin.SF.2
## Bsmt.Unf.SF            1    Bsmt.Unf.SF
```

```
## Total.Bsmt.SF         1  Total.Bsmt.SF
## Electrical            1     Electrical
## Garage.Cars           1    Garage.Cars
## Garage.Area           1    Garage.Area
```

```
# drop columns that have very high number of NA values and those that are not related to the response variable
# Misc.Feature is directly associated with Misc.Val - so dropping Misc.Val
drops <- c("Pool.QC", "Misc.Feature", "Alley","Fence", "Fireplace.Qu", "Order", "PID", "Misc.Val")
ames_data.df <- ames_data.df[ , !(names(ames_data.df) %in% drops)]
```

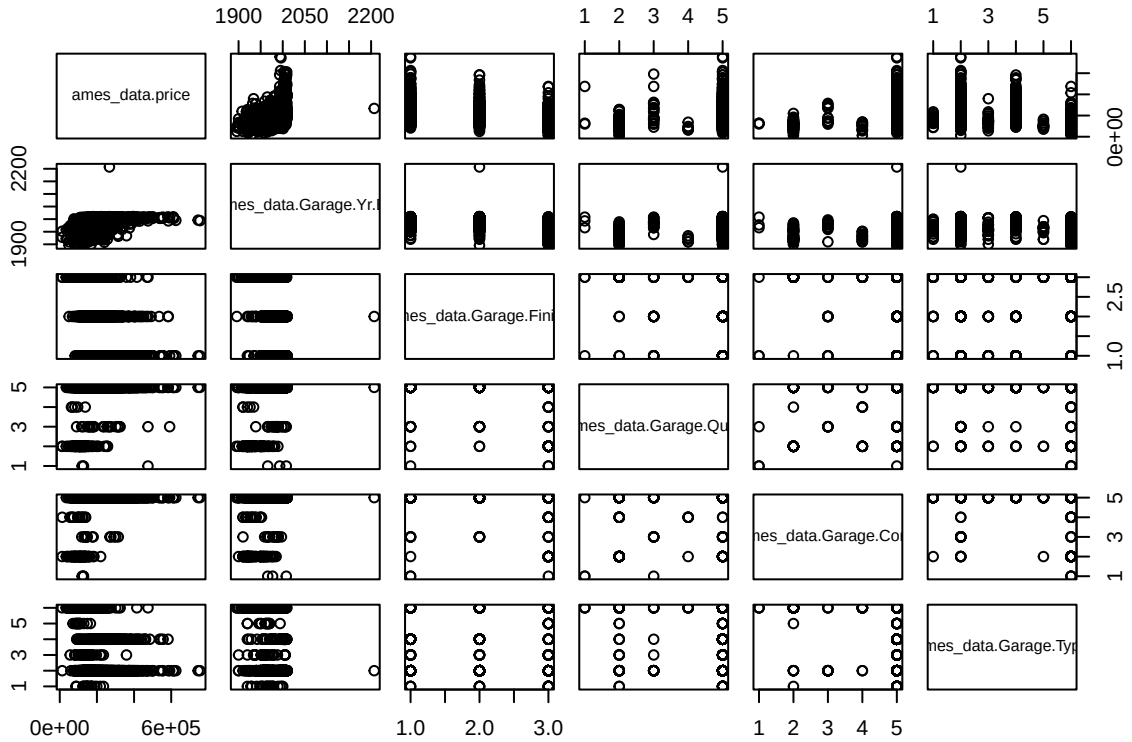
```
#for the rest of columns that have somewhat high NA values, we remove the NA rows and check the correlation be
`%notin%` <- Negate(`%in%`)
na_count <- na_count[na_count$name %notin% drops,]
na_count
```

```
##                 na_count           name
## Lot.Frontage         490   Lot.Frontage
## Garage.Yr.Blt        159  Garage.Yr.Blt
## Garage.Finish        159  Garage.Finish
## Garage.Qual          159    Garage.Qual
## Garage.Cond          159    Garage.Cond
## Garage.Type          157    Garage.Type
## Bsmt.Exposure         83  Bsmt.Exposure
## BsmtFin.Type.2        81 BsmtFin.Type.2
## Bsmt.Qual             80      Bsmt.Qual
## Bsmt.Cond             80      Bsmt.Cond
## BsmtFin.Type.1        80 BsmtFin.Type.1
## Mas.Vnr.Type          23   Mas.Vnr.Type
## Mas.Vnr.Area          23   Mas.Vnr.Area
## Bsmt.Full.Bath         2 Bsmt.Full.Bath
## Bsmt.Half.Bath         2 Bsmt.Half.Bath
## BsmtFin.SF.1           1   BsmtFin.SF.1
## BsmtFin.SF.2           1   BsmtFin.SF.2
## Bsmt.Unf.SF            1    Bsmt.Unf.SF
## Total.Bsmt.SF          1  Total.Bsmt.SF
## Electrical             1     Electrical
## Garage.Cars            1    Garage.Cars
## Garage.Area            1    Garage.Area
```

```
# Lot.Frontage has 490 NA values
Lot.Frontage_df <- data.frame(ames_data$price,ames_data$Lot.Frontage)
Lot.Frontage_df <- drop_na(Lot.Frontage_df)
# low correlation between Lot.Frontage and SalePrice
cor(Lot.Frontage_df$ames_data.price,Lot.Frontage_df$ames_data.Lot.Frontage)
```

```
## [1] 0.3573179
```

```
# Garage.Yr.Blt, Garage.Finish , Garage.Qual, Garage.Cond and Garage.Type are all related to Garage and seem t
Garage_df <- data.frame(ames_data$price, ames_data$Garage.Yr.Blt, ames_data$Garage.Finish, ames_data$Garage.Qu
Garage_df <- drop_na(Garage_df)
# no significant correlation is observed with the response variable
plot(Garage_df)
```




```
drops <- c(drops, "Lot.Frontage","Garage.Yr.Blt", "Garage.Finish", "Garage.Qual","Garage.Cond", "Garage.Type")
ames_data.df <- ames_data.df[ , !(names(ames_data.df) %in% drops)]
ames.train_data.df <- ames.train_data.df[ , !(names(ames.train_data.df) %in% drops)]
ames.test_data.df <- ames.test_data.df[ , !(names(ames.test_data.df) %in% drops)]
```

```
na_count <- na_count[na_count$name %notin% drops,]
na_count
```

```
##                 na_count           name
## Bsmt.Exposure         83  Bsmt.Exposure
## BsmtFin.Type.2        81 BsmtFin.Type.2
## Bsmt.Qual             80      Bsmt.Qual
## Bsmt.Cond             80      Bsmt.Cond
## BsmtFin.Type.1        80 BsmtFin.Type.1
## Mas.Vnr.Type          23   Mas.Vnr.Type
## Mas.Vnr.Area          23   Mas.Vnr.Area
## Bsmt.Full.Bath         2 Bsmt.Full.Bath
## Bsmt.Half.Bath         2 Bsmt.Half.Bath
## BsmtFin.SF.1           1   BsmtFin.SF.1
## BsmtFin.SF.2           1   BsmtFin.SF.2
## Bsmt.Unf.SF            1    Bsmt.Unf.SF
## Total.Bsmt.SF          1  Total.Bsmt.SF
## Electrical             1     Electrical
## Garage.Cars            1    Garage.Cars
## Garage.Area            1    Garage.Area
```

```
# the remaining columns that have any NA rows are low in number, so we will clean the data for those rows
ames_data.df <- drop_na(ames_data.df)
ames.train_data.df <- drop_na(ames.train_data.df)
ames.test_data.df <- drop_na(ames.test_data.df)
```

```
# check for significance of categorical variables remaining
categoricalVarsColumnNames <- c("MS.SubClass", "MS.Zoning", "Street", "Lot.Shape", "Land.Contour", "Utilities"

# generic function to test chi square test for a categorical variable
# here we calculate anova for a model with the categorical variable and another model without it
modelWithAllColumns <- glm(price~., data = ames.train_data.df)
chiTest <- function(columnName) {
```

```
  options(scipen = 999)
  ames.train_data.withoutColumnPassed <- ames.train_data.df[ , !(names(ames.train_data.df) %in% c(columnName))]
  modelWithoutColumnPassed <- glm(price~.,data = ames.train_data.withoutColumnPassed)
  aqq <- anova(modelWithAllColumns,modelWithoutColumnPassed,test="Chisq")
  return (aqq$`Pr(>Chi)`[2])
}

# data frame to hold results for each categorical variable
chiTestResults <- data.frame(matrix(ncol = 2, nrow = 0))
x <- c("Column Name", "ChiValue")
colnames(chiTestResults) <- x

# pass each categorical variable in the function
for (i in categoricalVarsColumnNames){
  chiResult <- chiTest(i)
  chiTestResults[nrow(chiTestResults) + 1,] = c(i,chiResult)
}
chiTestResults
```

```
##       Column Name
## 1     MS.SubClass
## 2       MS.Zoning
## 3          Street
## 4       Lot.Shape
## 5    Land.Contour
## 6       Utilities
## 7      Lot.Config
## 8      Land.Slope
## 9    Neighborhood
## 10    Condition.1
## 11    Condition.2
## 12      Bldg.Type
## 13    House.Style
## 14   Overall.Qual
## 15   Overall.Cond
## 16     Year.Built
## 17 Year.Remod.Add
## 18     Roof.Style
## 19      Roof.Matl
## 20   Exterior.1st
## 21   Exterior.2nd
## 22   Mas.Vnr.Type
## 23     Exter.Qual
## 24     Exter.Cond
## 25     Foundation
## 26      Bsmt.Qual
## 27      Bsmt.Cond
## 28  Bsmt.Exposure
## 29 BsmtFin.Type.1
## 30 BsmtFin.Type.2
## 31        Heating
## 32     Heating.QC
## 33    Central.Air
## 34     Electrical
## 35 Bsmt.Full.Bath
## 36 Bsmt.Half.Bath
## 37      Full.Bath
## 38      Half.Bath
## 39  Bedroom.AbvGr
## 40  Kitchen.AbvGr
```

```
## 41   Kitchen.Qual
## 42  TotRms.AbvGrd
## 43     Functional
## 44     Fireplaces
## 45    Garage.Cars
## 46    Paved.Drive
## 47        Mo.Sold
## 48        Yr.Sold
## 49      Sale.Type
## 50 Sale.Condition
##                                                                   ChiValue
## 1                                                      0.00982202464751872
## 2                                                       0.0246571077399909
## 3                                                        0.186374657101112
## 4                                                        0.526404982988352
## 5                                                      0.00125197761757141
## 6                                                        0.155675476270618
## 7                                                      0.00342135752607489
## 8                                                     0.000261848770051797
## 9  0.00000000000000000000000000000000000000000000000017593464109825
## 10                                                  0.00000447360211598997
## 11                                                   0.0000254475544510119
## 12                                                    0.00042927632933306
## 13                                                      0.0541156144863994
## 14                        0.00000000000000000000000154714884427247
## 15                                          0.0000000000000012613501415985
## 16                                           0.00000000000000823936980028913
## 17                                                      0.0639051181681767
## 18                                                      0.0995773508228519
## 19                                                   0.0000575211963484854
## 20                                                     0.00290809291596882
## 21                                                       0.508283476884134
## 22                                               0.00000000211257932325318
## 23                                            0.00000000000350197377580051
## 24                                                       0.593751192776786
## 25                                                       0.525187091801889
## 26                                                0.0000000310825776933468
## 27                                                       0.665398583885502
## 28                                                0.0000000311641856674433
## 29                                                     0.00067901064936481
## 30                                                     0.00856317317319601
## 31                                                       0.577450998515744
## 32                                                        0.40498201910103
## 33                                                      0.0170189736060096
## 34                                                       0.654217728729041
## 35                                                       0.401191146094886
## 36                                                       0.316797893980862
## 37                                                      0.0619593773173535
## 38                                                      0.0413267228201847
## 39                                                     0.00128743769979668
## 40                                                       0.109947315995142
## 41                                         0.0000000000000077755526634036 3
## 42                                                       0.642554626597336
## 43                                                 0.000000135521043203782
## 44                                                      0.0469907281511189
## 45                                                       0.272499605633982
## 46                                                       0.278415048030658
## 47                                                       0.880814862962162
## 48                                                      0.0661603345324412
## 49                                                       0.182562000631321
```

```
## 50                                          0.0809737315739009
```

```
insignificantCategoricalColumns <- chiTestResults[chiTestResults$ChiValue > 0.05,]
insignificantCategoricalColumns
```

```
##       Column Name           ChiValue
## 3          Street  0.186374657101112
## 4       Lot.Shape  0.526404982988352
## 6       Utilities  0.155675476270618
## 13    House.Style 0.0541156144863994
## 17 Year.Remod.Add 0.0639051181681767
## 18     Roof.Style 0.0995773508228519
## 21   Exterior.2nd  0.508283476884134
## 24     Exter.Cond  0.593751192776786
## 25     Foundation  0.525187091801889
## 27      Bsmt.Cond  0.665398583885502
## 31        Heating  0.577450998515744
## 32     Heating.QC   0.40498201910103
## 34     Electrical  0.654217728729041
## 35 Bsmt.Full.Bath  0.401191146094886
## 36 Bsmt.Half.Bath  0.316797893980862
## 37      Full.Bath 0.0619593773173535
## 40  Kitchen.AbvGr  0.109947315995142
## 42  TotRms.AbvGrd  0.642554626597336
## 45    Garage.Cars  0.272499605633982
## 46    Paved.Drive  0.278415048030658
## 47        Mo.Sold  0.880814862962162
## 48        Yr.Sold 0.0661603345324412
## 49      Sale.Type  0.182562000631321
## 50 Sale.Condition 0.0809737315739009
```

```
# Now looking at the insignificant categorical columns to see if we want to keep any that seem to be relevant
# retaining Year.Remod.Add, Roof.Style, Full.Bath, Kitchen.AbvGr, Yr.Sold, Sale.Condition

insignificantCategoricalColumns <- insignificantCategoricalColumns[insignificantCategoricalColumns$`Column Nam
insignificantCategoricalColumns
```

```
##       Column Name           ChiValue
## 3          Street  0.186374657101112
## 4       Lot.Shape  0.526404982988352
## 6       Utilities  0.155675476270618
## 13    House.Style 0.0541156144863994
## 21   Exterior.2nd  0.508283476884134
## 24     Exter.Cond  0.593751192776786
## 25     Foundation  0.525187091801889
## 27      Bsmt.Cond  0.665398583885502
## 31        Heating  0.577450998515744
## 32     Heating.QC   0.40498201910103
## 34     Electrical  0.654217728729041
## 35 Bsmt.Full.Bath  0.401191146094886
## 36 Bsmt.Half.Bath  0.316797893980862
## 42  TotRms.AbvGrd  0.642554626597336
## 45    Garage.Cars  0.272499605633982
## 46    Paved.Drive  0.278415048030658
## 47        Mo.Sold  0.880814862962162
## 49      Sale.Type  0.182562000631321
```

```
# cleaning data for insignificant categorical columns
ames_data.df <- ames_data.df[ , !(names(ames_data.df) %in% insignificantCategoricalColumns$`Column Name`)]
ames.train_data.df <- ames.train_data.df[ , !(names(ames.train_data.df) %in% insignificantCategoricalColumns$`
ames.test_data.df <- ames.test_data.df[ , !(names(ames.test_data.df) %in% insignificantCategoricalColumns$`Col
```

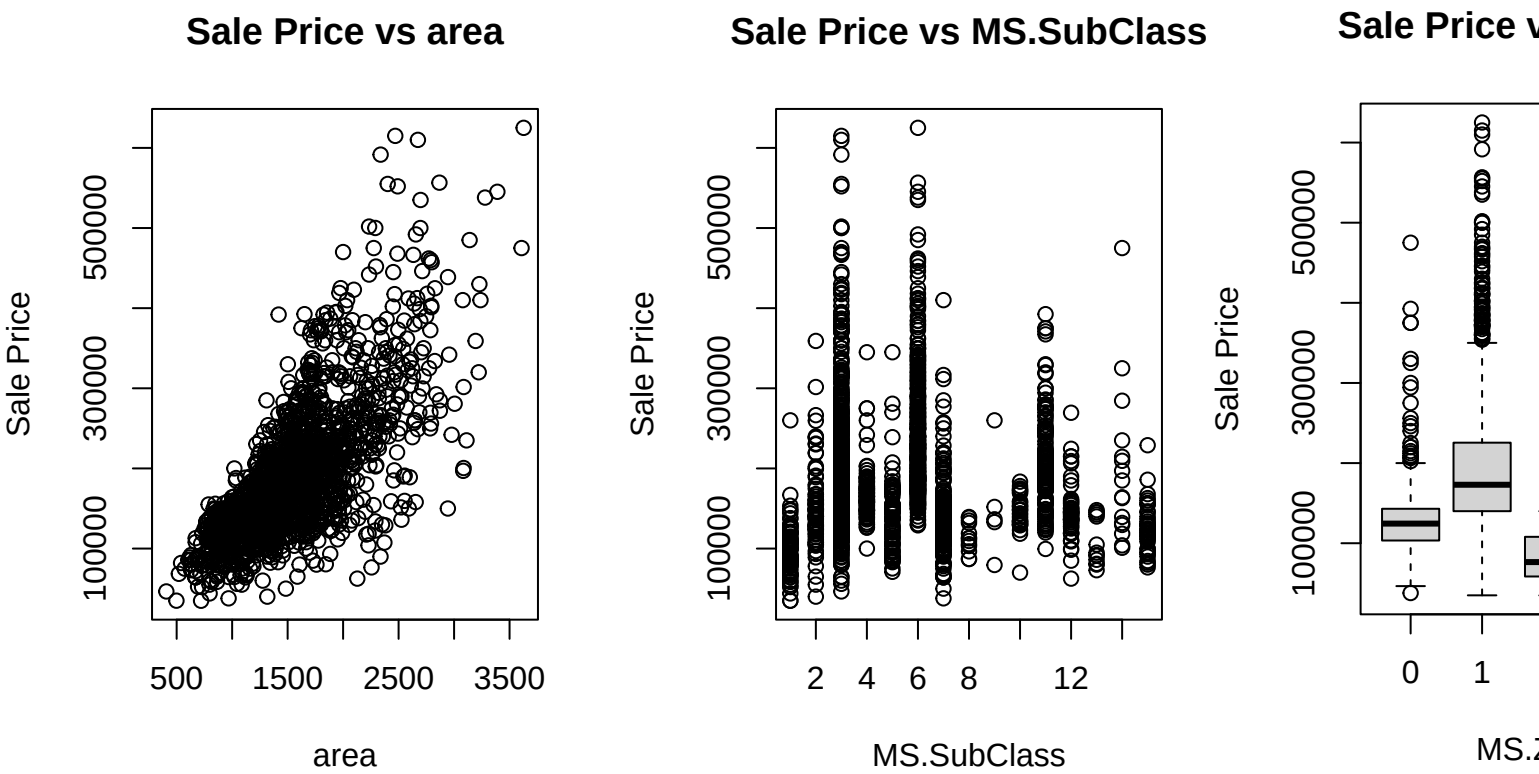
```r
categoricalVarsColumnNames.df<- data.frame(categoricalVarsColumnNames)
categoricalVarsColumnNames.df <- categoricalVarsColumnNames.df[categoricalVarsColumnNames.df$categoricalVarsCo

# function to factor numeric values for categorical variables
factorCatVars <- function(df) {
  c <- unique(df[name])
  levels <- c[,1]
  labels <- c()
  numberOfUniqueValues <- nrow(c)
  for (i in 0:(numberOfUniqueValues-1)) {
    labels <- append(labels, i)
  }
  df[,name] <- factor(df[,name],levels = levels,labels = labels)
  if(length(c[,name])>5){
    df[,name] <- as.numeric(df[,name])
  }
  return (df)
}

for(name in categoricalVarsColumnNames.df){
  ames.train_data.df = factorCatVars(ames.train_data.df)
  ames.test_data.df = factorCatVars(ames.test_data.df)
}
str(ames.train_data.df)
```

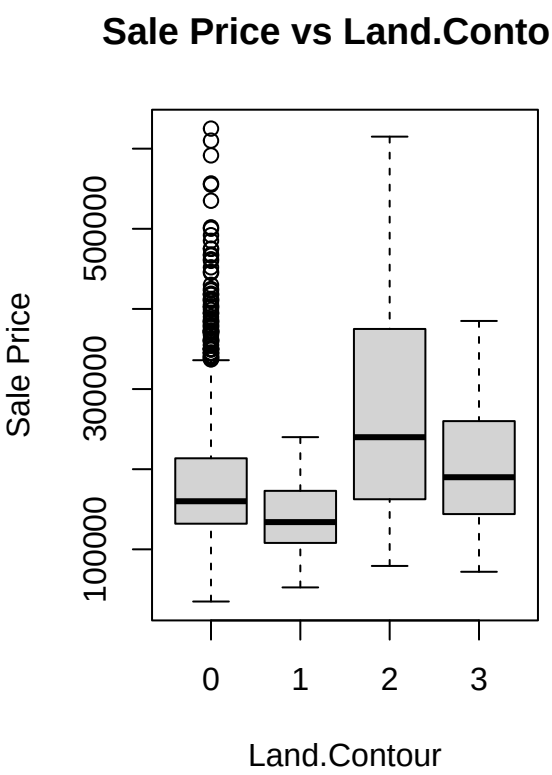
```
## 'data.frame':    1967 obs. of  50 variables:
##  $ area           : num  765 1510 1040 1165 984 ...
##  $ price          : num  87000 108000 123000 174900 110000 ...
##  $ MS.SubClass    : num  1 2 3 4 3 4 3 3 3 5 ...
##  $ MS.Zoning      : Factor w/ 5 levels "0","1","2","3",..: 1 2 2 2 1 2 1 2 2 1 ...
##  $ Lot.Area       : num  5000 9084 11677 9020 9750 ...
##  $ Land.Contour   : Factor w/ 4 levels "0","1","2","3": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Lot.Config     : Factor w/ 5 levels "0","1","2","3",..: 1 1 1 1 1 2 1 1 1 1 ...
##  $ Land.Slope     : Factor w/ 3 levels "0","1","2": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Neighborhood   : num  1 2 3 4 5 4 5 6 3 7 ...
##  $ Condition.1    : num  1 2 3 3 3 2 2 3 3 3 ...
##  $ Condition.2    : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ Bldg.Type      : Factor w/ 5 levels "0","1","2","3",..: 1 1 1 1 1 1 1 1 1 2 ...
##  $ Overall.Qual   : num  1 1 2 3 2 3 3 3 2 1 ...
##  $ Overall.Cond   : num  1 2 3 1 4 2 1 4 2 1 ...
##  $ Year.Built     : num  1 2 3 3 4 5 6 7 5 8 ...
##  $ Year.Remod.Add : num  1 1 2 2 3 4 5 6 4 7 ...
##  $ Roof.Style     : num  1 1 1 1 1 1 1 2 2 1 ...
##  $ Roof.Matl      : Factor w/ 5 levels "0","1","2","3",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Exterior.1st   : num  1 1 2 2 1 3 1 4 2 5 ...
##  $ Mas.Vnr.Type   : Factor w/ 5 levels "0","1","2","3",..: 1 1 2 2 2 2 1 2 1 1 ...
##  $ Mas.Vnr.Area   : num  0 0 442 183 164 399 0 196 0 0 ...
##  $ Exter.Qual     : Factor w/ 4 levels "0","1","2","3": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Bsmt.Qual      : Factor w/ 5 levels "0","1","2","3",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Bsmt.Exposure  : Factor w/ 4 levels "0","1","2","3": 1 2 3 4 1 4 1 1 1 1 ...
##  $ BsmtFin.Type.1 : num  1 2 1 1 1 1 3 4 5 2 ...
##  $ BsmtFin.SF.1   : num  188 0 249 312 200 672 600 888 68 0 ...
##  $ BsmtFin.Type.2 : num  1 1 2 3 1 3 1 1 2 1 ...
##  $ BsmtFin.SF.2   : num  0 0 761 539 0 690 0 0 884 0 ...
##  $ Bsmt.Unf.SF    : num  577 755 30 276 784 0 312 228 28 546 ...
##  $ Total.Bsmt.SF  : num  765 755 1040 1127 984 ...
##  $ Central.Air    : Factor w/ 2 levels "0","1": 1 2 2 2 2 2 2 2 2 2 ...
##  $ X1st.Flr.SF    : num  765 755 1040 1165 984 ...
##  $ X2nd.Flr.SF    : num  0 755 0 0 0 0 0 0 0 546 ...
##  $ Low.Qual.Fin.SF: num  0 0 0 0 0 0 0 0 0 0 ...
```

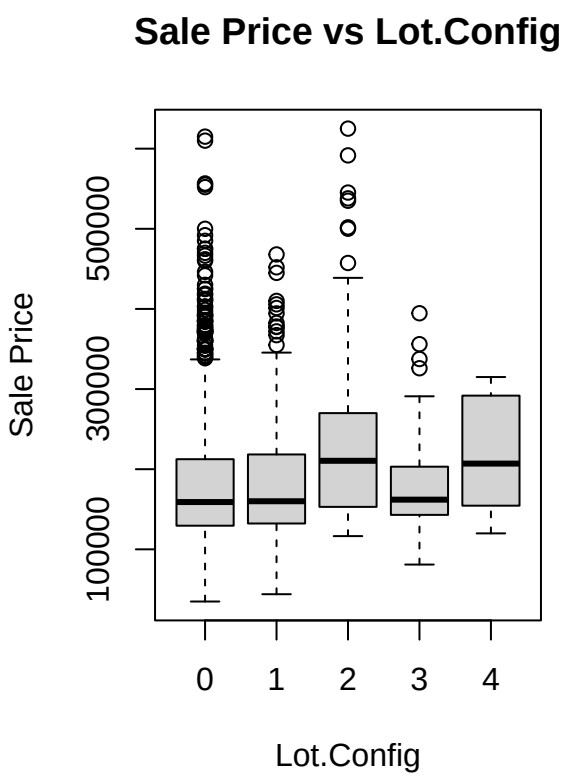
```
##  $ Full.Bath      : Factor w/ 5 levels "0","1","2","3",..: 1 1 1 1 1 1 1 2 1 1 ...
##  $ Half.Bath      : Factor w/ 3 levels "0","1","2": 1 1 1 2 1 1 1 1 1 2 ...
##  $ Bedroom.AbvGr  : num  1 2 3 3 1 3 1 3 3 3 ...
##  $ Kitchen.AbvGr  : Factor w/ 4 levels "0","1","2","3": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Kitchen.Qual   : Factor w/ 4 levels "0","1","2","3": 1 2 2 2 3 2 2 2 2 2 ...
##  $ Functional     : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ Fireplaces     : Factor w/ 5 levels "0","1","2","3",..: 1 2 1 1 1 2 1 2 1 1 ...
##  $ Garage.Area    : num  200 296 264 490 308 884 923 528 400 0 ...
##  $ Wood.Deck.SF   : num  135 120 0 0 0 0 0 0 0 0 ...
##  $ Open.Porch.SF  : num  0 0 90 129 0 0 158 0 28 0 ...
##  $ Enclosed.Porch : num  41 0 0 0 0 252 158 0 0 0 ...
##  $ X3Ssn.Porch    : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Screen.Porch   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Pool.Area      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Yr.Sold        : Factor w/ 5 levels "0","1","2","3",..: 1 2 3 4 4 2 1 2 2 5 ...
##  $ Sale.Condition : num  1 1 1 1 1 1 1 1 1 1 ...
```

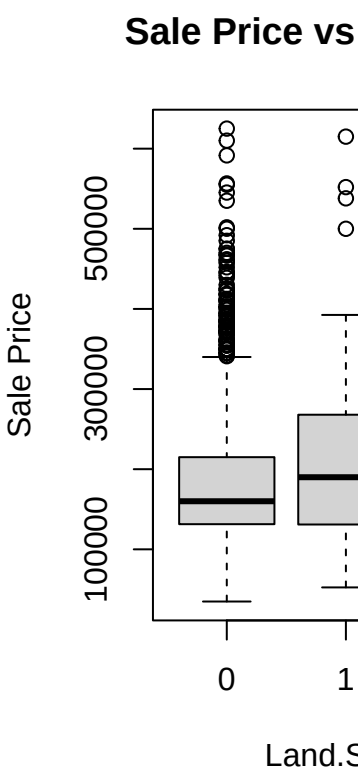
5. Create scatter plots and a correlation matrix for the train data set. Interpret the possible relationship between the response and the covariates.

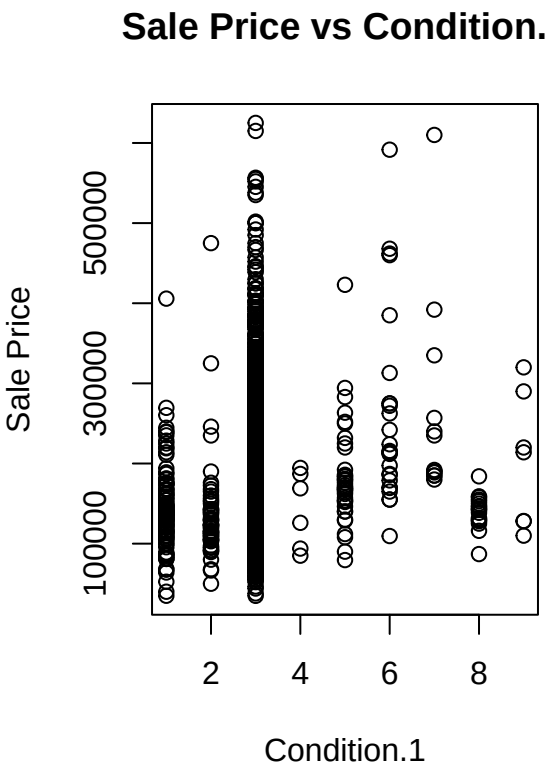
```
predictorVars <- ames.train_data.df[ ,!(names(ames.train_data.df) %in% c("price"))]
par(mfrow=c(1,2))
for (i in 1:ncol(predictorVars)) {
  columnName <- colnames(predictorVars[i])
  mainText <- paste("Sale Price vs", columnName, sep=" ")
  plot(predictorVars[,i],xlab = columnName, ames.train_data.df$price, ylab = "Sale Price", main = mainText)
}
```

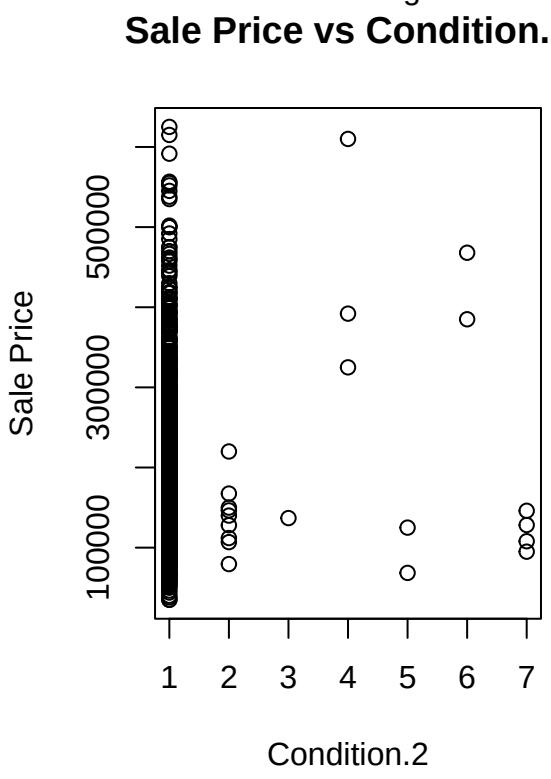


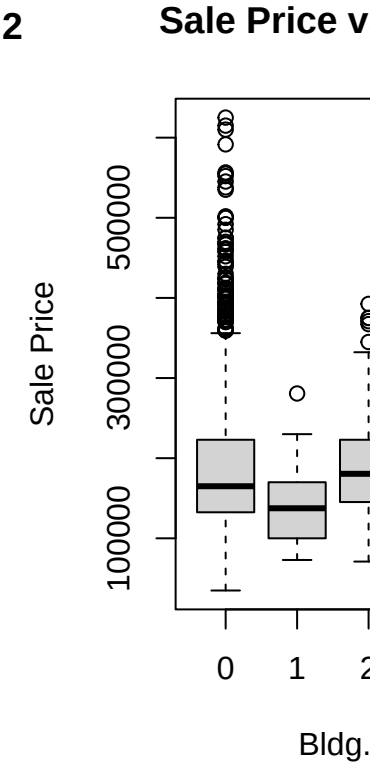

Sale Price vs Land.Contour
Sale Price
100000
300000
500000
0
1
2
3
Land.Contour

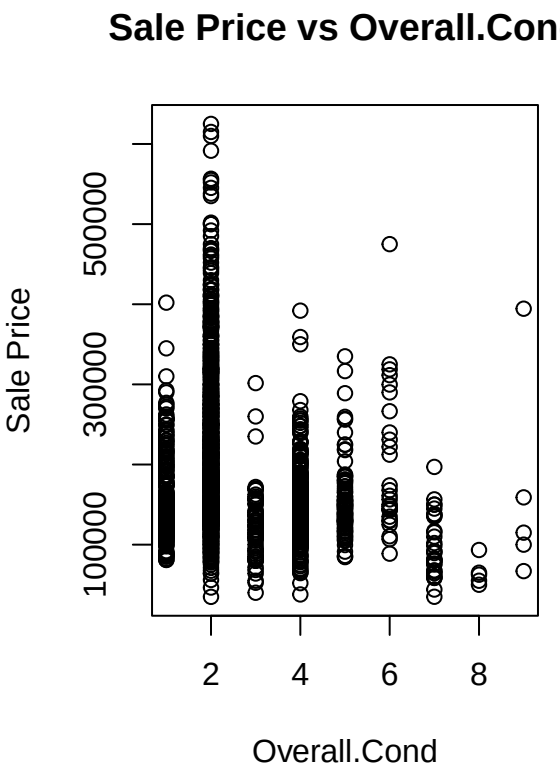

Sale Price vs Lot.Config
Sale Price
100000
300000
500000
0
1
2
3
4
Lot.Config

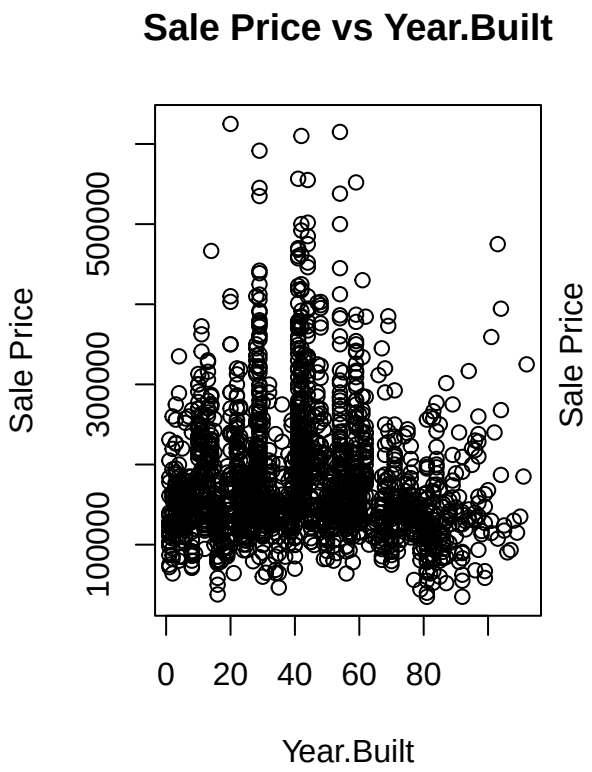

Sale Price vs
Sale Price
100000
300000
500000
0
1
Land.S

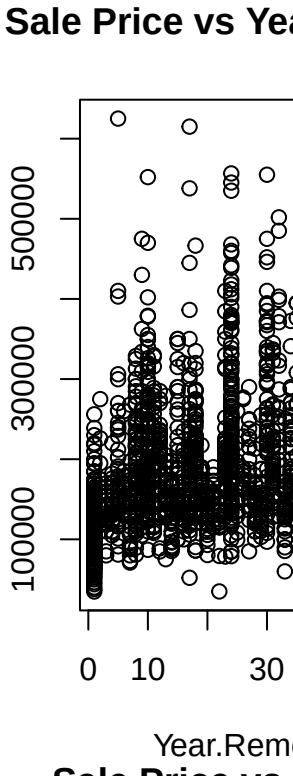

Sale Price vs Condition.1
Sale Price
100000
300000
500000
2
4
6
8
Condition.1

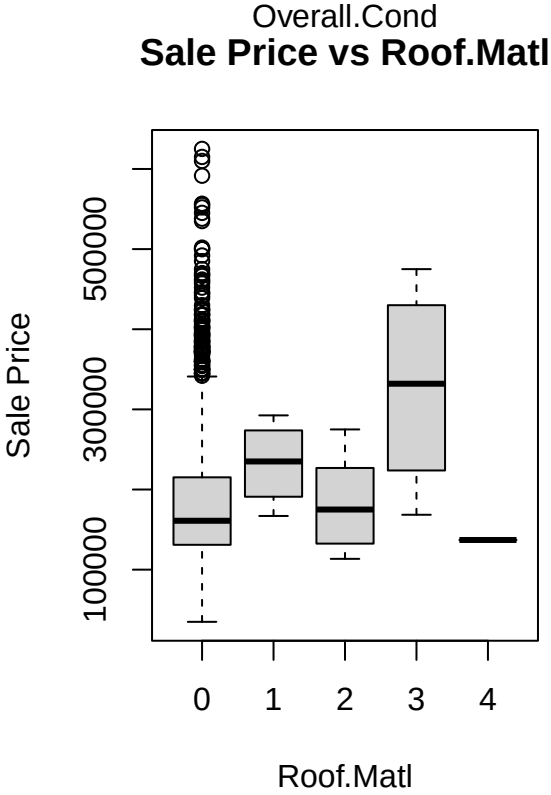

Sale Price vs Condition.2
Sale Price
100000
300000
500000
1
2
3
4
5
6
7
Condition.2

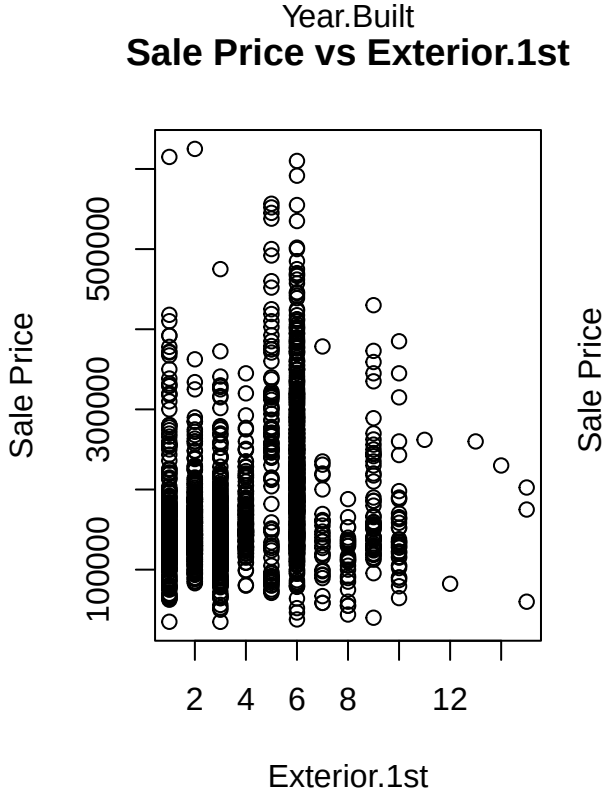

Sale Price v
Sale Price
100000
300000
500000
0
1
Bldg.

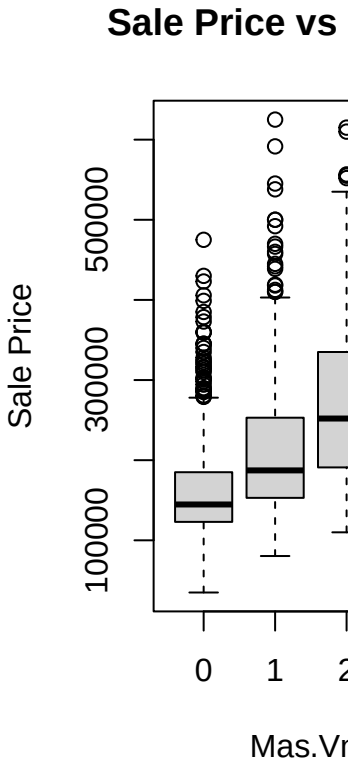

Sale Price vs Overall.Cond
Sale Price
100000
300000
500000
2
4
6
8
Overall.Cond

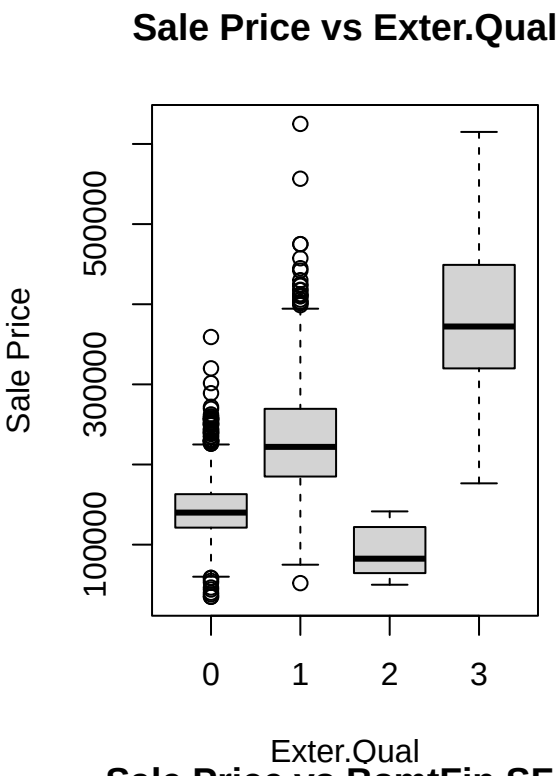

Sale Price vs Year.Built
Sale Price
100000
300000
500000
0
20
40
60
80
Year.Built

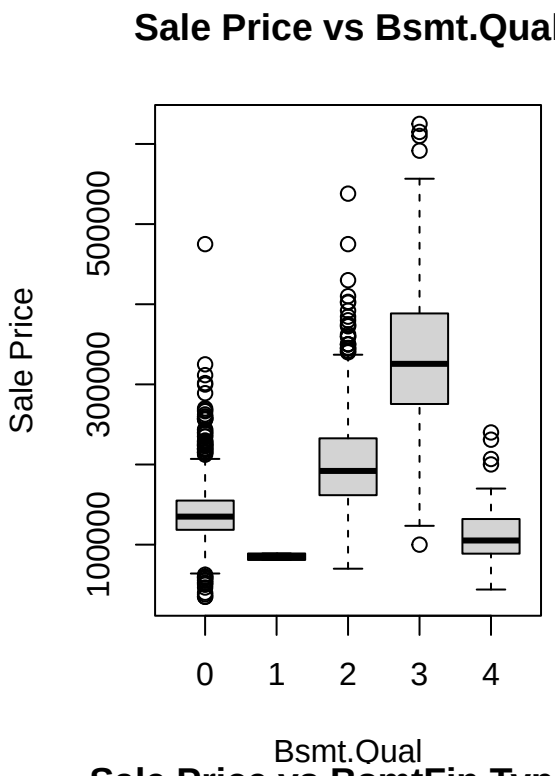

Sale Price vs Yea
Sale Price
100000
300000
500000
0
10
30
Year.Rem

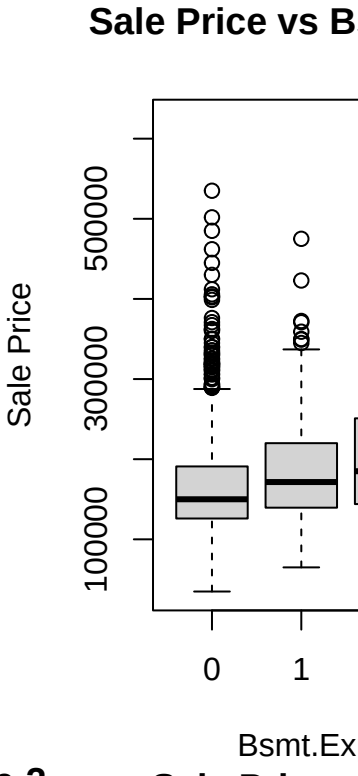

Sale Price vs Roof.Matl
Sale Price
100000
300000
500000
0
1
2
3
4
Roof.Matl

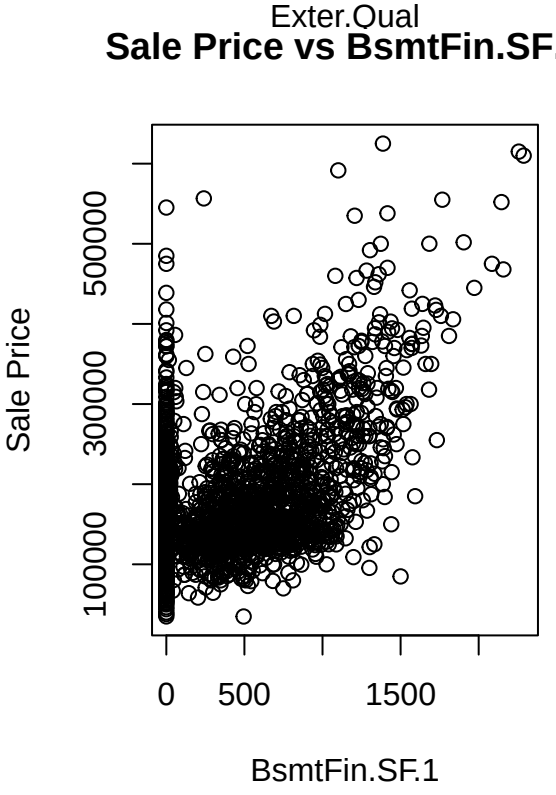

Sale Price vs Exterior.1st
Sale Price
100000
300000
500000
2
4
6
8
12
Exterior.1st

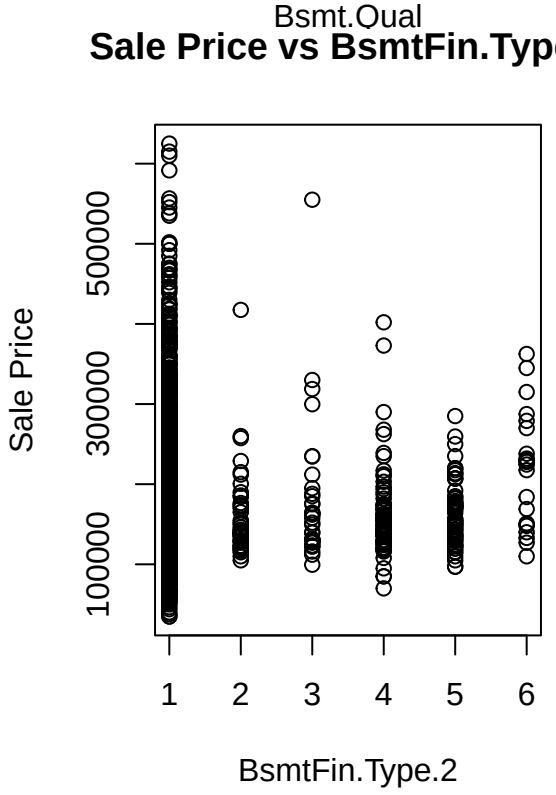

Sale Price vs
Sale Price
100000
300000
500000
0
1
Mas.Vr

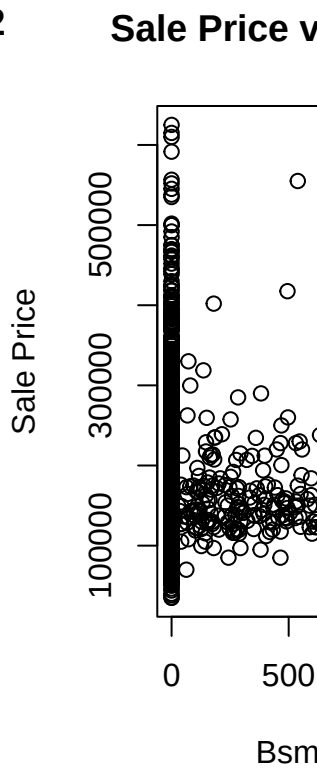

Sale Price vs Exter.Qual
Sale Price
100000
300000
500000
0
1
2
3
Exter.Qual

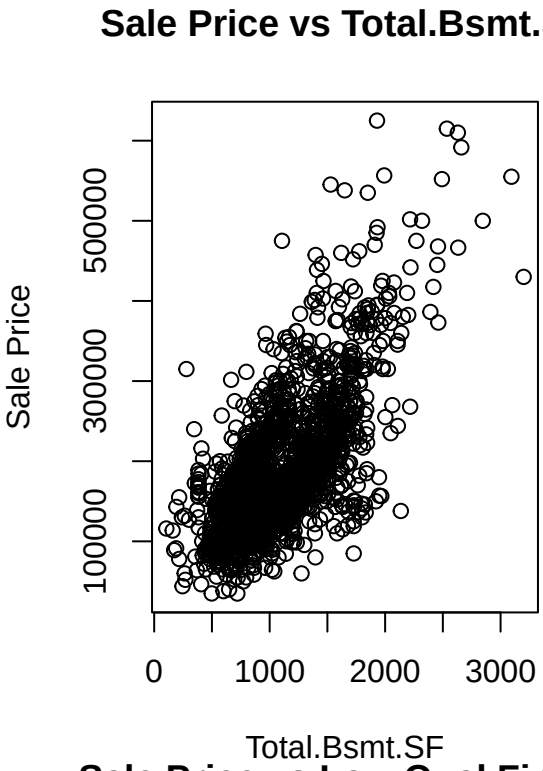

Sale Price vs Bsmt.Qual
Sale Price
100000
300000
500000
0
1
2
3
4
Bsmt.Qual

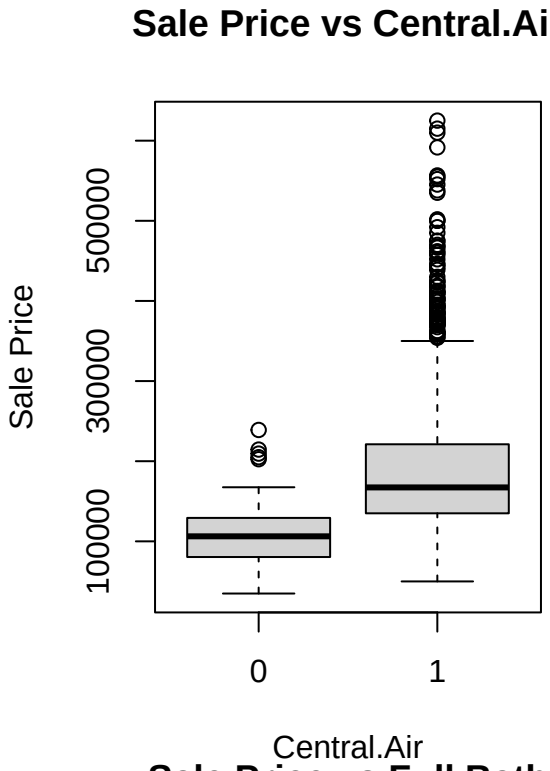

Sale Price vs B
Sale Price
100000
300000
500000
0
1
Bsmt.Ex

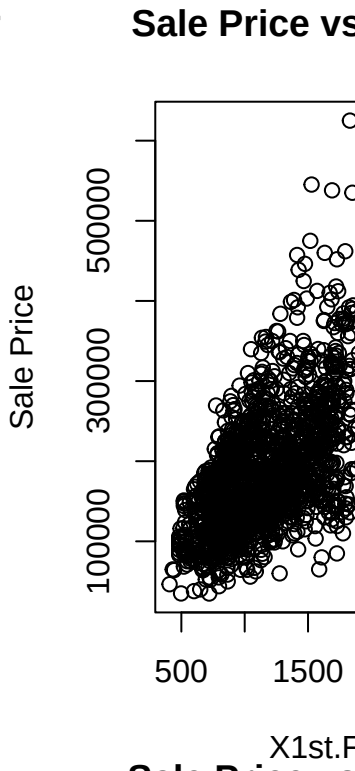

Sale Price vs BsmtFin.SF.1
Sale Price
100000
300000
500000
0
500
1500
BsmtFin.SF.1

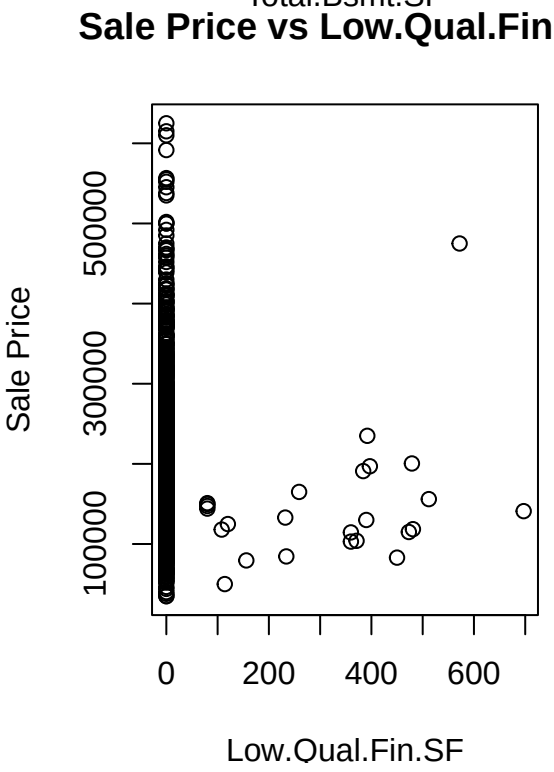

Sale Price vs BsmtFin.Type.2
Sale Price
100000
300000
500000
1
2
3
4
5
6
BsmtFin.Type.2

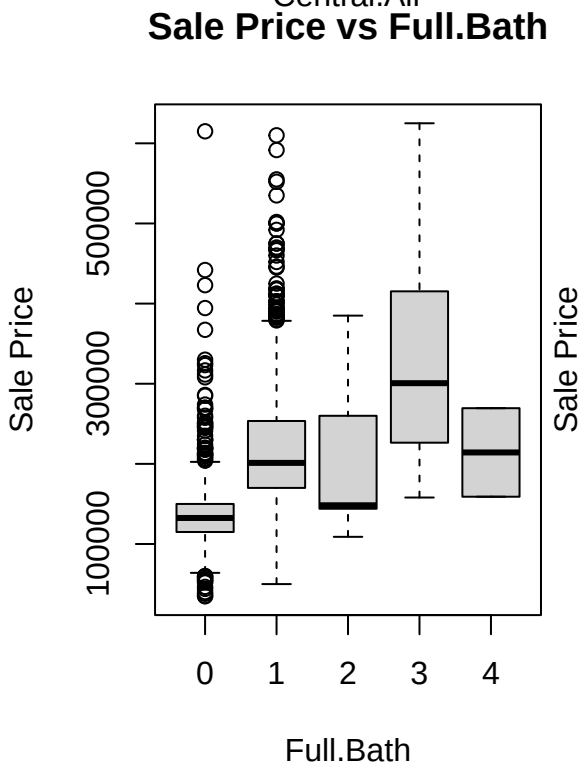

Sale Price v
Sale Price
100000
300000
500000
0
500
Bsm

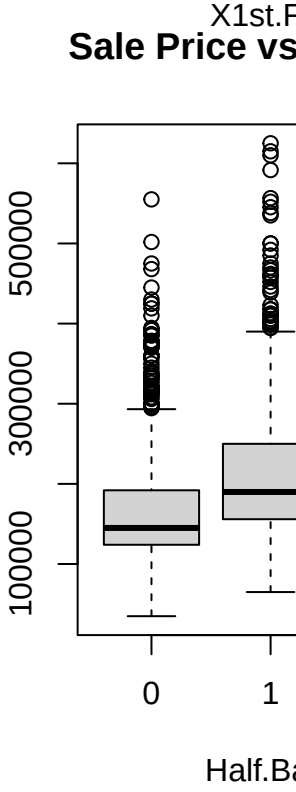

Sale Price vs Total.Bsmt.SF
Sale Price
100000
300000
500000
0
1000
2000
3000
Total.Bsmt.SF

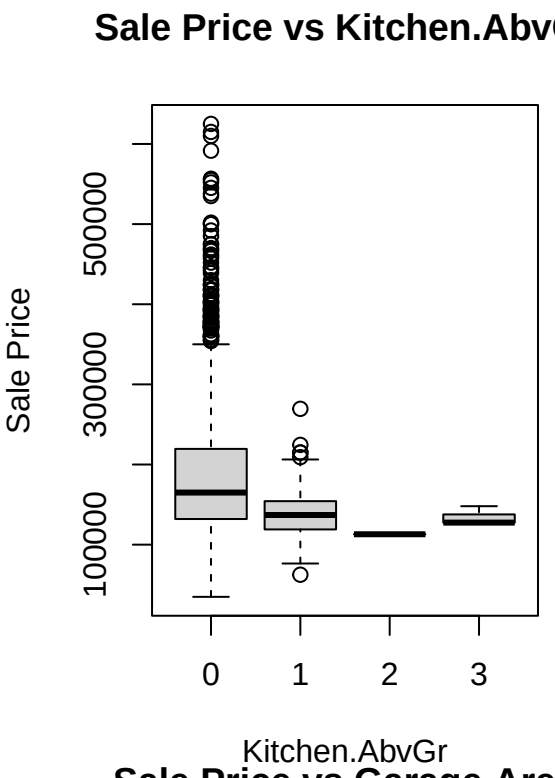

Sale Price vs Central.Air
Sale Price
100000
300000
500000
0
1
Central.Air

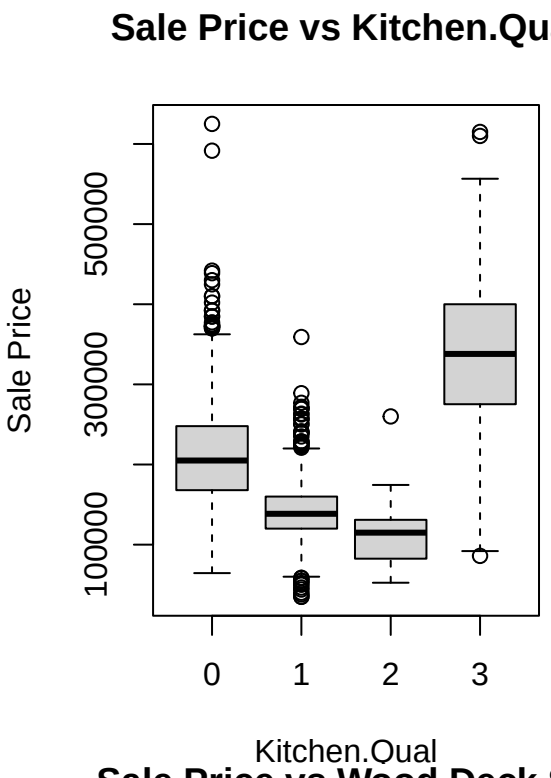

Sale Price vs
Sale Price
100000
300000
500000
500
1500
X1st.F

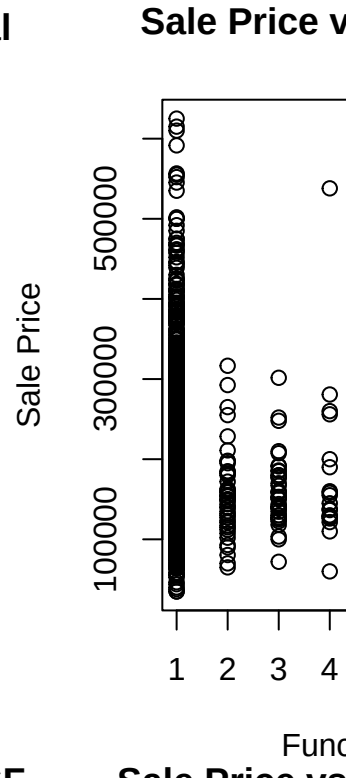

Sale Price vs Low.Qual.Fin.SF
Sale Price
100000
300000
500000
0
200
400
600
Low.Qual.Fin.SF

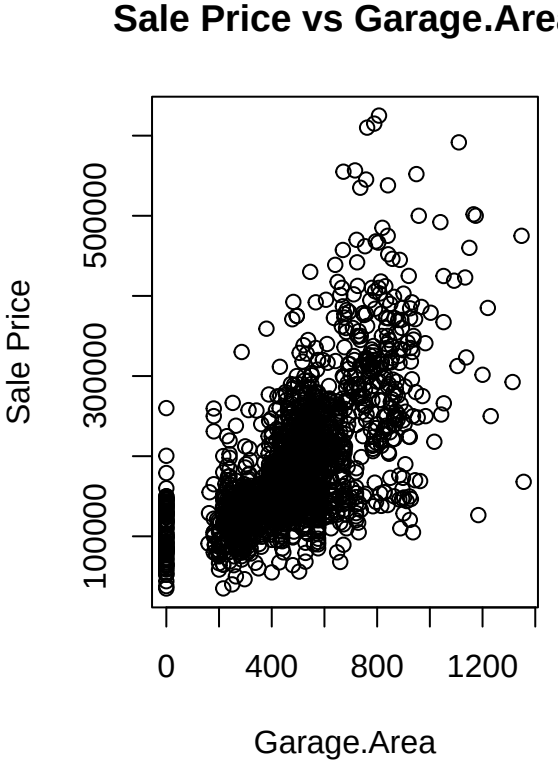

Sale Price vs Full.Bath
Sale Price
100000
300000
500000
0
1
2
3
4
Full.Bath
Sale Price

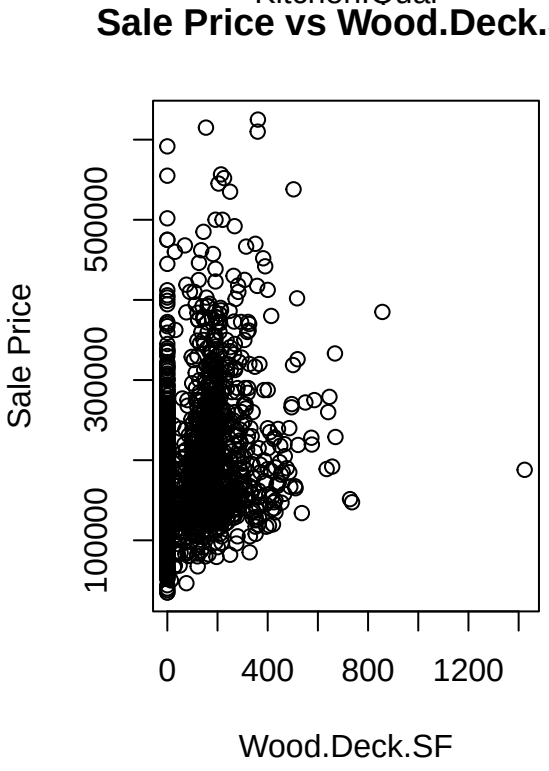

Sale Price vs
100000
300000
500000
0
1
Half.Ba

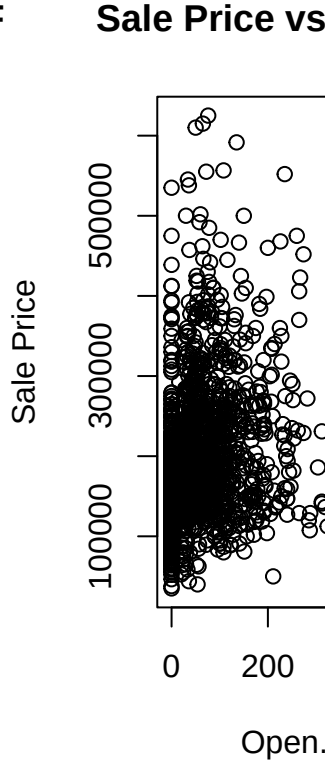

Sale Price vs Kitchen.AbvGr
Sale Price
100000
300000
500000
0
1
2
3
Kitchen.AbvGr

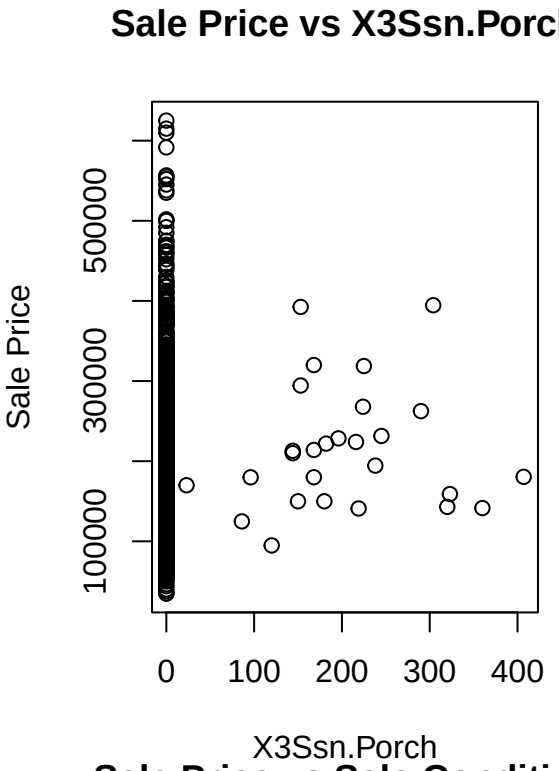

Sale Price vs Kitchen.Qual
Sale Price
100000
300000
500000
0
1
2
3
Kitchen.Qual

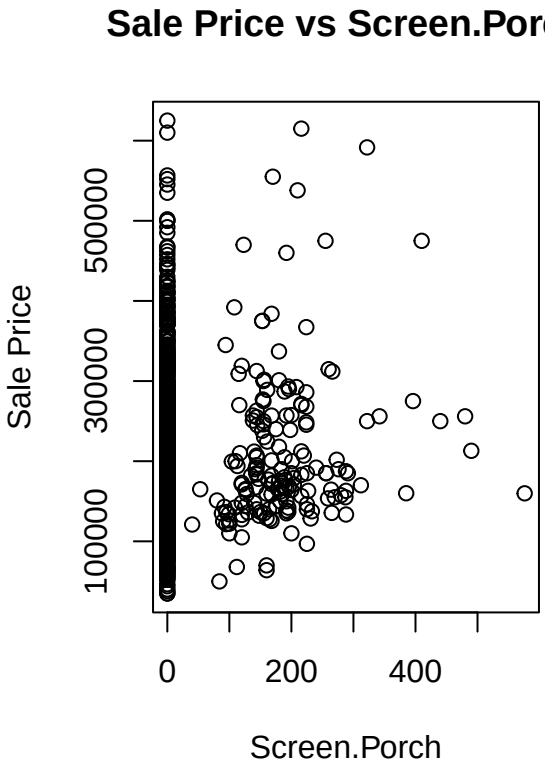

Sale Price v
Sale Price
100000
300000
500000
1
2
3
4
Func

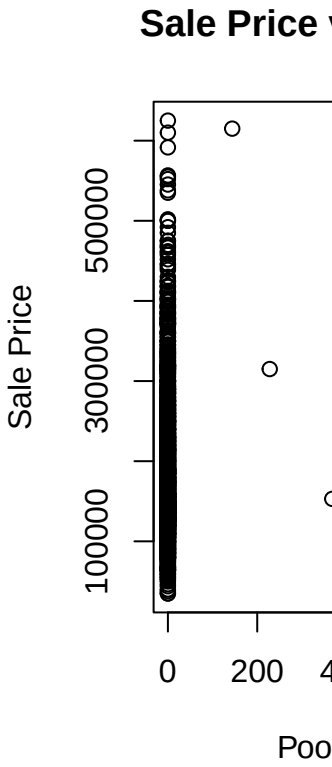

Sale Price vs Garage.Area
Sale Price
100000
300000
500000
0
400
800
1200
Garage.Area

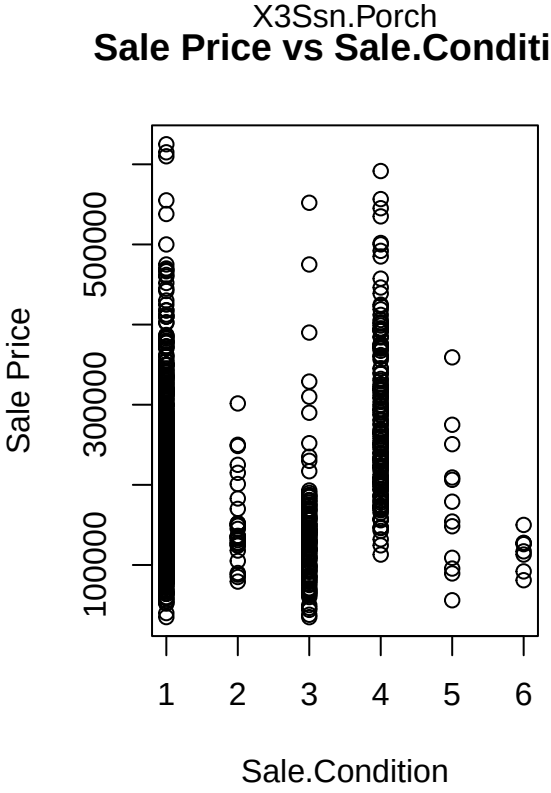

Sale Price vs Wood.Deck.SF
Sale Price
100000
300000
500000
0
400
800
1200
Wood.Deck.SF


Sale Price vs
Sale Price
100000
300000
500000
0
200
Open.



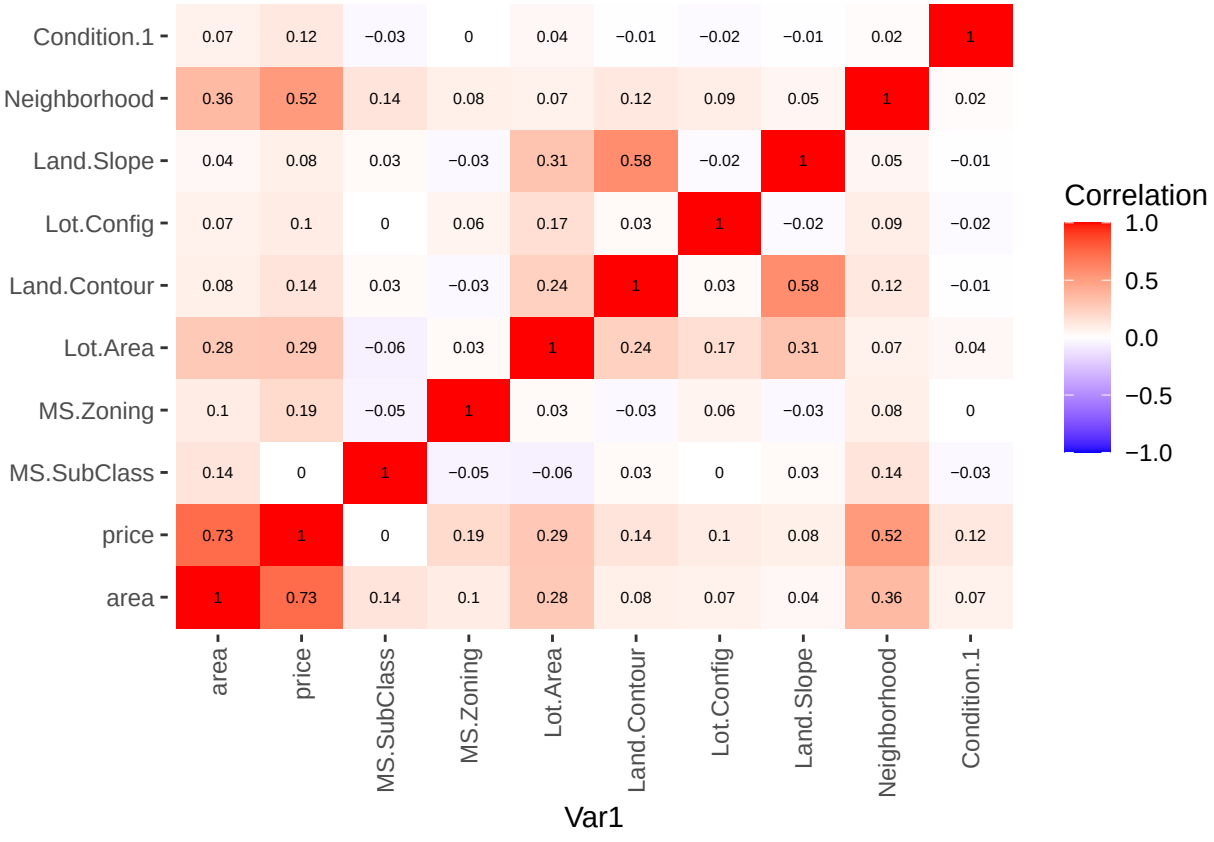


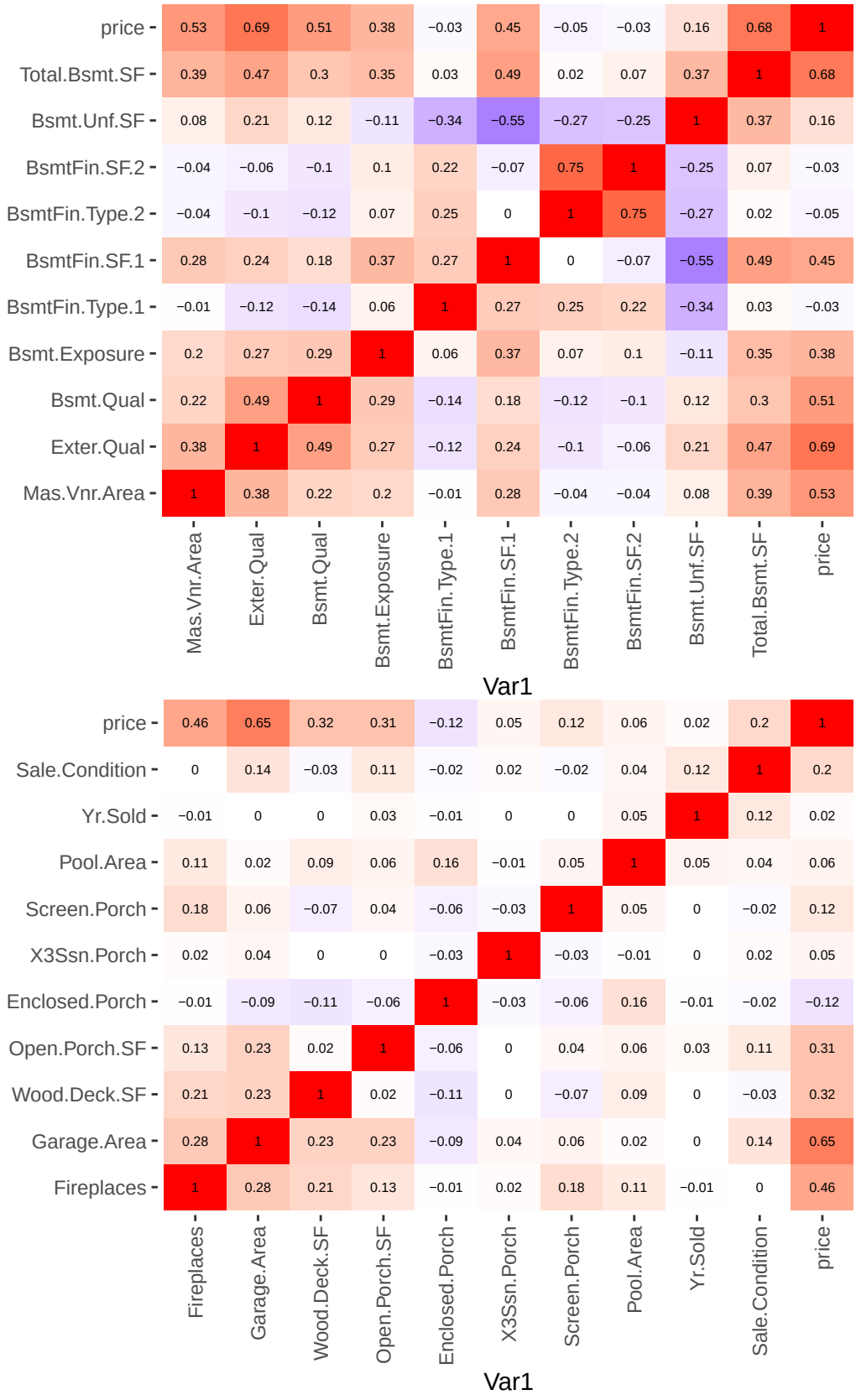


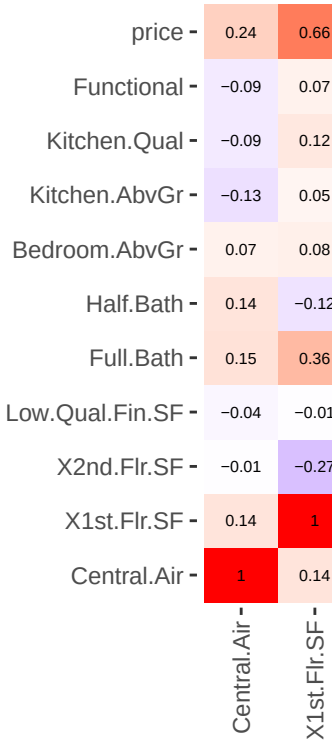



```
temp.df <- ames.train_data.df
for(name in categoricalVarsColumnNames.df){
  temp.df[,name] <- as.numeric(temp.df[,name])
}
totalColumns <- ncol(temp.df)
step_size <- 10
startIndex <- 1
endIndex <- step_size

# function that will be called multiple times to print correlation matrices
printCorrelationMatrix <- function(start,end) {
  temp1.df <- data.frame(temp.df[start:end])
```

```
  tempColumnNames <- colnames(temp1.df)
  if("price" %notin% tempColumnNames){
    temp1.df$price <- temp.df$price
  }
  Correlations<-round(cor(temp1.df),2)
  melted_cormat <- melt(Correlations)
  print(ggplot(data = melted_cormat, aes(x = Var1, y = Var2, fill = value)) +
    geom_tile() +
    geom_text(aes(Var2, Var1, label = value), size = 2) +
    scale_fill_gradient2(low = "blue", high = "red", limit = c(-1, 1), name = "Correlation") +
    theme(axis.title.x = element_text(), axis.title.y = element_blank(), axis.text.x = element_text(angle = 90
}
# end function

while(endIndex <= totalColumns){
  printCorrelationMatrix(startIndex,endIndex)
  startIndex <- startIndex + step_size
  endIndex <- endIndex + step_size
  if(endIndex > totalColumns & startIndex < totalColumns){
    printCorrelationMatrix(startIndex,totalColumns)
  }
}
```







6. Build several multiple linear models by using the stepwise selection methods. Compare the performance of the best two linear models.

```
# Model Fitting
# Build full model with all predictors
fm <- lm(price ~., data = ames.train_data.df)

# Build null model with no predictors
```

```r
nm <- lm(price ~ 1, data = ames.train_data.df)

# Stepwise selection using both AIC and BIC
sm_both_aic <- step(fm, direction = "both", scope = list(lower = nm, upper = fm), trace = FALSE, k = 2)
sm_both_bic <- step(fm, direction = "both", scope = list(lower = nm, upper = fm), trace = FALSE, k = log(nrow(

# Stepwise selection using p-value significance level
sm_both_p <- ols_step_both_p(fm, p_enter = 0.05, p_remove = 0.05, details = FALSE)

# Display Summary of Stepwise models
# Summary of the model using both AIC
cat("Summary of Stepwise Model (Both Directions - AIC):\n")
```

```
## Summary of Stepwise Model (Both Directions - AIC):
```

```r
summary(sm_both_aic)
```

```
##
## Call:
## lm(formula = price ~ MS.Zoning + Lot.Area + Land.Contour + Lot.Config +
##     Land.Slope + Neighborhood + Condition.1 + Bldg.Type + Overall.Qual +
##     Overall.Cond + Year.Built + Year.Remod.Add + Roof.Matl +
##     Mas.Vnr.Type + Mas.Vnr.Area + Exter.Qual + Bsmt.Qual + Bsmt.Exposure +
##     BsmtFin.SF.1 + BsmtFin.Type.2 + BsmtFin.SF.2 + Bsmt.Unf.SF +
##     Central.Air + X1st.Flr.SF + X2nd.Flr.SF + Full.Bath + Half.Bath +
##     Bedroom.AbvGr + Kitchen.AbvGr + Kitchen.Qual + Functional +
##     Fireplaces + Garage.Area + Wood.Deck.SF + Screen.Porch +
##     Sale.Condition, data = ames.train_data.df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -160135  -12573     871   12283  146136
##
## Coefficients:
##                    Estimate   Std. Error t value             Pr(>|t|)
## (Intercept)      16262.2225    5138.3066   3.165             0.001576 **
## MS.Zoning1        3348.5073    1794.8868   1.866             0.062254 .
## MS.Zoning2      -22162.8737    6225.5354  -3.560             0.000380 ***
## MS.Zoning3       19340.6244    3161.9869   6.117 0.000000001158272028 ***
## MS.Zoning4       -2982.3376    5824.3504  -0.512             0.608678
## Lot.Area             0.4399       0.1047   4.200 0.000027874117901285 ***
## Land.Contour1    -4094.8373    3116.1184  -1.314             0.188977
## Land.Contour2    13994.4744    3019.5879   4.635 0.000003820875067584 ***
## Land.Contour3    -9974.0231    4831.6194  -2.064             0.039123 *
## Lot.Config1      -3701.1657    1469.5152  -2.519             0.011863 *
## Lot.Config2       6848.2737    2314.3966   2.959             0.003125 **
## Lot.Config3      -5006.2156    3149.3215  -1.590             0.112088
## Lot.Config4      -2976.1831    7241.0132  -0.411             0.681106
## Land.Slope1        347.9820    3337.6107   0.104             0.916973
## Land.Slope2     -27006.8477    9753.3697  -2.769             0.005678 **
## Neighborhood       945.9472     121.2556   7.801 0.000000000000010048 ***
## Condition.1       1088.2544     578.3533   1.882             0.060038 .
## Bldg.Type1      -20501.7536    3372.5142  -6.079 0.000000001458109523 ***
## Bldg.Type2      -14856.2101    2393.2215  -6.208 0.000000000659230076 ***
## Bldg.Type3       -7002.3190    5635.1464  -1.243             0.214163
## Bldg.Type4      -13526.4867    4577.4597  -2.955             0.003165 **
## Overall.Qual      4594.8305     522.3089   8.797 < 0.0000000000000002 ***
## Overall.Cond     -1669.5926     424.3817  -3.934 0.000086493364536119 ***
## Year.Built         -70.7364      23.4999  -3.010             0.002646 **
## Year.Remod.Add      57.8324      37.0380   1.561             0.118589
```

```
## Roof.Matl1       -7783.2126    8692.3745  -0.895             0.370683
## Roof.Matl2       -3111.9392    6493.0753  -0.479             0.631802
## Roof.Matl3       55636.0348   13033.9958   4.269 0.000020648710567730 ***
## Roof.Matl4      -14904.4835   24076.8646  -0.619             0.535966
## Mas.Vnr.Type1    -6717.6324    1701.8462  -3.947 0.000081937875556025 ***
## Mas.Vnr.Type2     5184.4695    2438.0288   2.127             0.033591 *
## Mas.Vnr.Type3   -16739.9770    6187.5672  -2.705             0.006883 **
## Mas.Vnr.Type4  -100248.0830   23869.0365  -4.200 0.000027939824523420 ***
## Mas.Vnr.Area        29.1664       4.6980   6.208 0.000000000656430099 ***
## Exter.Qual1      11640.4044    1907.8798   6.101 0.000000001273050716 ***
## Exter.Qual2     -13123.7930    6131.6603  -2.140             0.032455 *
## Exter.Qual3      41968.8672    4191.1033  10.014 < 0.0000000000000002 ***
## Bsmt.Qual1       22437.2711   17909.8314   1.253             0.210437
## Bsmt.Qual2        6379.2995    1700.2108   3.752             0.000181 ***
## Bsmt.Qual3       28570.4330    3100.7270   9.214 < 0.0000000000000002 ***
## Bsmt.Qual4       -2038.4352    3235.8034  -0.630             0.528795
## Bsmt.Exposure1   -1168.9200    1982.2361  -0.590             0.555464
## Bsmt.Exposure2    5533.6349    1697.4069   3.260             0.001134 **
## Bsmt.Exposure3   14799.5761    2229.3133   6.639 0.000000000041222152 ***
## BsmtFin.SF.1        43.0822       3.3563  12.836 < 0.0000000000000002 ***
## BsmtFin.Type.2   -1192.6584     773.6142  -1.542             0.123321
## BsmtFin.SF.2        32.1546       5.6839   5.657 0.000000017740266811 ***
## Bsmt.Unf.SF         22.2379       3.2887   6.762 0.000000000018086985 ***
## Central.Air1      7027.3390    2706.2015   2.597             0.009484 **
## X1st.Flr.SF         57.0136       3.5254  16.172 < 0.0000000000000002 ***
## X2nd.Flr.SF         52.4948       2.3711  22.139 < 0.0000000000000002 ***
## Full.Bath1        -266.9180    1741.5490  -0.153             0.878206
## Full.Bath2       14555.4233    9231.1115   1.577             0.115013
## Full.Bath3       24725.6976    4609.4412   5.364 0.000000091304601092 ***
## Full.Bath4       31564.1394   18299.7096   1.725             0.084719 .
## Half.Bath1        4641.1220    1620.4929   2.864             0.004229 **
## Half.Bath2      -30179.2237    7197.8505  -4.193 0.000028822169669591 ***
## Bedroom.AbvGr    -2040.2978     525.2243  -3.885             0.000106 ***
## Kitchen.AbvGr1  -17498.3911    5454.3244  -3.208             0.001358 **
## Kitchen.AbvGr2   15636.2844   24093.5467   0.649             0.516429
## Kitchen.AbvGr3   -9594.1783   14828.0906  -0.647             0.517693
## Kitchen.Qual1    -9409.1693    1593.4424  -5.905 0.000000004171426646 ***
## Kitchen.Qual2    -8389.5272    4135.2693  -2.029             0.042621 *
## Kitchen.Qual3    21728.2926    2880.1483   7.544 0.000000000000070251 ***
## Functional       -7344.0220     891.2560  -8.240 0.000000000000000317 ***
## Fireplaces1       3371.9028    1326.5272   2.542             0.011104 *
## Fireplaces2      10609.2918    2430.5616   4.365 0.000013400393881788 ***
## Fireplaces3      -4784.2646    8141.4666  -0.588             0.556843
## Fireplaces4     -27454.5918   24437.2948  -1.123             0.261380
## Garage.Area         27.1080       3.4333   7.896 0.000000000000004852 ***
## Wood.Deck.SF        10.8905       4.7081   2.313             0.020821 *
## Screen.Porch        38.1948       9.6775   3.947 0.000082102610642733 ***
## Sale.Condition    1180.6302     578.7996   2.040             0.041510 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 23430 on 1894 degrees of freedom
## Multiple R-squared:  0.9172, Adjusted R-squared:  0.9141
## F-statistic: 291.5 on 72 and 1894 DF,  p-value: < 0.00000000000000022
```

```
cat("AIC of Model (Both Directions - AIC):", AIC(sm_both_aic), "\n\n")
```

```
## AIC of Model (Both Directions - AIC): 45239.46
```

```
# Summary of the model using both BIC
cat("Summary of Stepwise Model (Both Directions - BIC):\n")
```

Summary of Stepwise Model (Both Directions - BIC):

```
summary(sm_both_bic)
```

```
##
## Call:
## lm(formula = price ~ area + MS.Zoning + Lot.Area + Land.Contour +
##     Neighborhood + Bldg.Type + Overall.Qual + Overall.Cond +
##     Year.Built + Mas.Vnr.Type + Mas.Vnr.Area + Exter.Qual + Bsmt.Qual +
##     Bsmt.Exposure + BsmtFin.SF.1 + BsmtFin.SF.2 + Bsmt.Unf.SF +
##     Central.Air + Full.Bath + Half.Bath + Bedroom.AbvGr + Kitchen.Qual +
##     Functional + Garage.Area + Screen.Porch, data = ames.train_data.df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -148781  -13155     817   12446  149300
##
## Coefficients:
##                    Estimate   Std. Error t value             Pr(>|t|)
## (Intercept)     17252.61507   4642.00670   3.717             0.000208 ***
## area               54.15963      2.10149  25.772 < 0.0000000000000002 ***
## MS.Zoning1       5046.11725   1782.20129   2.831             0.004683 **
## MS.Zoning2     -22208.74000   6338.66867  -3.504             0.000469 ***
## MS.Zoning3      19532.49337   3202.74821   6.099   0.0000000012902880 ***
## MS.Zoning4      -3766.62316   5910.49937  -0.637             0.524021
## Lot.Area            0.43671      0.09274   4.709   0.0000026671556936 ***
## Land.Contour1   -4252.04733   3076.13610  -1.382             0.167050
## Land.Contour2   14414.56377   2872.75774   5.018   0.0000005713309523 ***
## Land.Contour3   -9403.98049   4309.93754  -2.182             0.029236 *
## Neighborhood      938.86621    122.23702   7.681   0.0000000000000251 ***
## Bldg.Type1     -21031.45010   3393.79458  -6.197   0.0000000007024999 ***
## Bldg.Type2     -14824.71322   2395.66740  -6.188   0.0000000007424800 ***
## Bldg.Type3     -23544.67762   3799.88639  -6.196   0.0000000007063501 ***
## Bldg.Type4     -20752.04814   4074.78529  -5.093   0.0000003874280305 ***
## Overall.Qual     4889.37170    526.76910   9.282 < 0.0000000000000002 ***
## Overall.Cond    -1720.75949    431.52092  -3.988   0.0000692317202026 ***
## Year.Built        -63.11091     22.62099  -2.790             0.005324 **
## Mas.Vnr.Type1   -5886.58485   1717.80549  -3.427             0.000624 ***
## Mas.Vnr.Type2    6506.79972   2444.69722   2.662             0.007842 **
## Mas.Vnr.Type3  -13939.24778   6272.19738  -2.222             0.026373 *
## Mas.Vnr.Type4  -94719.29143  24296.78548  -3.898             0.000100 ***
## Mas.Vnr.Area       29.44673      4.75323   6.195   0.0000000007110044 ***
## Exter.Qual1     10655.92075   1928.49613   5.526   0.0000000373428172 ***
## Exter.Qual2    -11441.51189   6224.45677  -1.838             0.066194 .
## Exter.Qual3     42512.66336   4260.75699   9.978 < 0.0000000000000002 ***
## Bsmt.Qual1      20118.58456  18192.88232   1.106             0.268931
## Bsmt.Qual2       7721.64436   1702.57498   4.535   0.0000061087610588 ***
## Bsmt.Qual3      29823.56806   3117.66052   9.566 < 0.0000000000000002 ***
## Bsmt.Qual4      -1744.46705   3276.74440  -0.532             0.594526
## Bsmt.Exposure1   -578.56219   2016.60257  -0.287             0.774220
## Bsmt.Exposure2   6660.58923   1716.16989   3.881             0.000108 ***
## Bsmt.Exposure3  16594.87713   2214.39085   7.494   0.0000000000001014 ***
## BsmtFin.SF.1       48.63008      2.31571  21.000 < 0.0000000000000002 ***
## BsmtFin.SF.2       32.28098      3.82345   8.443 < 0.0000000000000002 ***
## Bsmt.Unf.SF        26.64659      2.25735  11.804 < 0.0000000000000002 ***
## Central.Air1     8857.29939   2715.30382   3.262             0.001126 **
## Full.Bath1      -1206.27223   1732.62613  -0.696             0.486382
## Full.Bath2      23911.30405   8634.21119   2.769             0.005671 **
## Full.Bath3      24728.97544   4633.62260   5.337   0.0000001057834091 ***
## Full.Bath4      35352.86252  18594.96498   1.901             0.057425 .
```

```
## Half.Bath1       5287.69767   1548.76616   3.414             0.000653 ***
## Half.Bath2     -32755.15748   7159.54359  -4.575   0.0000050662627051 ***
## Bedroom.AbvGr   -2209.76808    531.31714  -4.159   0.0000333696631671 ***
## Kitchen.Qual1  -10025.72569   1609.28590  -6.230   0.0000000005721305 ***
## Kitchen.Qual2  -10640.58941   4183.03064  -2.544             0.011045 *
## Kitchen.Qual3   20881.21513   2918.97278   7.154   0.0000000000011978 ***
## Functional      -7620.49349    896.31837  -8.502 < 0.0000000000000002 ***
## Garage.Area        27.18436      3.45040   7.879   0.0000000000000055 ***
## Screen.Porch       43.57766      9.61950   4.530   0.0000062576042032 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 23930 on 1917 degrees of freedom
## Multiple R-squared:  0.9127, Adjusted R-squared:  0.9104
## F-statistic: 408.8 on 49 and 1917 DF,  p-value: < 0.00000000000000022
```

```
cat("AIC of Model (Both Directions - BIC):", AIC(sm_both_bic), "\n")
```

```
## AIC of Model (Both Directions - BIC): 45299.54
```

```
sm_both_p = lm(price ~ area + Total.Bsmt.SF + Overall.Qual + BsmtFin.SF.1 + Garage.Area + Exter.Qual + Functio

# Summary of the model using p-value significance level
cat("Summary of Stepwise Model (Both Directions - p-value):\n")
```

```
## Summary of Stepwise Model (Both Directions - p-value):
```

```
summary(sm_both_p)
```

```
##
## Call:
## lm(formula = price ~ area + Total.Bsmt.SF + Overall.Qual + BsmtFin.SF.1 +
##     Garage.Area + Exter.Qual + Functional + Neighborhood + Bsmt.Qual +
##     Lot.Area + Kitchen.Qual + Central.Air + Screen.Porch + Mas.Vnr.Area +
##     Bsmt.Exposure + Overall.Cond + Half.Bath + Bedroom.AbvGr +
##     Full.Bath + Bldg.Type + MS.Zoning + Mas.Vnr.Type + Land.Contour +
##     Wood.Deck.SF + Year.Built + Fireplaces + Lot.Config + Kitchen.AbvGr +
##     Low.Qual.Fin.SF + Roof.Matl + Land.Slope, data = ames.train_data.df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -158085  -12683     616   12328  150097
##
## Coefficients:
##                    Estimate  Std. Error t value             Pr(>|t|)
## (Intercept)      19740.2882   4662.2727   4.234 0.000024048310794325 ***
## area                53.0898      2.2020  24.110 < 0.0000000000000002 ***
## Total.Bsmt.SF       26.6521      2.2184  12.014 < 0.0000000000000002 ***
## Overall.Qual      4539.2740    521.7498   8.700 < 0.0000000000000002 ***
## BsmtFin.SF.1        20.1556      1.5080  13.365 < 0.0000000000000002 ***
## Garage.Area         27.5986      3.4247   8.059 0.000000000000001350 ***
## Exter.Qual1      11317.8109   1903.5187   5.946 0.000000003267571519 ***
## Exter.Qual2     -12587.5048   6134.4441  -2.052             0.040312 *
## Exter.Qual3      42371.4097   4197.9234  10.093 < 0.0000000000000002 ***
## Functional       -7194.1660    888.1846  -8.100 0.000000000000000974 ***
## Neighborhood       911.9232    121.0582   7.533 0.000000000000076289 ***
## Bsmt.Qual1       20856.8845  17909.5548   1.165             0.244340
## Bsmt.Qual2        6635.4669   1699.4724   3.904 0.000097748833178485 ***
## Bsmt.Qual3       29184.5642   3082.5827   9.468 < 0.0000000000000002 ***
## Bsmt.Qual4       -1648.9484   3231.3620  -0.510             0.609904
## Lot.Area             0.4447      0.1047   4.245 0.000022876754911451 ***
## Kitchen.Qual1    -9718.1654   1592.8454  -6.101 0.000000001273124817 ***
```

```
## Kitchen.Qual2    -9280.9040   4126.4758  -2.249             0.024620 *
## Kitchen.Qual3    21941.8796   2885.0148   7.605 0.000000000000044371 ***
## Central.Air1      7638.2394   2679.0289   2.851             0.004404 **
## Screen.Porch        38.6976      9.6754   4.000 0.000065879363689502 ***
## Mas.Vnr.Area        29.1601      4.7081   6.194 0.000000000718947378 ***
## Bsmt.Exposure1    -929.5029   1974.9830  -0.471             0.637953
## Bsmt.Exposure2    5906.8658   1695.1997   3.484             0.000504 ***
## Bsmt.Exposure3   15320.8050   2221.4314   6.897 0.000000000007216317 ***
## Overall.Cond     -1698.2471    424.9587  -3.996 0.000066808351122440 ***
## Half.Bath1        4455.0573   1540.4870   2.892             0.003872 **
## Half.Bath2      -30264.3810   7177.8860  -4.216 0.000025997062351801 ***
## Bedroom.AbvGr    -1958.6425    525.5457  -3.727             0.000200 ***
## Full.Bath1        -261.2157   1736.0831  -0.150             0.880416
## Full.Bath2       16562.5484   9228.8884   1.795             0.072870 .
## Full.Bath3       24458.1879   4593.7524   5.324 0.000000113403595482 ***
## Full.Bath4       28887.6632  18315.7468   1.577             0.114915
## Bldg.Type1      -20307.7875   3359.5955  -6.045 0.000000001797267264 ***
## Bldg.Type2      -14487.1228   2380.5610  -6.086 0.000000001400387305 ***
## Bldg.Type3       -6234.1123   5637.7977  -1.106             0.268966
## Bldg.Type4      -13121.2259   4582.1915  -2.864             0.004236 **
## MS.Zoning1        4026.4914   1768.5179   2.277             0.022911 *
## MS.Zoning2      -20917.5423   6233.1014  -3.356             0.000807 ***
## MS.Zoning3       19890.8304   3158.9064   6.297 0.000000000376619959 ***
## MS.Zoning4       -2959.4610   5834.2753  -0.507             0.612035
## Mas.Vnr.Type1    -6469.4799   1697.8036  -3.810             0.000143 ***
## Mas.Vnr.Type2     6072.4264   2426.4984   2.503             0.012414 *
## Mas.Vnr.Type3   -14848.6057   6177.7039  -2.404             0.016331 *
## Mas.Vnr.Type4   -95815.2961  23896.2115  -4.010 0.000063163506935299 ***
## Land.Contour1    -3933.5991   3129.5553  -1.257             0.208937
## Land.Contour2    14125.2099   3012.0432   4.690 0.000002932817864150 ***
## Land.Contour3    -9621.3652   4814.1069  -1.999             0.045796 *
## Wood.Deck.SF        10.8037      4.6821   2.307             0.021138 *
## Year.Built         -63.7032     22.3416  -2.851             0.004401 **
## Fireplaces1       3059.3140   1318.1951   2.321             0.020401 *
## Fireplaces2      10341.8887   2422.8657   4.268 0.000020653378096453 ***
## Fireplaces3      -3949.7843   8132.3826  -0.486             0.627246
## Fireplaces4     -24780.3699  24430.5432  -1.014             0.310560
## Lot.Config1      -3784.2179   1464.2042  -2.584             0.009826 **
## Lot.Config2       7469.5430   2307.4222   3.237             0.001228 **
## Lot.Config3      -5507.8794   3146.2424  -1.751             0.080173 .
## Lot.Config4      -3387.7801   7251.9936  -0.467             0.640445
## Kitchen.AbvGr1  -18290.5677   5460.2361  -3.350             0.000825 ***
## Kitchen.AbvGr2   18648.6209  24103.4281   0.774             0.439209
## Kitchen.AbvGr3   -9786.6715  14778.7552  -0.662             0.507916
## Low.Qual.Fin.SF    -37.2179     14.0078  -2.657             0.007951 **
## Roof.Matl1       -4874.3648   8649.2265  -0.564             0.573120
## Roof.Matl2         304.8632   6412.8769   0.048             0.962088
## Roof.Matl3       52356.0947  13202.1065   3.966 0.000075883235260210 ***
## Roof.Matl4      -20558.0354  24073.3225  -0.854             0.393226
## Land.Slope1       -207.5147   3319.5732  -0.063             0.950161
## Land.Slope2     -27230.5136   9744.6864  -2.794             0.005252 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 23490 on 1899 degrees of freedom
## Multiple R-squared:  0.9166, Adjusted R-squared:  0.9137
## F-statistic: 311.5 on 67 and 1899 DF,  p-value: < 0.00000000000000022
```

```
cat("AIC of Model (Both Directions - p-value):", AIC(sm_both_p), "\n")
```

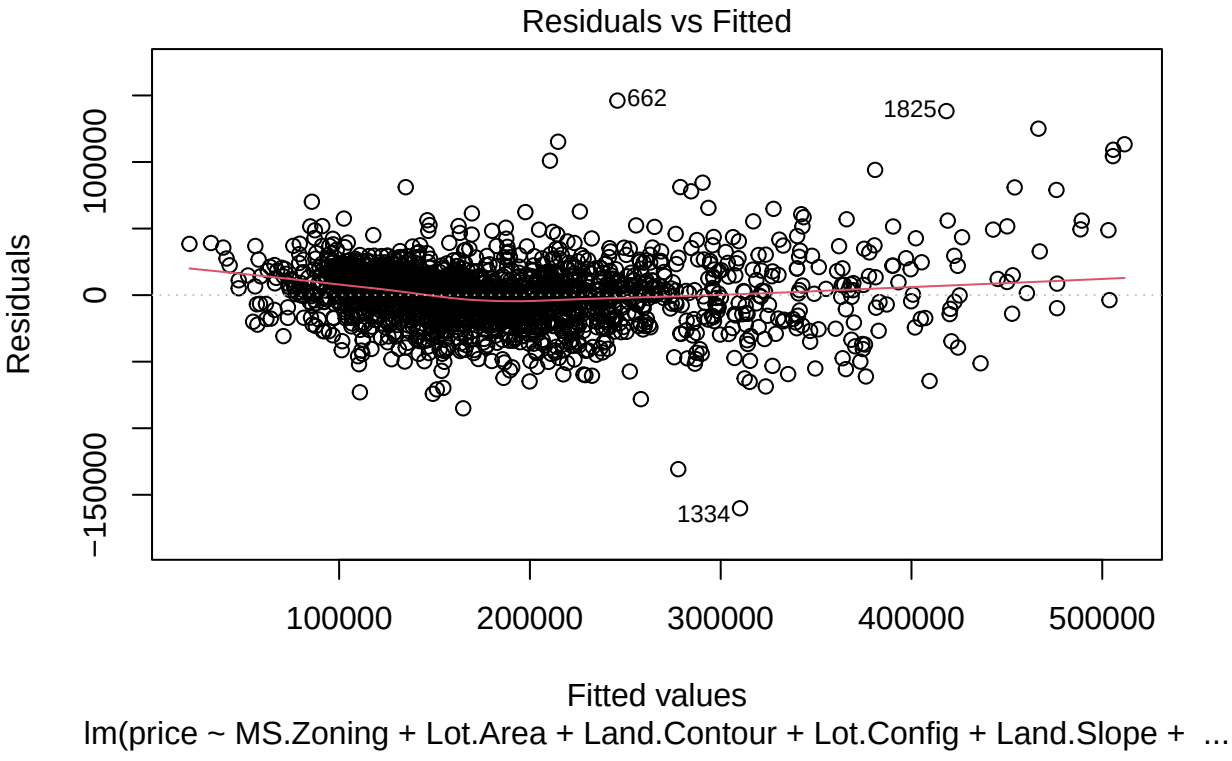
```
## AIC of Model (Both Directions - p-value): 45244.57
```

```
#RMSE,Rsquared,MAE show that sm_both_aic model performs best on train data
# Predictions for sm_both_aic
y_hat_both_aic <- predict(sm_both_aic, newdata = ames.train_data.df)
Model.pred_both_aic <- data.frame(obs = ames.train_data.df$price, pred = y_hat_both_aic)
both_aic <- defaultSummary(Model.pred_both_aic)

# Predictions for sm_both_bic
y_hat_both_bic <- predict(sm_both_bic, newdata = ames.train_data.df)
Model.pred_both_bic <- data.frame(obs = ames.train_data.df$price, pred = y_hat_both_bic)
both_bic <- defaultSummary(Model.pred_both_bic)

# Predictions for sm_both_p
y_hat_both_p <- predict(sm_both_p, newdata = ames.train_data.df)
Model.pred_both_p <- data.frame(obs = ames.train_data.df$price, pred = y_hat_both_p)
both_p <- defaultSummary(Model.pred_both_p)

#default summary for model performance
out<-rbind(both_aic,both_bic,both_p)
dimnames(out)[[1]]<-c("both_aic","both_bic","both_p")
out
```

```
##              RMSE  Rsquared      MAE
## both_aic 22995.41 0.9172402 16406.90
## both_bic 23623.91 0.9126545 16914.71
## both_p   23083.90 0.9166021 16467.76
```

7. Make sure that model assumption(s) are checked for the final model. Apply remedy measures (transformation, etc.) that helps satisfy the linear model assumptions.

```
#Constancy of the Error Variance
#Residuals vs Fitted and breusch pagan test show that Error Variance is
#not constant
plot(sm_both_aic, which = 1)
```

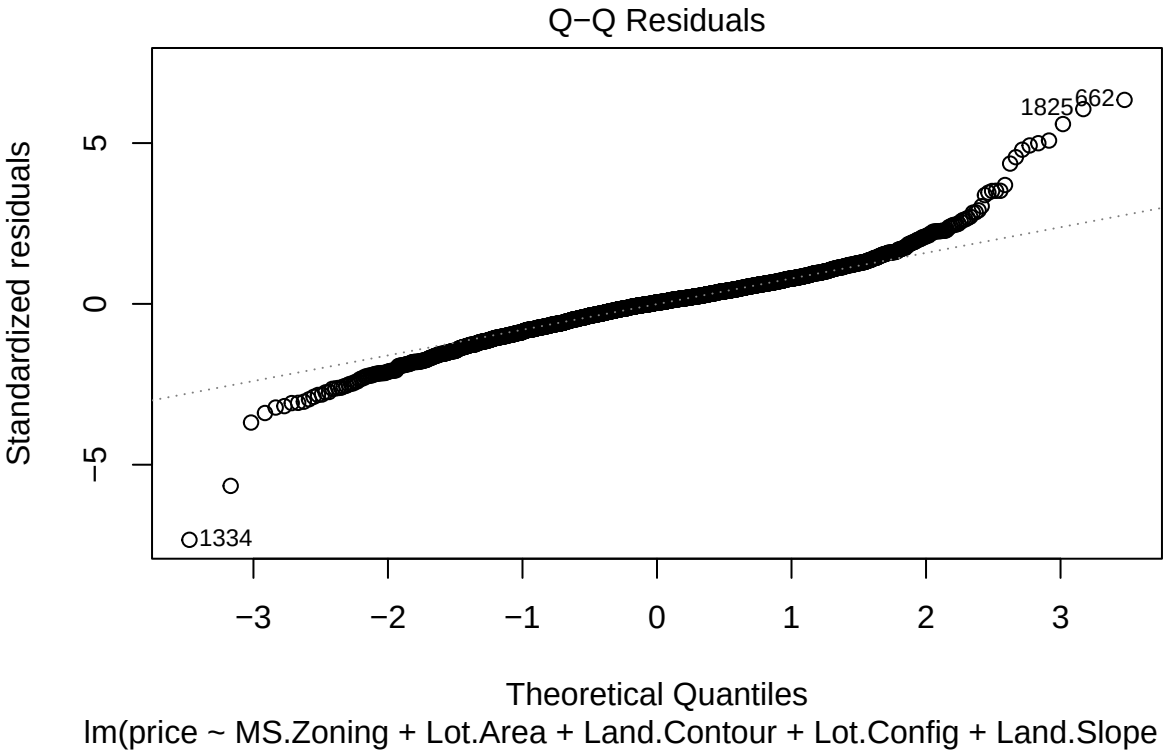



```
ols_test_breusch_pagan(sm_both_aic)
```

```
##
##  Breusch Pagan Test for Heteroskedasticity
##  -----------------------------------------
##  Ho: the variance is constant
##  Ha: the variance is not constant
##
##                Data
##  ---------------------------------
##  Response : price
##  Variables: fitted values of price
##
##                                                                  Test Summary
##  ---------------------------------------------------------------------------------------------------------------------
##  DF            =    1
##  Chi2          =    474.7071
##  Prob > Chi2   =    0.0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
```

```
#QQ Plot for Normality
#Normality plot and Shapiro test show that residuals
#do not fulfill normality assumption
plot(sm_both_aic, which = 2)
```

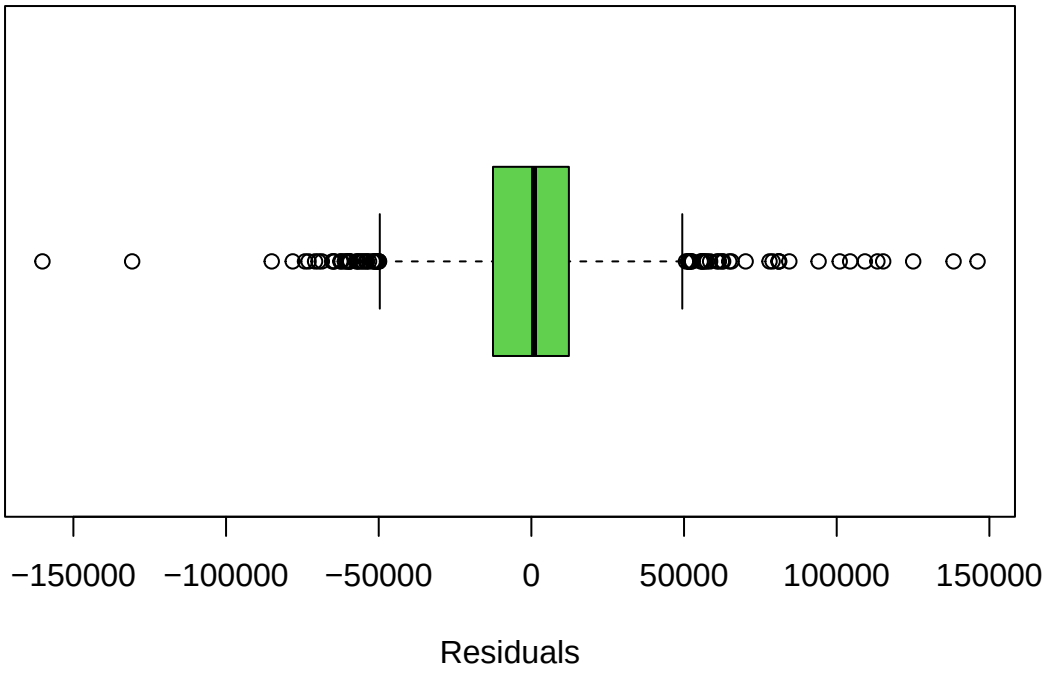
```
## Warning: not plotting observations with leverage one:
##   462, 524, 891, 1645
```



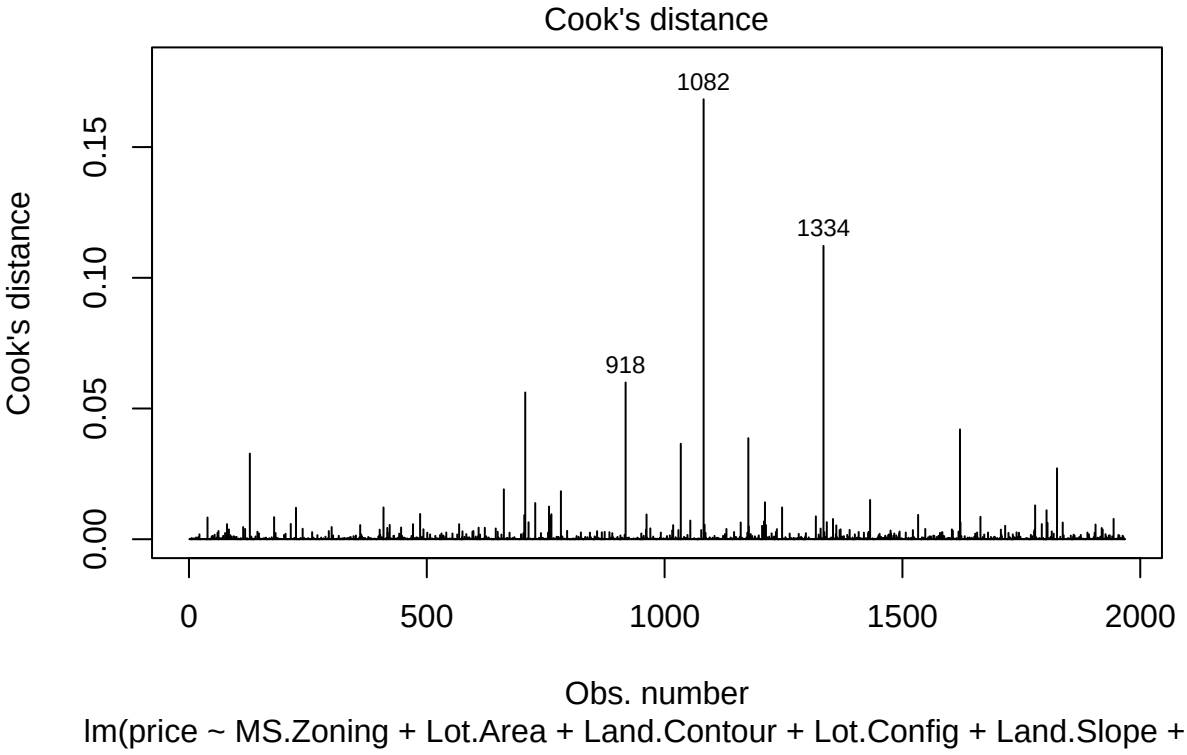

```
#Shapiro Test
shapiro.test(sm_both_aic$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sm_both_aic$residuals
## W = 0.94814, p-value < 0.00000000000000022
```

```
#Presence of outliers.
#Boxplot of residuals for potential outliers
boxplot(sm_both_aic$residuals,horizontal=TRUE,staplewex=0.5,col=3,xlab="Residuals")
```

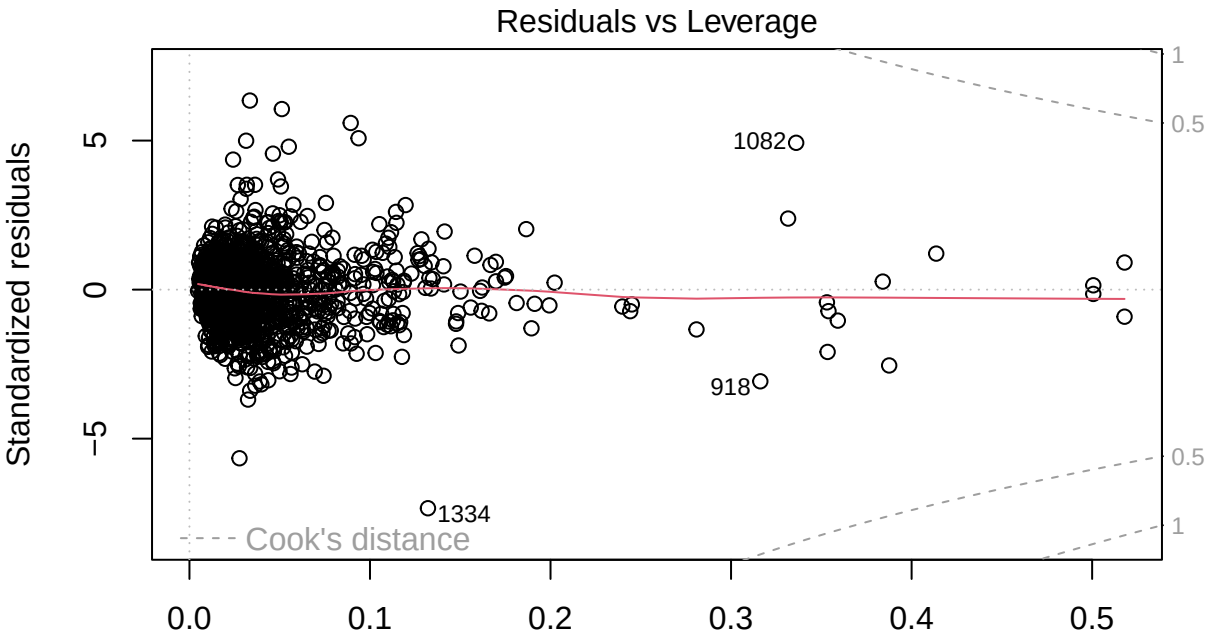



```
#Cooks distance plot
plot(sm_both_aic,which=4)
```

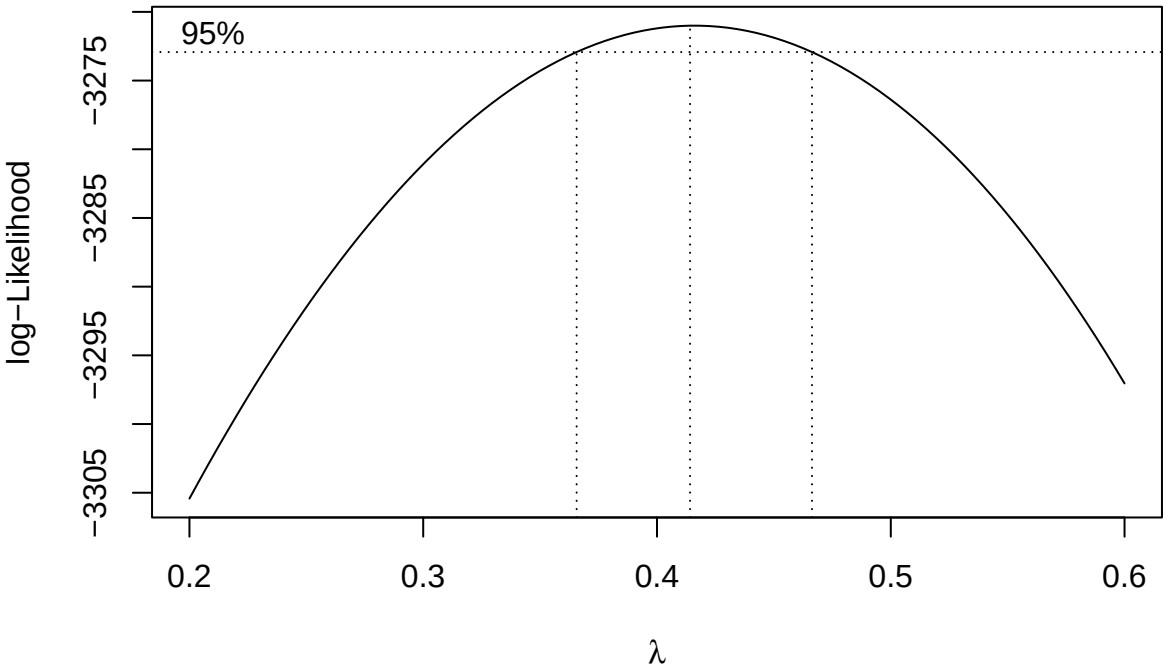



```
#Leverage vs standardized residuals
plot(sm_both_aic, which = 5)
```

```
## Warning: not plotting observations with leverage one:
##   462, 524, 891, 1645
```




```
#Transforming sm_both_aic
# Perform Box-Cox transformation
boxcox_model = boxcox(sm_both_aic,lambda=seq(0.2,0.6, by=.1))
```




```
#lamda calculation
lambda = boxcox_model$x[which.max(boxcox_model$y)]
lambda
```

```
## [1] 0.4141414
```

```
summary(sm_both_aic)
```

```
##
## Call:
## lm(formula = price ~ MS.Zoning + Lot.Area + Land.Contour + Lot.Config +
##     Land.Slope + Neighborhood + Condition.1 + Bldg.Type + Overall.Qual +
```

```
##     Overall.Cond + Year.Built + Year.Remod.Add + Roof.Matl +
##     Mas.Vnr.Type + Mas.Vnr.Area + Exter.Qual + Bsmt.Qual + Bsmt.Exposure +
##     BsmtFin.SF.1 + BsmtFin.Type.2 + BsmtFin.SF.2 + Bsmt.Unf.SF +
##     Central.Air + X1st.Flr.SF + X2nd.Flr.SF + Full.Bath + Half.Bath +
##     Bedroom.AbvGr + Kitchen.AbvGr + Kitchen.Qual + Functional +
##     Fireplaces + Garage.Area + Wood.Deck.SF + Screen.Porch +
##     Sale.Condition, data = ames.train_data.df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -160135  -12573     871   12283  146136
##
## Coefficients:
##                    Estimate   Std. Error t value             Pr(>|t|)
## (Intercept)      16262.2225    5138.3066   3.165             0.001576 **
## MS.Zoning1        3348.5073    1794.8868   1.866             0.062254 .
## MS.Zoning2      -22162.8737    6225.5354  -3.560             0.000380 ***
## MS.Zoning3       19340.6244    3161.9869   6.117 0.000000001158272028 ***
## MS.Zoning4       -2982.3376    5824.3504  -0.512             0.608678
## Lot.Area             0.4399       0.1047   4.200 0.000027874117901285 ***
## Land.Contour1    -4094.8373    3116.1184  -1.314             0.188977
## Land.Contour2    13994.4744    3019.5879   4.635 0.000003820875067584 ***
## Land.Contour3    -9974.0231    4831.6194  -2.064             0.039123 *
## Lot.Config1      -3701.1657    1469.5152  -2.519             0.011863 *
## Lot.Config2       6848.2737    2314.3966   2.959             0.003125 **
## Lot.Config3      -5006.2156    3149.3215  -1.590             0.112088
## Lot.Config4      -2976.1831    7241.0132  -0.411             0.681106
## Land.Slope1        347.9820    3337.6107   0.104             0.916973
## Land.Slope2     -27006.8477    9753.3697  -2.769             0.005678 **
## Neighborhood       945.9472     121.2556   7.801 0.000000000000010048 ***
## Condition.1       1088.2544     578.3533   1.882             0.060038 .
## Bldg.Type1      -20501.7536    3372.5142  -6.079 0.000000001458109523 ***
## Bldg.Type2      -14856.2101    2393.2215  -6.208 0.000000000659230076 ***
## Bldg.Type3       -7002.3190    5635.1464  -1.243             0.214163
## Bldg.Type4      -13526.4867    4577.4597  -2.955             0.003165 **
## Overall.Qual      4594.8305     522.3089   8.797 < 0.0000000000000002 ***
## Overall.Cond     -1669.5926     424.3817  -3.934 0.000086493364536119 ***
## Year.Built         -70.7364      23.4999  -3.010             0.002646 **
## Year.Remod.Add      57.8324      37.0380   1.561             0.118589
## Roof.Matl1       -7783.2126    8692.3745  -0.895             0.370683
## Roof.Matl2       -3111.9392    6493.0753  -0.479             0.631802
## Roof.Matl3       55636.0348   13033.9958   4.269 0.000020648710567730 ***
## Roof.Matl4      -14904.4835   24076.8646  -0.619             0.535966
## Mas.Vnr.Type1    -6717.6324    1701.8462  -3.947 0.000081937875556025 ***
## Mas.Vnr.Type2     5184.4695    2438.0288   2.127             0.033591 *
## Mas.Vnr.Type3   -16739.9770    6187.5672  -2.705             0.006883 **
## Mas.Vnr.Type4  -100248.0830   23869.0365  -4.200 0.000027939824523420 ***
## Mas.Vnr.Area        29.1664       4.6980   6.208 0.000000000656430099 ***
## Exter.Qual1      11640.4044    1907.8798   6.101 0.000000001273050716 ***
## Exter.Qual2     -13123.7930    6131.6603  -2.140             0.032455 *
## Exter.Qual3      41968.8672    4191.1033  10.014 < 0.0000000000000002 ***
## Bsmt.Qual1       22437.2711   17909.8314   1.253             0.210437
## Bsmt.Qual2        6379.2995    1700.2108   3.752             0.000181 ***
## Bsmt.Qual3       28570.4330    3100.7270   9.214 < 0.0000000000000002 ***
## Bsmt.Qual4       -2038.4352    3235.8034  -0.630             0.528795
## Bsmt.Exposure1   -1168.9200    1982.2361  -0.590             0.555464
## Bsmt.Exposure2    5533.6349    1697.4069   3.260             0.001134 **
## Bsmt.Exposure3   14799.5761    2229.3133   6.639 0.000000000041222152 ***
## BsmtFin.SF.1        43.0822       3.3563  12.836 < 0.0000000000000002 ***
## BsmtFin.Type.2   -1192.6584     773.6142  -1.542             0.123321
```

```
## BsmtFin.SF.2        32.1546       5.6839   5.657 0.000000017740266811 ***
## Bsmt.Unf.SF         22.2379       3.2887   6.762 0.000000000018086985 ***
## Central.Air1      7027.3390    2706.2015   2.597              0.009484 **
## X1st.Flr.SF         57.0136       3.5254  16.172 < 0.0000000000000002 ***
## X2nd.Flr.SF         52.4948       2.3711  22.139 < 0.0000000000000002 ***
## Full.Bath1        -266.9180    1741.5490  -0.153              0.878206
## Full.Bath2       14555.4233    9231.1115   1.577              0.115013
## Full.Bath3       24725.6976    4609.4412   5.364 0.000000091304601092 ***
## Full.Bath4       31564.1394   18299.7096   1.725              0.084719 .
## Half.Bath1        4641.1220    1620.4929   2.864              0.004229 **
## Half.Bath2      -30179.2237    7197.8505  -4.193 0.000028822169669591 ***
## Bedroom.AbvGr    -2040.2978     525.2243  -3.885              0.000106 ***
## Kitchen.AbvGr1  -17498.3911    5454.3244  -3.208              0.001358 **
## Kitchen.AbvGr2   15636.2844   24093.5467   0.649              0.516429
## Kitchen.AbvGr3   -9594.1783   14828.0906  -0.647              0.517693
## Kitchen.Qual1    -9409.1693    1593.4424  -5.905 0.000000004171426646 ***
## Kitchen.Qual2    -8389.5272    4135.2693  -2.029              0.042621 *
## Kitchen.Qual3    21728.2926    2880.1483   7.544 0.000000000000070251 ***
## Functional       -7344.0220     891.2560  -8.240 0.000000000000000317 ***
## Fireplaces1       3371.9028    1326.5272   2.542              0.011104 *
## Fireplaces2      10609.2918    2430.5616   4.365 0.000013400393881788 ***
## Fireplaces3      -4784.2646    8141.4666  -0.588              0.556843
## Fireplaces4     -27454.5918   24437.2948  -1.123              0.261380
## Garage.Area         27.1080       3.4333   7.896 0.000000000000004852 ***
## Wood.Deck.SF        10.8905       4.7081   2.313              0.020821 *
## Screen.Porch        38.1948       9.6775   3.947 0.000082102610642733 ***
## Sale.Condition    1180.6302     578.7996   2.040              0.041510 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 23430 on 1894 degrees of freedom
## Multiple R-squared:  0.9172, Adjusted R-squared:  0.9141
## F-statistic: 291.5 on 72 and 1894 DF,  p-value: < 0.00000000000000022
```

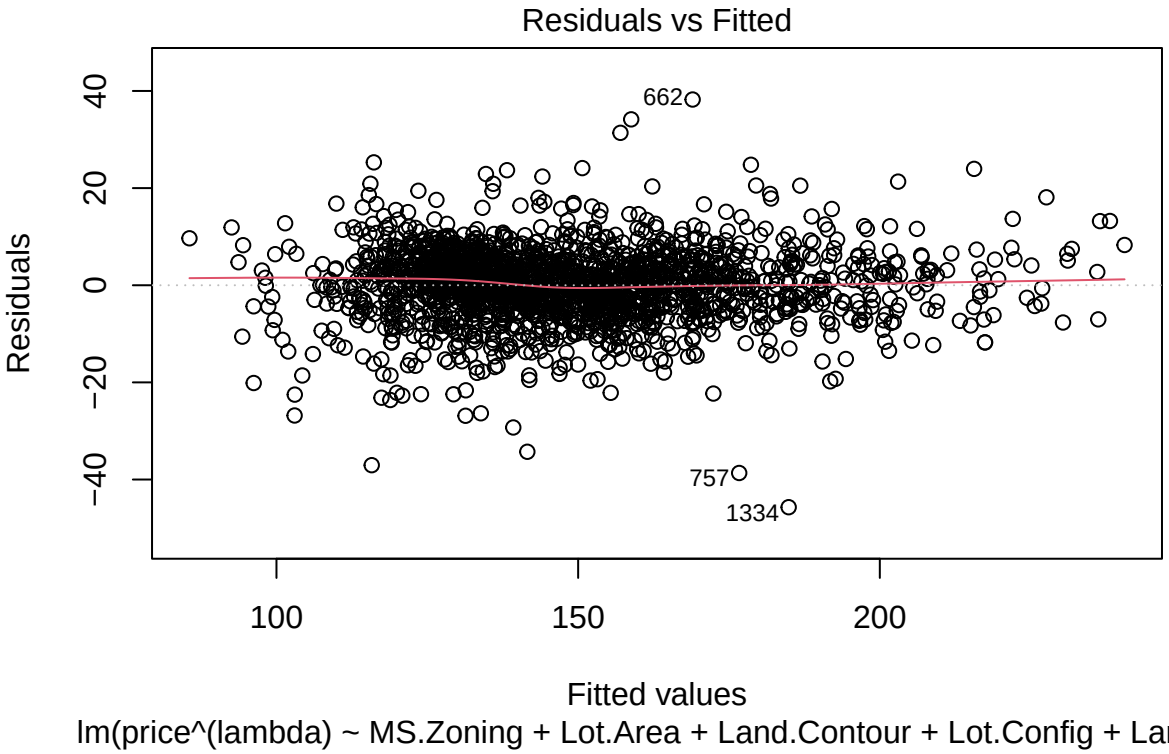
```
# Transformed model
transformed_model = lm(price^(lambda) ~ MS.Zoning + Lot.Area + Land.Contour + Lot.Config +
    Land.Slope + Neighborhood + Condition.1 + Bldg.Type + Overall.Qual +
    Overall.Cond + Year.Built + Year.Remod.Add + Roof.Matl +
    Mas.Vnr.Type + Mas.Vnr.Area + Exter.Qual + Bsmt.Qual + Bsmt.Exposure +
    BsmtFin.SF.1 + BsmtFin.Type.2 + BsmtFin.SF.2 + Bsmt.Unf.SF +
    Central.Air + X1st.Flr.SF + X2nd.Flr.SF + Full.Bath + Half.Bath +
    Bedroom.AbvGr + Kitchen.AbvGr + Kitchen.Qual + Functional +
    Fireplaces + Garage.Area + Wood.Deck.SF + Screen.Porch +
    Sale.Condition, data = ames.train_data.df)
summary(transformed_model)
```

```
##
## Call:
## lm(formula = price^(lambda) ~ MS.Zoning + Lot.Area + Land.Contour +
##     Lot.Config + Land.Slope + Neighborhood + Condition.1 + Bldg.Type +
##     Overall.Qual + Overall.Cond + Year.Built + Year.Remod.Add +
##     Roof.Matl + Mas.Vnr.Type + Mas.Vnr.Area + Exter.Qual + Bsmt.Qual +
##     Bsmt.Exposure + BsmtFin.SF.1 + BsmtFin.Type.2 + BsmtFin.SF.2 +
##     Bsmt.Unf.SF + Central.Air + X1st.Flr.SF + X2nd.Flr.SF + Full.Bath +
##     Half.Bath + Bedroom.AbvGr + Kitchen.AbvGr + Kitchen.Qual +
##     Functional + Fireplaces + Garage.Area + Wood.Deck.SF + Screen.Porch +
##     Sale.Condition, data = ames.train_data.df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -45.696  -3.861   0.338   4.163  38.235
```

```
##
## Coefficients:
##                    Estimate   Std. Error t value             Pr(>|t|)
## (Intercept)     93.05229074   1.61065587  57.773 < 0.0000000000000002 ***
## MS.Zoning1       2.41928057   0.56262601   4.300  0.00001794995946226 ***
## MS.Zoning2     -12.21062769   1.95145909  -6.257  0.00000000048346838 ***
## MS.Zoning3       8.23179741   0.99115781   8.305 < 0.0000000000000002 ***
## MS.Zoning4      -0.67703427   1.82570345  -0.371             0.710802
## Lot.Area         0.00010610   0.00003283   3.232             0.001251 **
## Land.Contour1   -0.58802518   0.97677985  -0.602             0.547244
## Land.Contour2    3.29637221   0.94652136   3.483             0.000508 ***
## Land.Contour3   -2.01931643   1.51452157  -1.333             0.182593
## Lot.Config1     -0.69506955   0.46063489  -1.509             0.131481
## Lot.Config2      1.46163485   0.72547179   2.015             0.044073 *
## Lot.Config3     -1.36513677   0.98718773  -1.383             0.166873
## Lot.Config4     -0.21754768   2.26977120  -0.096             0.923653
## Land.Slope1     -0.16024870   1.04620894  -0.153             0.878280
## Land.Slope2     -4.76680726   3.05729560  -1.559             0.119126
## Neighborhood     0.28212194   0.03800884   7.423  0.00000000000017266 ***
## Condition.1      0.41652213   0.18129088   2.298             0.021697 *
## Bldg.Type1      -7.71770777   1.05714980  -7.300  0.00000000000042004 ***
## Bldg.Type2      -3.80540300   0.75018027  -5.073  0.00000043065854514 ***
## Bldg.Type3      -2.04441248   1.76639549  -1.157             0.247258
## Bldg.Type4      -3.50649192   1.43485254  -2.444             0.014624 *
## Overall.Qual     0.99528721   0.16372319   6.079  0.00000000145798562 ***
## Overall.Cond    -0.65409899   0.13302687  -4.917  0.00000095440570791 ***
## Year.Built      -0.02413383   0.00736628  -3.276             0.001071 **
## Year.Remod.Add   0.03110254   0.01160996   2.679             0.007449 **
## Roof.Matl1      -3.29635002   2.72471559  -1.210             0.226508
## Roof.Matl2       0.31897334   2.03532228   0.157             0.875483
## Roof.Matl3      12.43662776   4.08564211   3.044             0.002367 **
## Roof.Matl4      -2.89361694   7.54714470  -0.383             0.701462
## Mas.Vnr.Type1   -0.51581497   0.53346147  -0.967             0.333707
## Mas.Vnr.Type2    2.80552689   0.76422559   3.671             0.000248 ***
## Mas.Vnr.Type3   -3.46160297   1.93955756  -1.785             0.074464 .
## Mas.Vnr.Type4  -34.20365749   7.48199881  -4.571  0.00000515636310121 ***
## Mas.Vnr.Area     0.00273857   0.00147263   1.860             0.063089 .
## Exter.Qual1      3.64853545   0.59804485   6.101  0.00000000127659415 ***
## Exter.Qual2     -6.40009170   1.92203297  -3.330             0.000886 ***
## Exter.Qual3     10.10810177   1.31374511   7.694  0.00000000000002276 ***
## Bsmt.Qual1       8.44567845   5.61402372   1.504             0.132648
## Bsmt.Qual2       2.75819407   0.53294883   5.175  0.00000025159903497 ***
## Bsmt.Qual3       7.64497758   0.97195527   7.866  0.00000000000000612 ***
## Bsmt.Qual4      -1.95561191   1.01429636  -1.928             0.053999 .
## Bsmt.Exposure1  -0.27805681   0.62135261  -0.448             0.654564
## Bsmt.Exposure2   1.61276854   0.53206992   3.031             0.002469 **
## Bsmt.Exposure3   4.18340844   0.69880152   5.987  0.00000000255767111 ***
## BsmtFin.SF.1     0.01157444   0.00105208  11.001 < 0.0000000000000002 ***
## BsmtFin.Type.2  -0.20922155   0.24249746  -0.863             0.388369
## BsmtFin.SF.2     0.00836222   0.00178168   4.693  0.00000287906301659 ***
## Bsmt.Unf.SF      0.00555173   0.00103089   5.385  0.00000008129394170 ***
## Central.Air1     4.78951243   0.84828713   5.646  0.00000001889631673 ***
## X1st.Flr.SF      0.01929997   0.00110507  17.465 < 0.0000000000000002 ***
## X2nd.Flr.SF      0.01669872   0.00074325  22.467 < 0.0000000000000002 ***
## Full.Bath1       1.50646687   0.54590673   2.760             0.005843 **
## Full.Bath2       4.81954459   2.89358832   1.666             0.095960 .
## Full.Bath3       6.29683861   1.44487748   4.358  0.00001382617511018 ***
## Full.Bath4      11.09089820   5.73623510   1.933             0.053326 .
## Half.Bath1       1.46575269   0.50796043   2.886             0.003951 **
## Half.Bath2      -8.72290105   2.25624143  -3.866             0.000114 ***
```

```
## Bedroom.AbvGr   -0.41173202   0.16463705  -2.501             0.012474 *
## Kitchen.AbvGr1  -5.67393522   1.70971494  -3.319             0.000922 ***
## Kitchen.AbvGr2   3.53227527   7.55237388   0.468             0.640050
## Kitchen.AbvGr3  -1.71878726   4.64801989  -0.370             0.711581
## Kitchen.Qual1   -3.93339882   0.49948115  -7.875  0.00000000000000569 ***
## Kitchen.Qual2   -5.17590313   1.29624336  -3.993  0.00006773300732811 ***
## Kitchen.Qual3    5.09150549   0.90281256   5.640  0.00000001961068436 ***
## Functional      -2.94738891   0.27937350 -10.550 < 0.0000000000000002 ***
## Fireplaces1      2.26505574   0.41581379   5.447  0.00000005782090125 ***
## Fireplaces2      4.10477993   0.76188493   5.388  0.00000008028653818 ***
## Fireplaces3      0.28770979   2.55202773   0.113             0.910250
## Fireplaces4     -7.94951900   7.66012531  -1.038             0.299505
## Garage.Area      0.01033064   0.00107621   9.599 < 0.0000000000000002 ***
## Wood.Deck.SF     0.00375560   0.00147580   2.545             0.011013 *
## Screen.Porch     0.01279155   0.00303350   4.217  0.00002595233678083 ***
## Sale.Condition  -0.03168366   0.18143076  -0.175             0.861387
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.346 on 1894 degrees of freedom
## Multiple R-squared:  0.9175, Adjusted R-squared:  0.9143
## F-statistic: 292.4 on 72 and 1894 DF,  p-value: < 0.00000000000000022
```

```
#Constancy of the Error Variance
#Residuals vs Fitted and breusch pagan test show that Error Variance is constant
plot(transformed_model, which=1)
```



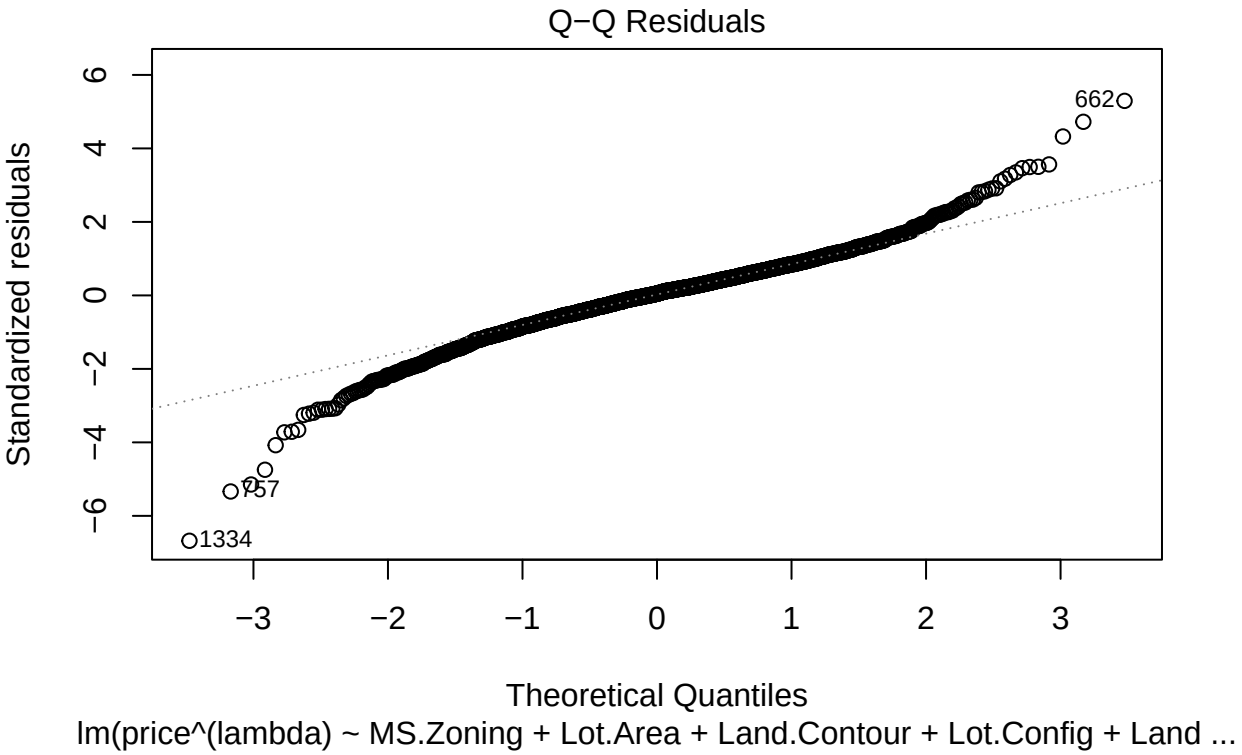

```
ols_test_breusch_pagan(transformed_model)
```

```
##
##  Breusch Pagan Test for Heteroskedasticity
##  -----------------------------------------
##  Ho: the variance is constant
##  Ha: the variance is not constant
##
##                  Data
##  -----------------------------------------
```

```
##  Response : price^(lambda)
##  Variables: fitted values of price^(lambda)
##
##          Test Summary
##  -----------------------------
##  DF            =    1
##  Chi2          =    0.002167362
##  Prob > Chi2   =    0.9628679
```

```
#QQ Plot for Normality
#Normality plot and Shapiro test show that residuals
#do not fulfill normality assumption
plot(transformed_model, which = 2)
```

```
## Warning: not plotting observations with leverage one:
##   462, 524, 891, 1645
```




```
#Shapiro Test
shapiro.test(transformed_model$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  transformed_model$residuals
## W = 0.96628, p-value < 0.00000000000000022
```

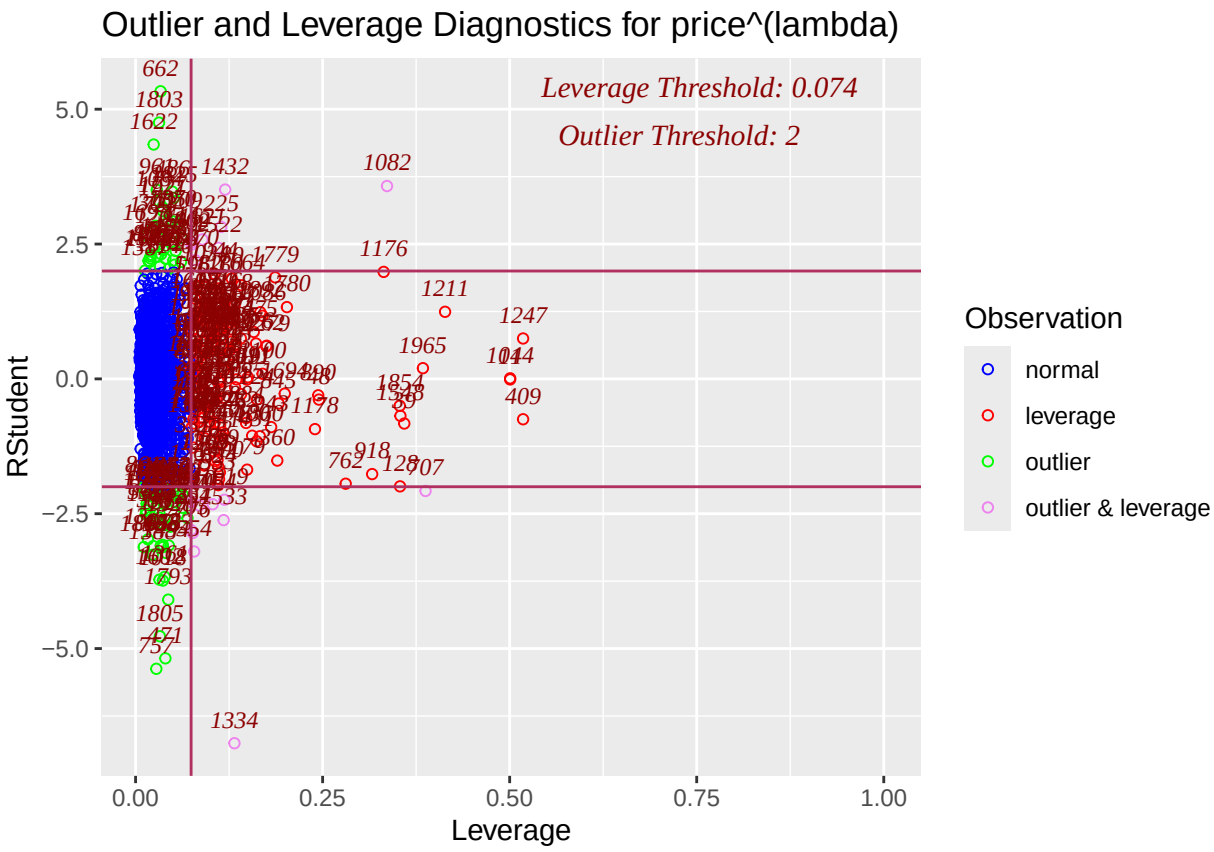
```
# copy transformed model and data to temp var - this will be changed recursively
temp_model <- transformed_model
temp_data <- ames.train_data.df

# function to get transformed model from filtered dataset
getRevisedModel <- function(filteredData) {
  revisedModel <- lm(price^(lambda) ~ MS.Zoning + Lot.Area + Land.Contour + Lot.Config +
    Land.Slope + Neighborhood + Condition.1 + Bldg.Type + Overall.Qual +
    Overall.Cond + Year.Built + Year.Remod.Add + Roof.Matl +
    Mas.Vnr.Type + Mas.Vnr.Area + Exter.Qual + Bsmt.Qual + Bsmt.Exposure +
    BsmtFin.SF.1 + BsmtFin.Type.2 + BsmtFin.SF.2 + Bsmt.Unf.SF +
    Central.Air + X1st.Flr.SF + X2nd.Flr.SF + Full.Bath + Half.Bath +
```

```
    Bedroom.AbvGr + Kitchen.AbvGr + Kitchen.Qual + Functional +
    Fireplaces + Garage.Area + Wood.Deck.SF + Screen.Porch +
    Sale.Condition, data = filteredData)
  return (revisedModel)
}

# function to recursively remove influential outliers
removeInfluentials <- function(model,temp_data){
  print(paste("Number of Observations ", nrow(temp_data), sep=""))
  a <- ols_plot_resid_lev(model)
  influentials <- a$data[a$data$fct_color=="outlier & leverage",]
  influentials <- drop_na(influentials)
  numInfluentials <- nrow(influentials)
  print(paste("Number of influentials -", numInfluentials,sep = " "))
  if(numInfluentials >0){
    temp_data <- temp_data[ -c(influentials$obs), ]
    temp_model <- getRevisedModel(temp_data)
    removeInfluentials(temp_model,temp_data)
  }else{
    return (temp_data)
  }
}

# use train data without influentials
ames.train_data.df <- temp_data
transformed_model <- getRevisedModel(ames.train_data.df)


### The below is just an exercise to see how the plots look like when we get rid of all outliers
# although the model fits well after outlier removal, removing about 400 observations seemed like a case of ov
# the function below is not changing any data or model that is used in teh subsequent sections

# function to recursively remove outliers
removeOutliers <- function(model,temp_data){
  print(paste("Number of Observations ", nrow(temp_data), sep=""))
  a <- ols_plot_cooksd_bar(model)
  outliers <- a$data[a$data$fct_color=="outlier",]
  outliers <- drop_na(outliers)
  numOutliers <- nrow(outliers)
  print(paste("Number of outliers -", numOutliers,sep = " "))
  if(numOutliers >0){
    temp_data <- temp_data[ -c(outliers$obs), ]
    temp_model <- getRevisedModel(temp_data)
    removeOutliers(temp_model,temp_data)
  }else{
    return (temp_data)
  }
}

print(paste("Number of rows before removing influentials -",nrow(temp_data)))
```

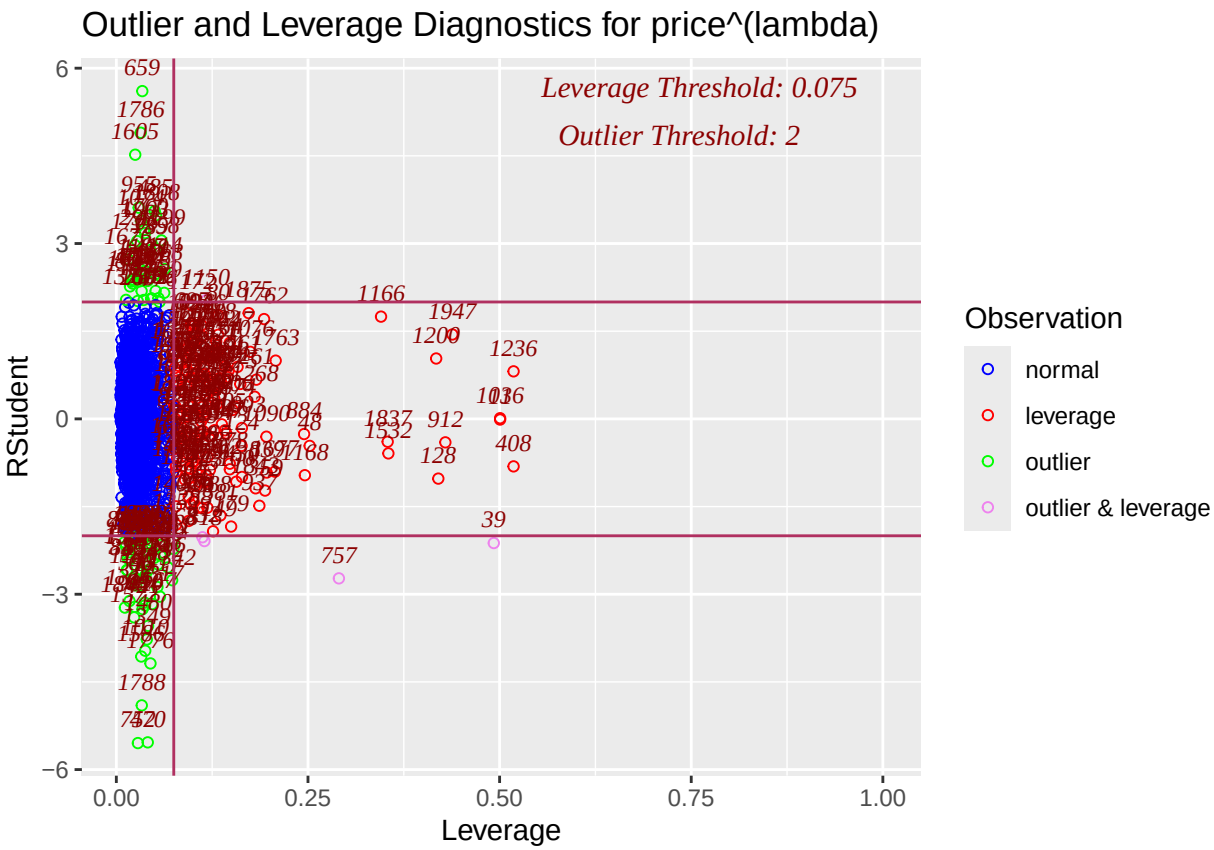
```
## [1] "Number of rows before removing influentials - 1967"
```

```
temp_data <- removeInfluentials(temp_model,temp_data)
```

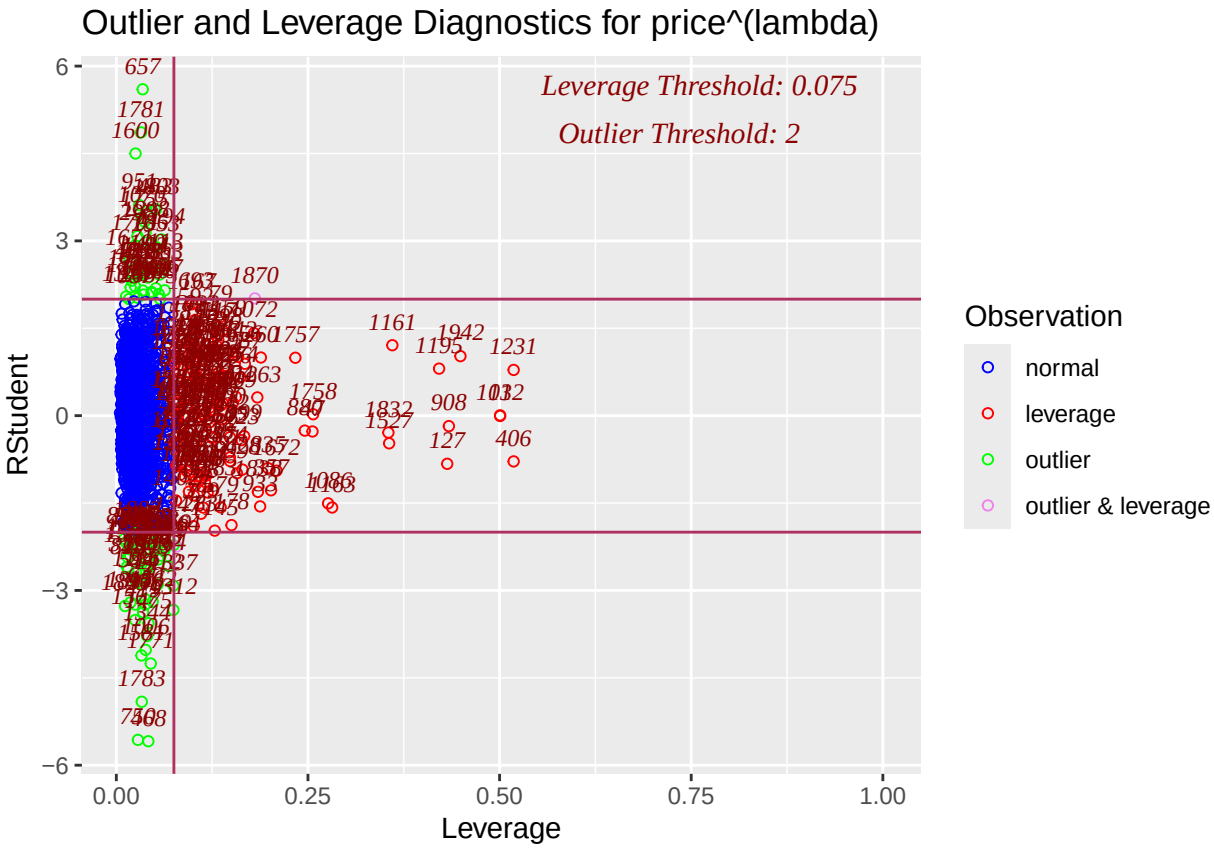
```
## [1] "Number of Observations 1967"
```



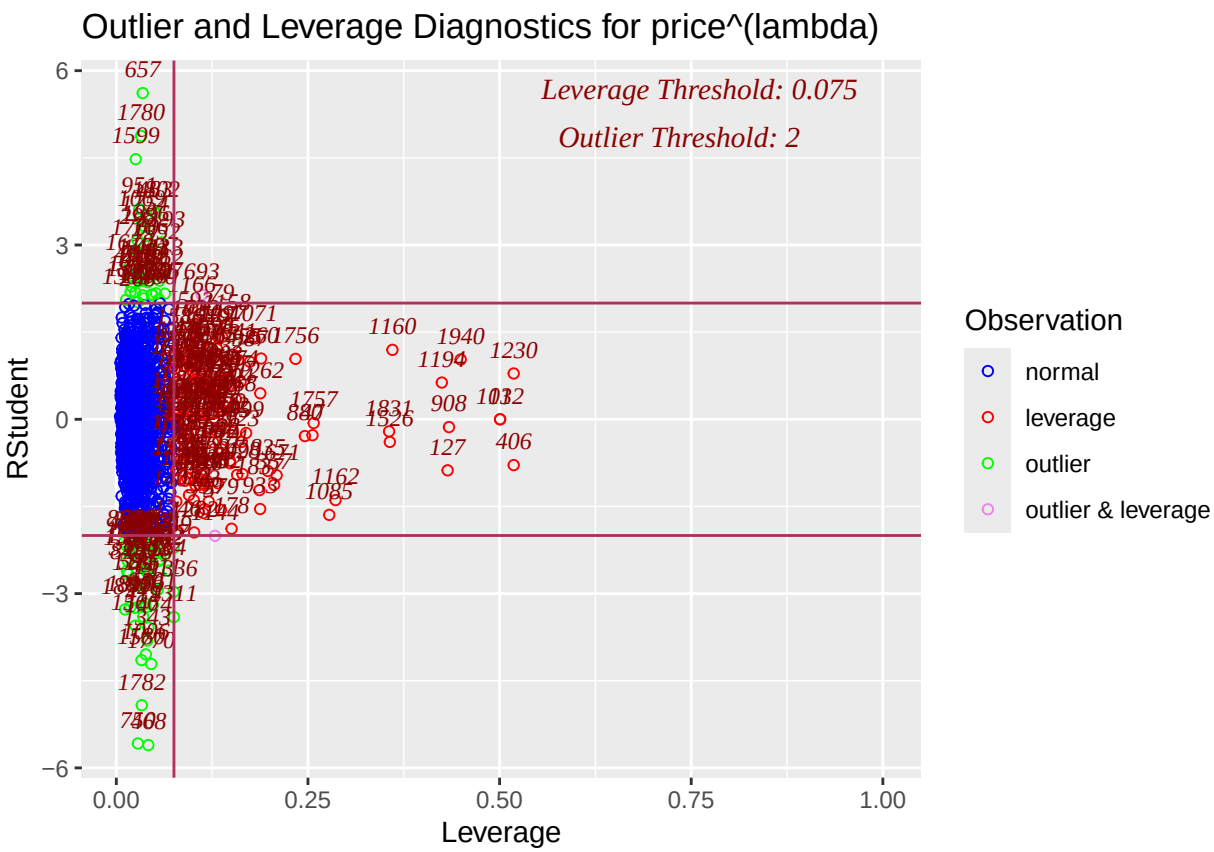

```
## [1] "Number of influentials - 18"
## [1] "Number of Observations 1949"
```



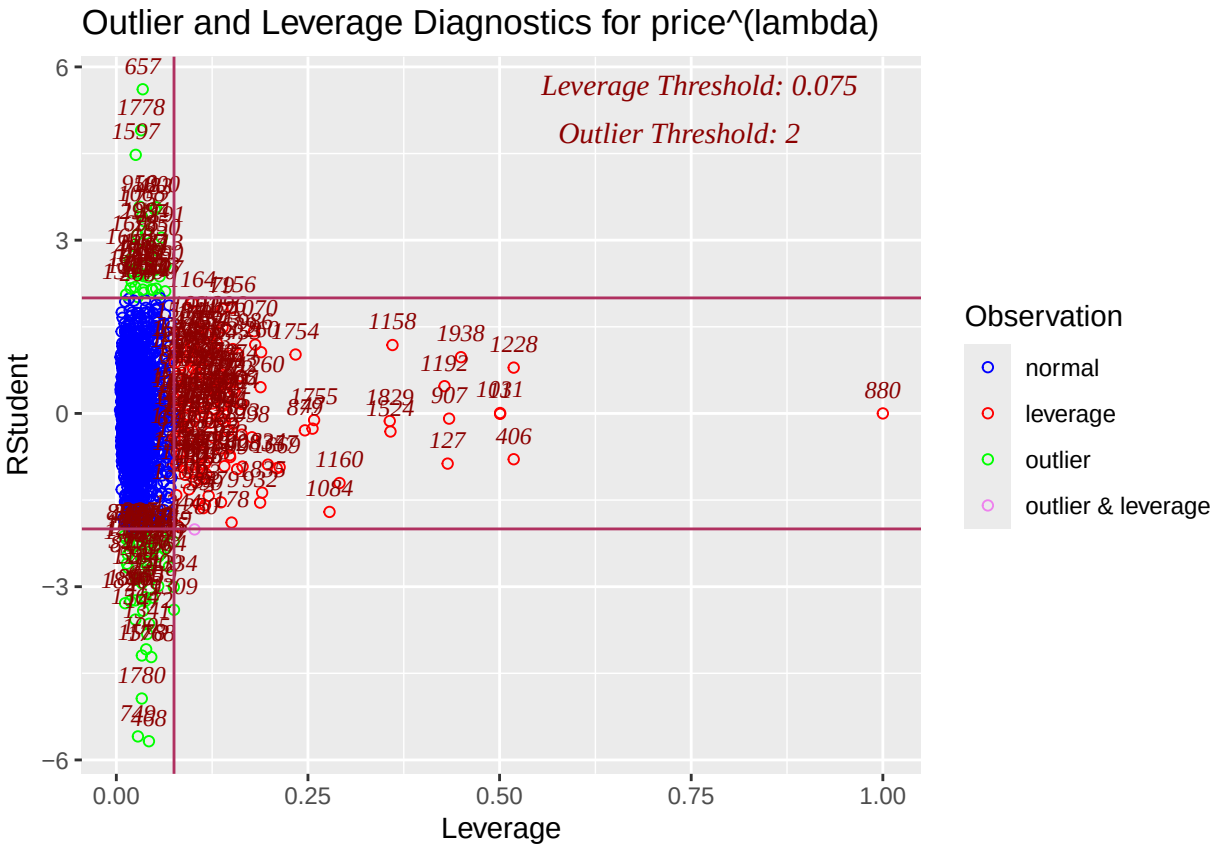

```
## [1] "Number of influentials - 5"
## [1] "Number of Observations 1944"
```



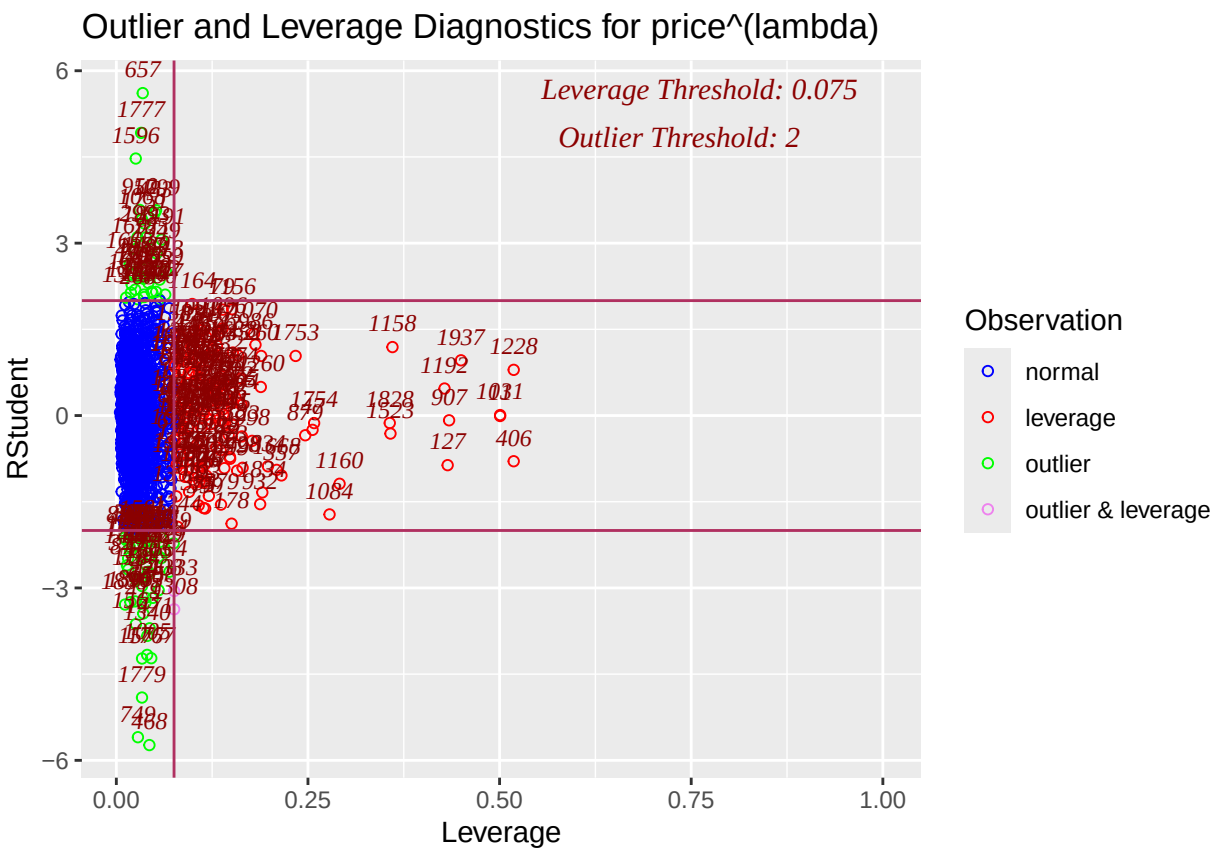

```
## [1] "Number of influentials - 2"
## [1] "Number of Observations 1942"
```



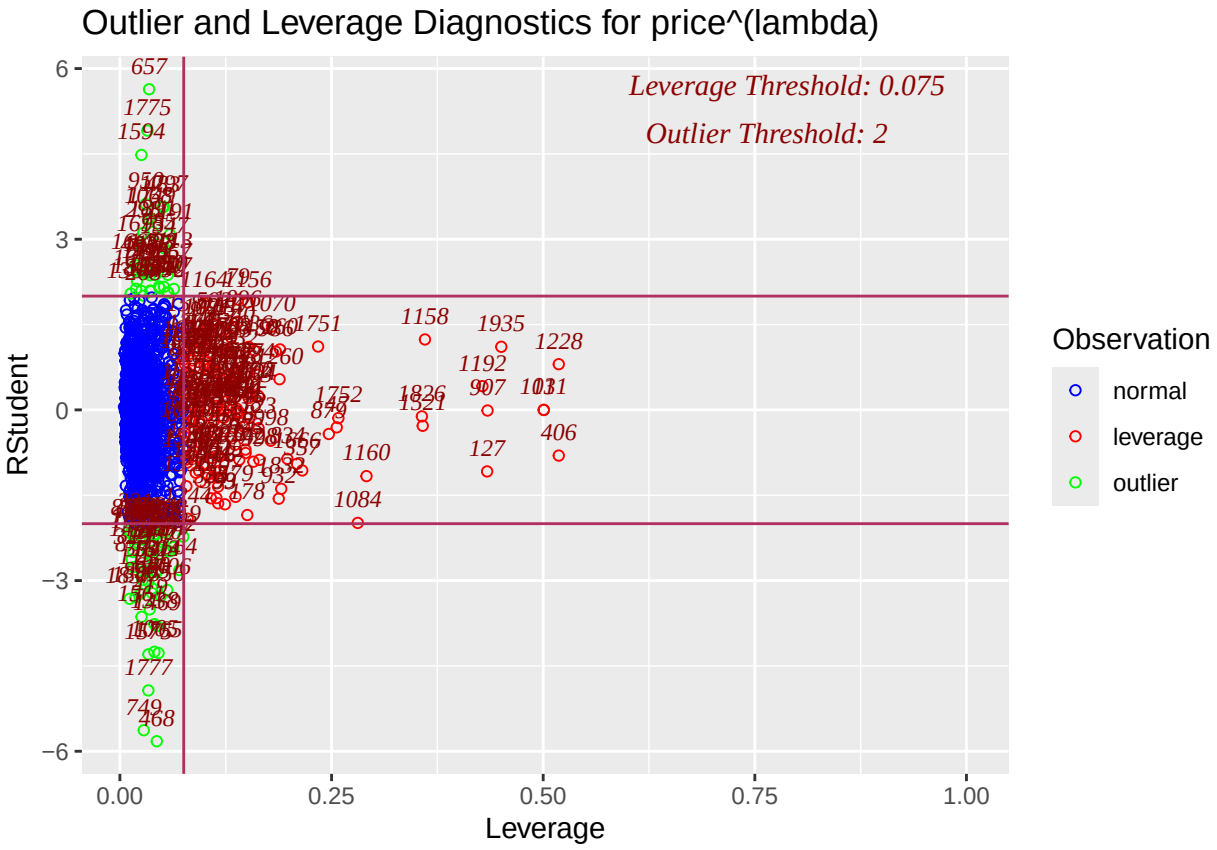

```
## [1] "Number of influentials - 2"
## [1] "Number of Observations 1940"
```




```
## [1] "Number of influentials - 1"
## [1] "Number of Observations 1939"
```




```
## [1] "Number of influentials - 2"
## [1] "Number of Observations 1937"
```

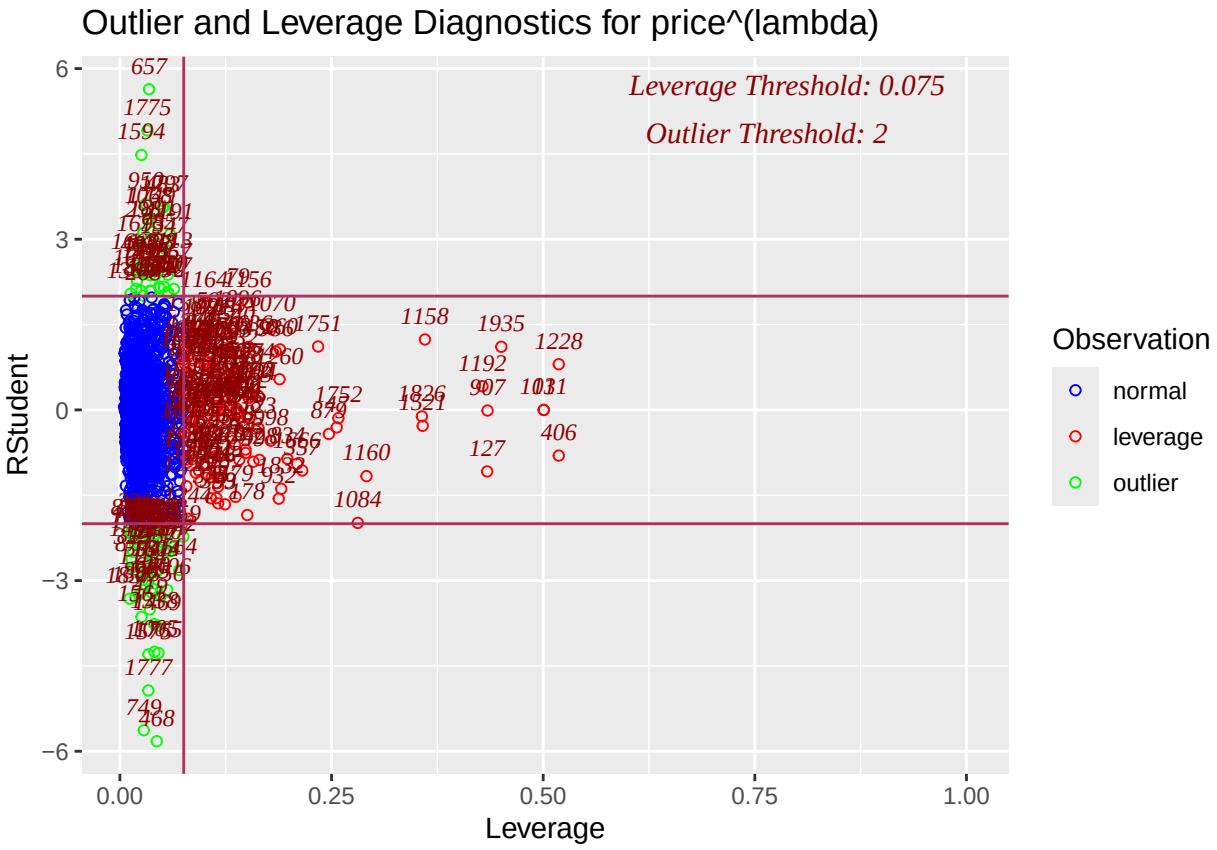



```
## [1] "Number of influentials - 0"
```

```
print(paste("Number of rows after removing influentials -",nrow(temp_data)))
```

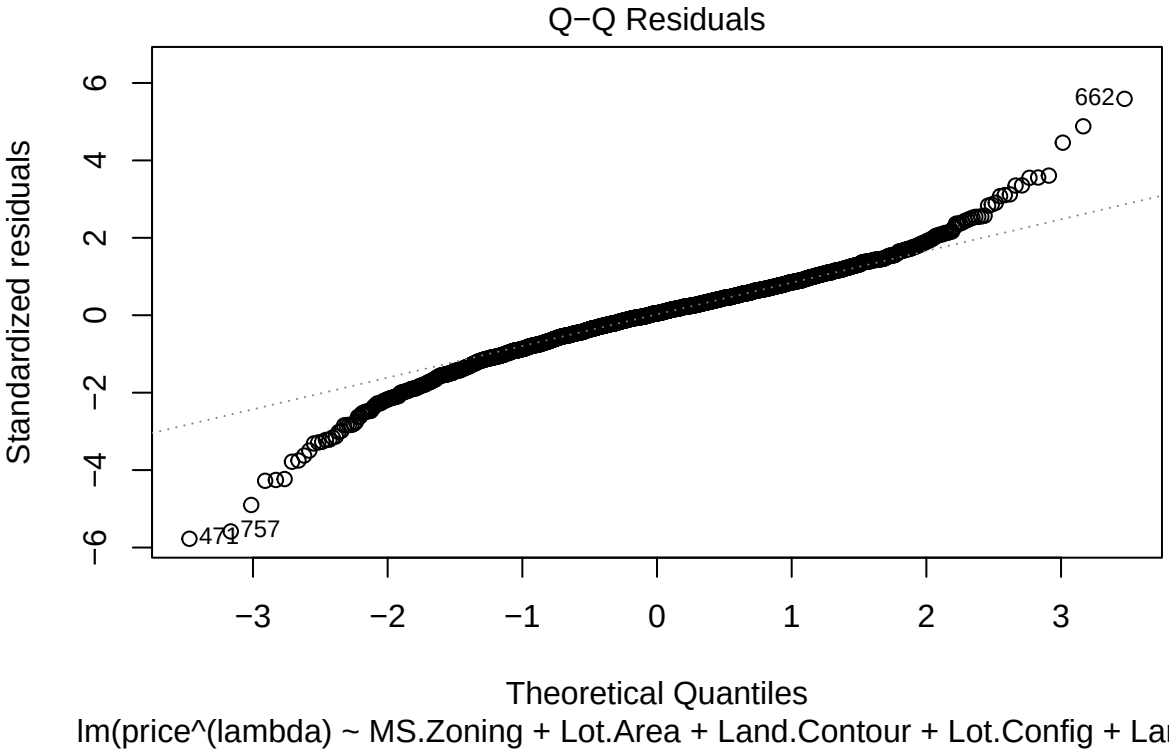
```
## [1] "Number of rows after removing influentials - 1937"
```

```
temp_model <- getRevisedModel(temp_data)
# this should show no influential obs
ols_plot_resid_lev(temp_model)
```




```
# checking tails
plot(temp_model, which = 2)
```

```
## Warning: not plotting observations with leverage one:
##   459, 521, 880, 1617
```




```
#temp_data <- removeOutliers(temp_model,temp_data)
#print(paste("Number of rows after removing outliers -",nrow(temp_data)))
#temp_model <- getRevisedModel(temp_data)
```

```r
# this should show no outlier obs
#ols_plot_cooksd_bar(temp_model)
# checking tails
#plot(temp_model, which = 2)
```

```r
#Huber Robust Method on transformed model
huber_lm <- rlm(price^(lambda) ~ MS.Zoning + Lot.Area + Land.Contour + Lot.Config +
    Land.Slope + Neighborhood + Condition.1 + Bldg.Type + Overall.Qual +
    Overall.Cond + Year.Built + Year.Remod.Add + Roof.Matl +
    Mas.Vnr.Type + Mas.Vnr.Area + Exter.Qual + Bsmt.Qual + Bsmt.Exposure +
    BsmtFin.SF.1 + BsmtFin.Type.2 + BsmtFin.SF.2 + Bsmt.Unf.SF +
    Central.Air + X1st.Flr.SF + X2nd.Flr.SF + Full.Bath + Half.Bath +
    Bedroom.AbvGr + Kitchen.AbvGr + Kitchen.Qual + Functional +
    Fireplaces + Garage.Area + Wood.Deck.SF + Screen.Porch +
    Sale.Condition, data = ames.train_data.df, psi = psi.huber)
summary(huber_lm)
```

```
##
## Call: rlm(formula = price^(lambda) ~ MS.Zoning + Lot.Area + Land.Contour +
##     Lot.Config + Land.Slope + Neighborhood + Condition.1 + Bldg.Type +
##     Overall.Qual + Overall.Cond + Year.Built + Year.Remod.Add +
##     Roof.Matl + Mas.Vnr.Type + Mas.Vnr.Area + Exter.Qual + Bsmt.Qual +
##     Bsmt.Exposure + BsmtFin.SF.1 + BsmtFin.Type.2 + BsmtFin.SF.2 +
##     Bsmt.Unf.SF + Central.Air + X1st.Flr.SF + X2nd.Flr.SF + Full.Bath +
##     Half.Bath + Bedroom.AbvGr + Kitchen.AbvGr + Kitchen.Qual +
##     Functional + Fireplaces + Garage.Area + Wood.Deck.SF + Screen.Porch +
##     Sale.Condition, data = ames.train_data.df, psi = psi.huber)
## Residuals:
##      Min       1Q   Median       3Q      Max
## -53.9695  -3.9457   0.2204   3.7934  38.1635
##
## Coefficients:
##                Value    Std. Error t value
## (Intercept)     91.6414   1.3742    66.6878
## MS.Zoning1       2.6868   0.4800     5.5972
## MS.Zoning2     -11.8192   1.6650    -7.0988
## MS.Zoning3       8.2642   0.8456     9.7727
## MS.Zoning4      -0.0344   1.5577    -0.0221
## Lot.Area         0.0001   0.0000     4.5350
## Land.Contour1   -1.8865   0.8334    -2.2637
## Land.Contour2    2.7859   0.8076     3.4498
## Land.Contour3   -1.2217   1.2922    -0.9454
## Lot.Config1     -0.8863   0.3930    -2.2553
## Lot.Config2      1.5391   0.6190     2.4866
## Lot.Config3     -1.7694   0.8423    -2.1008
## Lot.Config4     -0.2590   1.9365    -0.1338
## Land.Slope1     -0.4779   0.8926    -0.5354
## Land.Slope2     -4.1961   2.6084    -1.6087
## Neighborhood     0.2525   0.0324     7.7870
## Condition.1      0.3562   0.1547     2.3027
## Bldg.Type1      -7.5130   0.9019    -8.3298
## Bldg.Type2      -3.9658   0.6400    -6.1961
## Bldg.Type3      -1.3169   1.5071    -0.8738
## Bldg.Type4      -3.2882   1.2242    -2.6860
## Overall.Qual     1.6299   0.1397    11.6683
## Overall.Cond    -0.5240   0.1135    -4.6166
## Year.Built      -0.0185   0.0063    -2.9368
## Year.Remod.Add   0.0190   0.0099     1.9164
## Roof.Matl1      -4.3377   2.3247    -1.8660
## Roof.Matl2      -2.1886   1.7365    -1.2604
```

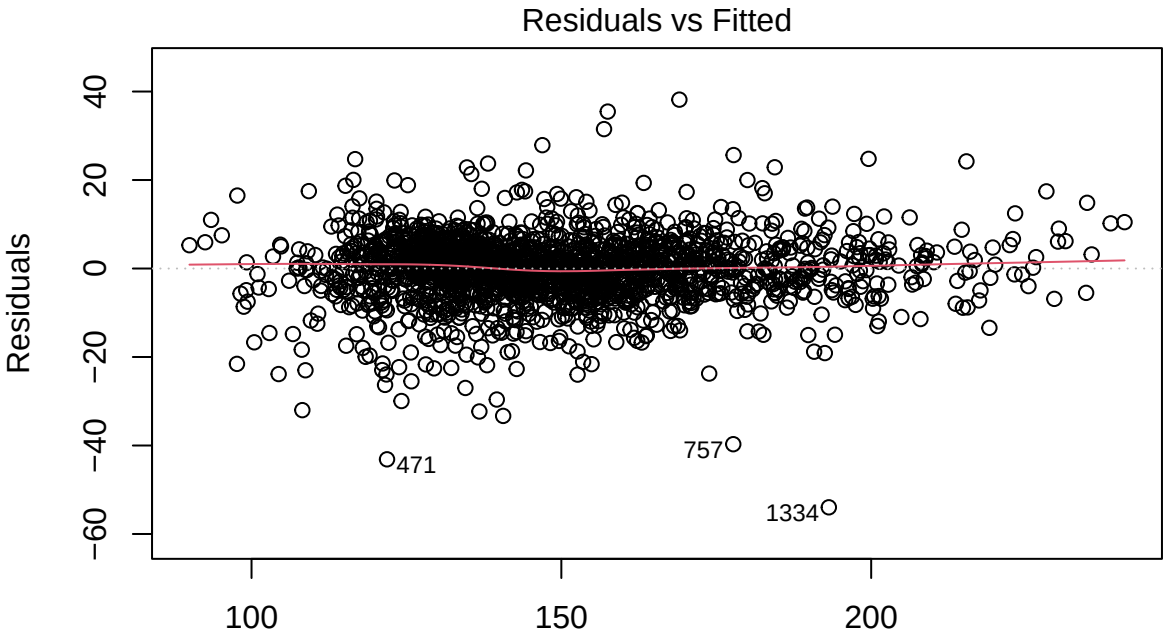
```
## Roof.Matl3       6.9008   3.4858     1.9797
## Roof.Matl4      -2.7834   6.4391    -0.4323
## Mas.Vnr.Type1   -0.6679   0.4551    -1.4674
## Mas.Vnr.Type2    2.1587   0.6520     3.3108
## Mas.Vnr.Type3   -3.9115   1.6548    -2.3637
## Mas.Vnr.Type4  -36.0053   6.3835    -5.6404
## Mas.Vnr.Area     0.0043   0.0013     3.4370
## Exter.Qual1      3.1163   0.5102     6.1075
## Exter.Qual2     -8.1289   1.6398    -4.9571
## Exter.Qual3      8.4612   1.1209     7.5488
## Bsmt.Qual1       8.1829   4.7898     1.7084
## Bsmt.Qual2       2.5848   0.4547     5.6847
## Bsmt.Qual3       7.3632   0.8293     8.8793
## Bsmt.Qual4      -2.9436   0.8654    -3.4015
## Bsmt.Exposure1  -0.3803   0.5301    -0.7174
## Bsmt.Exposure2   1.2097   0.4540     2.6648
## Bsmt.Exposure3   3.6358   0.5962     6.0982
## BsmtFin.SF.1     0.0113   0.0009    12.6018
## BsmtFin.Type.2  -0.2579   0.2069    -1.2466
## BsmtFin.SF.2     0.0087   0.0015     5.7139
## Bsmt.Unf.SF      0.0054   0.0009     6.1384
## Central.Air1     4.3958   0.7237     6.0737
## X1st.Flr.SF      0.0190   0.0009    20.1557
## X2nd.Flr.SF      0.0165   0.0006    25.9998
## Full.Bath1       1.5171   0.4658     3.2573
## Full.Bath2       3.9630   2.4688     1.6052
## Full.Bath3       5.6175   1.2327     4.5569
## Full.Bath4       7.2915   4.8941     1.4899
## Half.Bath1       0.9492   0.4334     2.1902
## Half.Bath2      -4.6839   1.9250    -2.4332
## Bedroom.AbvGr   -0.3143   0.1405    -2.2372
## Kitchen.AbvGr1  -7.6460   1.4587    -5.2417
## Kitchen.AbvGr2   2.8453   6.4436     0.4416
## Kitchen.AbvGr3  -2.3821   3.9656    -0.6007
## Kitchen.Qual1   -3.3247   0.4261    -7.8018
## Kitchen.Qual2   -4.2504   1.1059    -3.8433
## Kitchen.Qual3    4.6049   0.7703     5.9783
## Functional      -2.6439   0.2384   -11.0921
## Fireplaces1      1.9387   0.3548     5.4648
## Fireplaces2      3.0682   0.6500     4.7201
## Fireplaces3      0.0505   2.1773     0.0232
## Fireplaces4     -9.5967   6.5355    -1.4684
## Garage.Area      0.0104   0.0009    11.3730
## Wood.Deck.SF     0.0047   0.0013     3.7225
## Screen.Porch     0.0132   0.0026     5.0843
## Sale.Condition   0.2255   0.1548     1.4569
##
## Residual standard error: 5.711 on 1894 degrees of freedom
```

```
huber_lm$coefficients
```

```
##    (Intercept)     MS.Zoning1     MS.Zoning2     MS.Zoning3     MS.Zoning4
##  91.6413521560   2.6868082156 -11.8191859747   8.2642071342  -0.0343997371
##       Lot.Area  Land.Contour1  Land.Contour2  Land.Contour3    Lot.Config1
##   0.0001270221  -1.8865248359   2.7859175313  -1.2216555434  -0.8863371638
##    Lot.Config2    Lot.Config3    Lot.Config4    Land.Slope1    Land.Slope2
##   1.5390935695  -1.7694017963  -0.2590290585  -0.4779057728  -4.1960647783
##   Neighborhood    Condition.1     Bldg.Type1     Bldg.Type2     Bldg.Type3
##   0.2525218151   0.3561751545  -7.5130435813  -3.9657856618  -1.3168855172
##     Bldg.Type4   Overall.Qual   Overall.Cond     Year.Built Year.Remod.Add
##  -3.2882080052   1.6298902906  -0.5239662839  -0.0184574397   0.0189828938
```

```
##     Roof.Matl1     Roof.Matl2     Roof.Matl3     Roof.Matl4  Mas.Vnr.Type1
##  -4.3377408237  -2.1886133120   6.9007532394  -2.7833591588  -0.6678580191
##  Mas.Vnr.Type2  Mas.Vnr.Type3  Mas.Vnr.Type4   Mas.Vnr.Area    Exter.Qual1
##   2.1587033390  -3.9114649858 -36.0052800869   0.0043183690   3.1162826658
##    Exter.Qual2    Exter.Qual3     Bsmt.Qual1     Bsmt.Qual2     Bsmt.Qual3
##  -8.1289283548   8.4612112405   8.1828777460   2.5848277190   7.3631859912
##     Bsmt.Qual4 Bsmt.Exposure1 Bsmt.Exposure2 Bsmt.Exposure3   BsmtFin.SF.1
##  -2.9435678063  -0.3802988783   1.2096805074   3.6357901189   0.0113115397
## BsmtFin.Type.2   BsmtFin.SF.2    Bsmt.Unf.SF   Central.Air1    X1st.Flr.SF
##  -0.2579211224   0.0086857258   0.0053989724   4.3958005060   0.0190033390
##    X2nd.Flr.SF     Full.Bath1     Full.Bath2     Full.Bath3     Full.Bath4
##   0.0164870864   1.5171166919   3.9629577612   5.6174674339   7.2914840705
##     Half.Bath1     Half.Bath2  Bedroom.AbvGr Kitchen.AbvGr1 Kitchen.AbvGr2
##   0.9491805901  -4.6839228369  -0.3142523037  -7.6460219439   2.8452643447
## Kitchen.AbvGr3  Kitchen.Qual1  Kitchen.Qual2  Kitchen.Qual3     Functional
##  -2.3820574645  -3.3247456810  -4.2504279404   4.6048754231  -2.6438755087
##    Fireplaces1    Fireplaces2    Fireplaces3    Fireplaces4    Garage.Area
##   1.9387059876   3.0681721304   0.0504556034  -9.5966988295   0.0104427563
##   Wood.Deck.SF   Screen.Porch Sale.Condition
##   0.0046871297   0.0131588825   0.2255178644
```

```
#Constancy of the Error Variance
#Residuals vs Fitted and breusch pagan test show that Error Variance is
#not constant
plot(huber_lm, which=1)
```



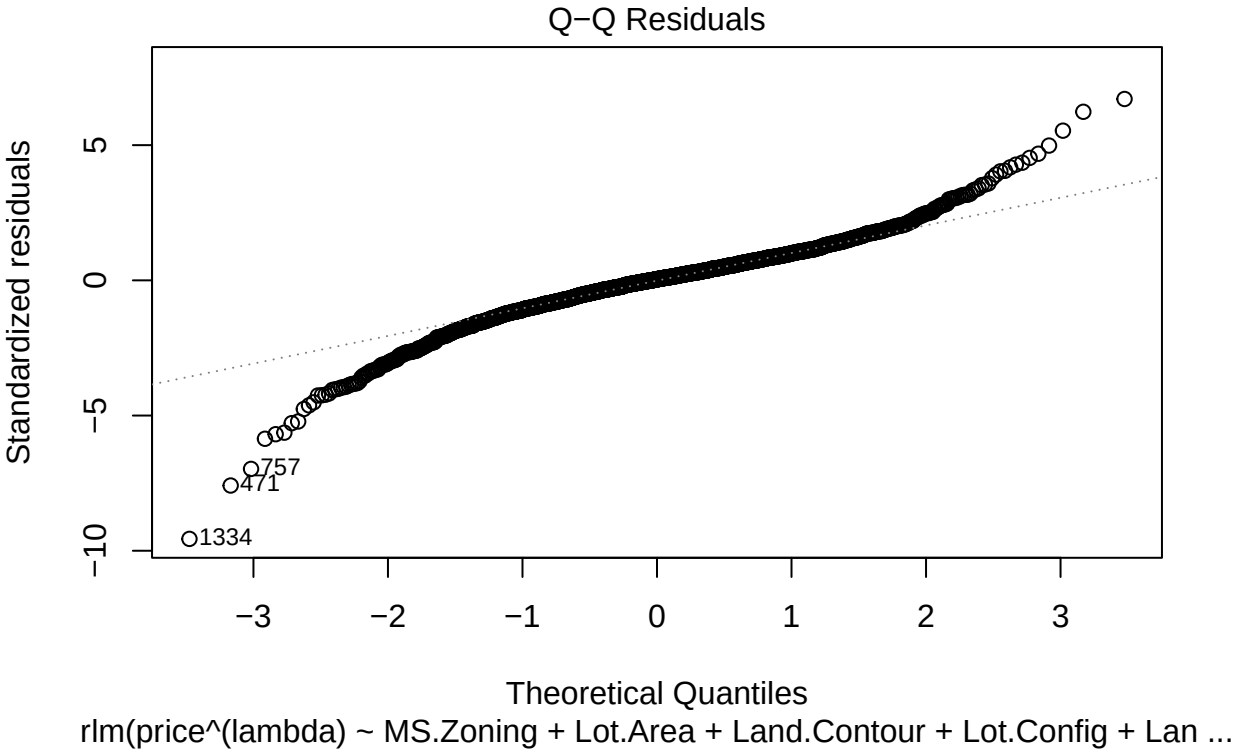

```
ols_test_breusch_pagan(huber_lm)
```

```
##
##  Breusch Pagan Test for Heteroskedasticity
##  -----------------------------------------
##  Ho: the variance is constant
##  Ha: the variance is not constant
##
##                   Data
##  -----------------------------------------
##  Response : price^(lambda)
```

```
##  Variables: fitted values of price^(lambda)
##
##        Test Summary
##  ---------------------------
##  DF            =    1
##  Chi2          =    0.1050719
##  Prob > Chi2   =    0.745826
```

```
#QQ Plot for Normality
#Normality plot and Shapiro test show that residuals
#do not fulfill normality assumption
plot(huber_lm, which = 2)
```

```
## Warning: not plotting observations with leverage one:
##   462, 524, 891, 1645
```




```
#Shapiro Test
shapiro.test(huber_lm$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  huber_lm$residuals
## W = 0.94521, p-value < 0.00000000000000022
```

8. Investigate unequal variances and multicollinearity. If necessary, apply remedial methods (WLS, Ridge, Elastic Net, Lasso, etc.).

These are on the high end of VIFs. None are greater than 10. remedial methods are not necessary. BsmtFin.SF.1: 7.887440 Bsmt.Unf.SF: 7.267218 Bldg.Type: 7.749723 X1st.Flr.SF: 6.223949 Exter.Qual: 6.279509

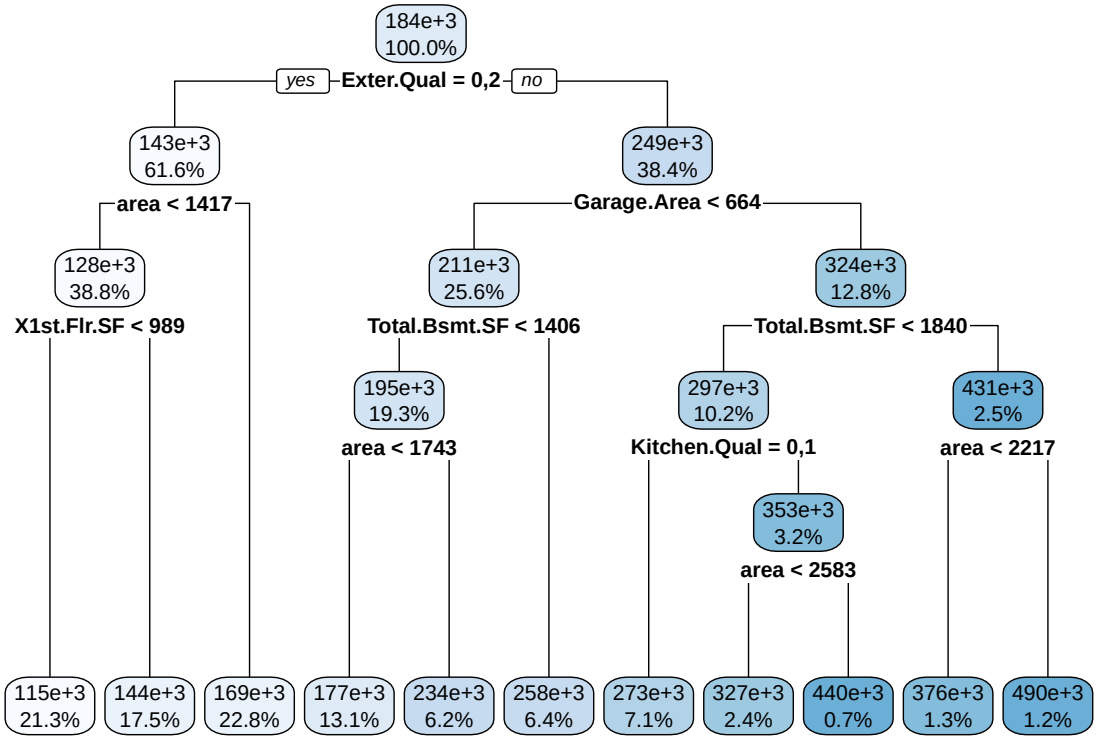
```
#According to VIF there seems to be multicollinearity issues
#since there are many VIFs greater than 10.

vif(transformed_model)
```

```
##                   GVIF Df GVIF^(1/(2*Df))
## MS.Zoning     2.662872  4        1.130235
## Lot.Area      1.880628  1        1.371360
```

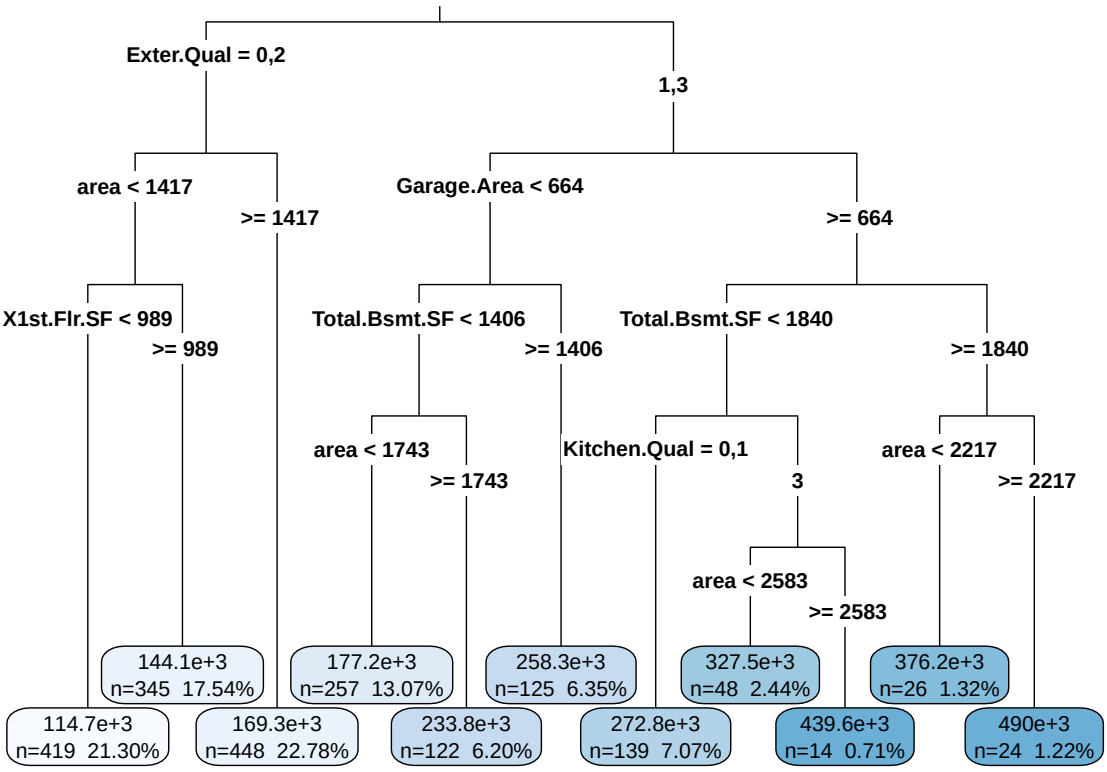
```
## Land.Contour   2.193854  3        1.139903
## Lot.Config     1.328221  4        1.036117
## Land.Slope     2.383592  2        1.242533
## Neighborhood   2.152699  1        1.467208
## Condition.1    1.090518  1        1.044279
## Bldg.Type      7.749723  4        1.291697
## Overall.Qual   2.428433  1        1.558343
## Overall.Cond   1.198958  1        1.094970
## Year.Built     1.253877  1        1.119767
## Year.Remod.Add 1.302311  1        1.141189
## Roof.Matl      1.727080  4        1.070691
## Mas.Vnr.Type   3.148530  4        1.154153
## Mas.Vnr.Area   2.583888  1        1.607448
## Exter.Qual     6.279509  3        1.358275
## Bsmt.Qual      6.149757  4        1.254895
## Bsmt.Exposure  1.892285  3        1.112153
## BsmtFin.SF.1   7.887440  1        2.808459
## BsmtFin.Type.2 2.449889  1        1.565212
## BsmtFin.SF.2   3.279165  1        1.810847
## Bsmt.Unf.SF    7.267218  1        2.695778
## Central.Air    1.455706  1        1.206526
## X1st.Flr.SF    6.223949  1        2.494784
## X2nd.Flr.SF    3.642971  1        1.908657
## Full.Bath      5.264357  4        1.230745
## Half.Bath      3.168249  2        1.334151
## Bedroom.AbvGr  1.179241  1        1.085929
## Kitchen.AbvGr  4.203934  3        1.270406
## Kitchen.Qual   4.533909  3        1.286507
## Functional     1.164028  1        1.078901
## Fireplaces     1.990375  4        1.089850
## Garage.Area    1.922141  1        1.386413
## Wood.Deck.SF   1.297785  1        1.139204
## Screen.Porch   1.131801  1        1.063861
## Sale.Condition 1.202447  1        1.096562
```

9. Build an alternative model based on one of the following approaches to predict price: regression tree, NN, or SVM. Then check again the applicable model assumptions.

```
set.seed(567)
par(mfrow=c(1,1))
ames.tree <- rpart(price ~ ., ames.train_data.df)
rpart.plot(ames.tree, digits=3)
```




```
rpart.plot(ames.tree, digits=4, fallen.leaves=TRUE, type=3, extra=101)
```




```
ames.tree
```

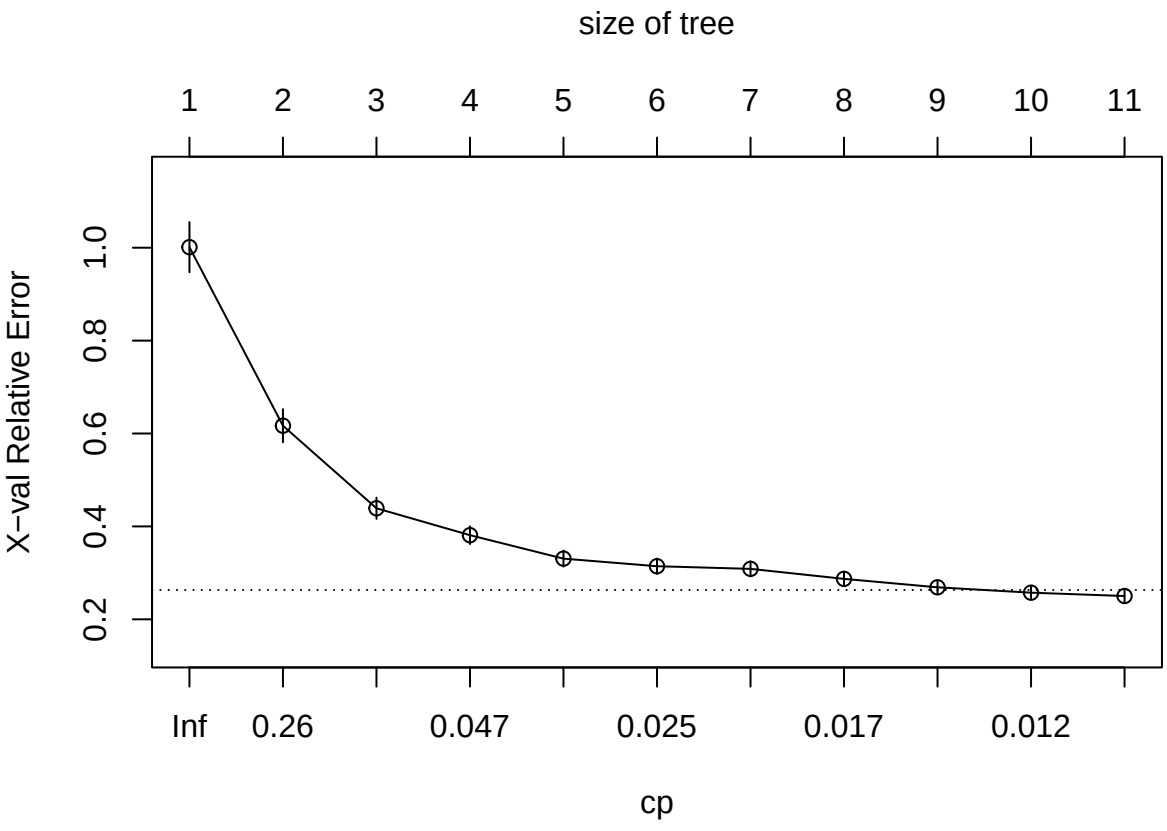
```
## n= 1967
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 1967 12568030000000 183675.7
##    2) Exter.Qual=0,2 1212  1839787000000 143235.6
##      4) area< 1417 764   633886400000 127979.7
```

```
##        8) X1st.Flr.SF< 989 419   253798000000 114716.6 *
##        9) X1st.Flr.SF>=989 345   216866200000 144087.6 *
##      5) area>=1417 448   724846500000 169252.3 *
##    3) Exter.Qual=1,3 755  5564264000000 248594.1
##      6) Garage.Area< 664 504  1486194000000 211031.5
##       12) Total.Bsmt.SF< 1405.5 379   722159000000 195446.6
##         24) area< 1742.5 257   274045100000 177241.9 *
##         25) area>=1742.5 122   183521500000 233795.8 *
##       13) Total.Bsmt.SF>=1405.5 125   392867900000 258285.1 *
##      7) Garage.Area>=664 251  1939051000000 324018.6
##       14) Total.Bsmt.SF< 1840 201   891008100000 297452.5
##         28) Kitchen.Qual=0,1 139   352898200000 272777.8 *
##         29) Kitchen.Qual=3 62   263749200000 352771.5
##           58) area< 2582.5 48    86511560000 327451.0 *
##           59) area>=2582.5 14    40952200000 439584.6 *
##       15) Total.Bsmt.SF>=1840 50   335919500000 430814.3
##         30) area< 2217 26    35015420000 376162.1 *
##         31) area>=2217 24   139115800000 490020.8 *
```

```
ames.tree$cptable
```

```
##            CP nsplit rel error    xerror       xstd
## 1  0.41088233      0 1.0000000 1.0011611 0.05401400
## 2  0.17019516      1 0.5891177 0.6167651 0.03561979
## 3  0.05666150      2 0.4189225 0.4390413 0.02288978
## 4  0.03827598      3 0.3622610 0.3811251 0.01911218
## 5  0.02953265      4 0.3239850 0.3306533 0.01721018
## 6  0.02183004      5 0.2944524 0.3142835 0.01648474
## 7  0.02105281      6 0.2726223 0.3086299 0.01624706
## 8  0.01298709      7 0.2515695 0.2870863 0.01454585
## 9  0.01287300      8 0.2385824 0.2688898 0.01376724
## 10 0.01084382      9 0.2257094 0.2573372 0.01329347
## 11 0.01000000     10 0.2148656 0.2501477 0.01306803
```

```
plotcp(ames.tree)
```

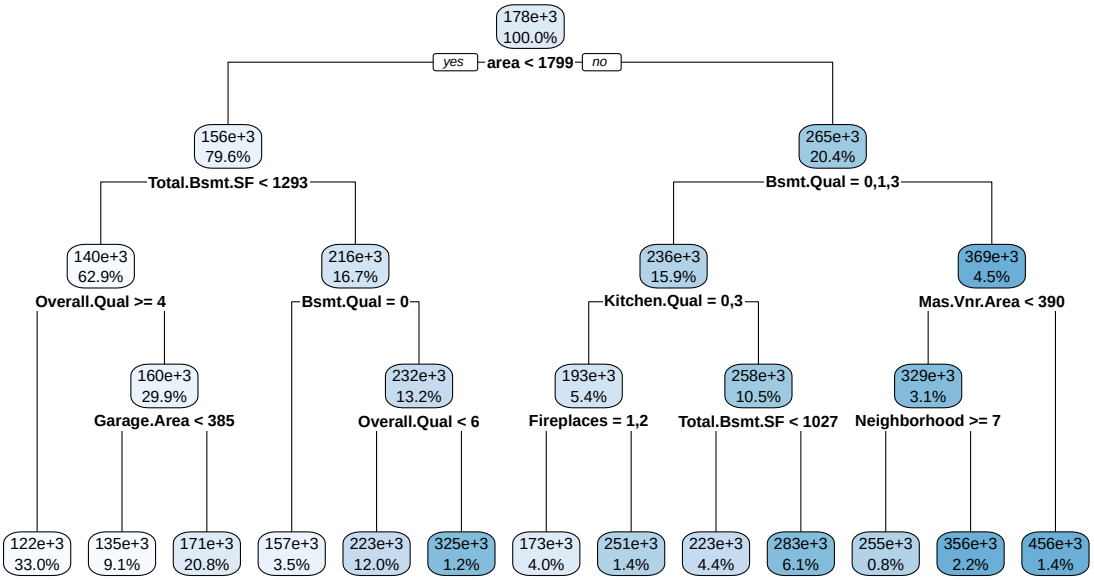


```
set.seed(567)
PredictedTest <- predict(ames.tree,ames.train_data.df)
ModelTest2 <- data.frame(obs = ames.train_data.df$price, pred=PredictedTest)
defaultSummary(ModelTest2)
```

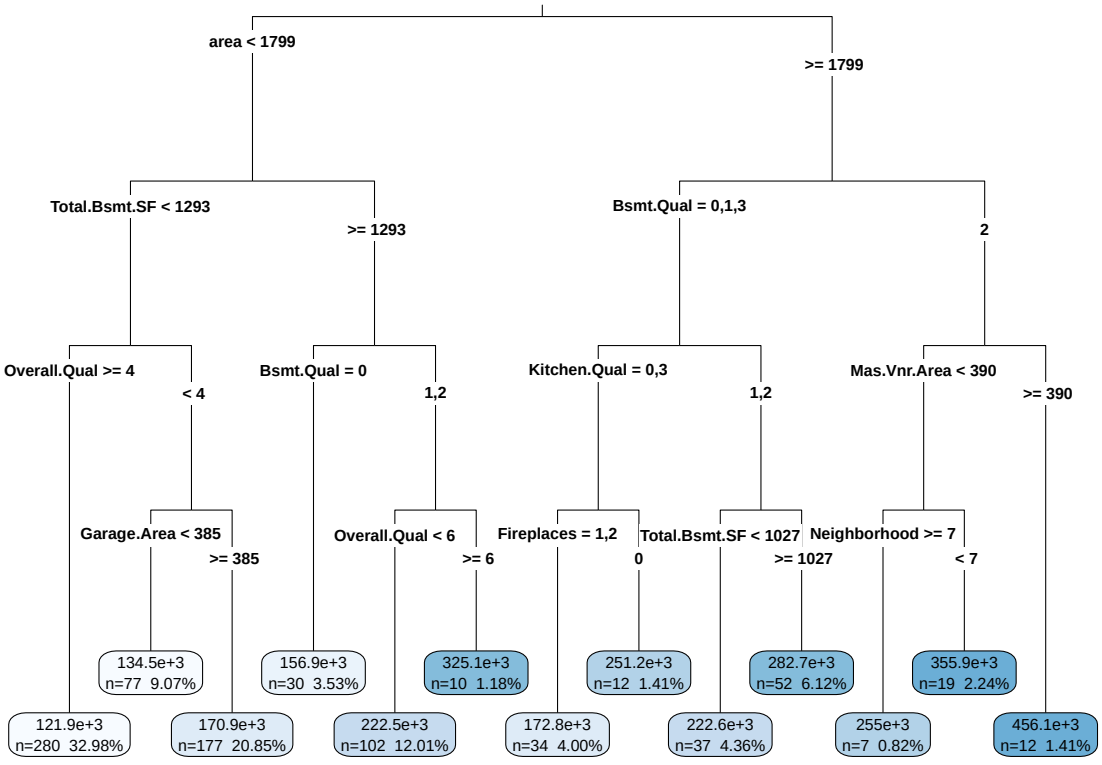
```
##           RMSE       Rsquared            MAE
## 37052.2810425      0.7851344 27149.9039110
```

10. Use the test data set to assess the model performances from above.

```
set.seed(567)
par(mfrow=c(1,1))
ames.tree <- rpart(price ~ ., ames.test_data.df)
rpart.plot(ames.tree, digits=3)
```




```
rpart.plot(ames.tree, digits=4, fallen.leaves=TRUE, type=3, extra=101)
```



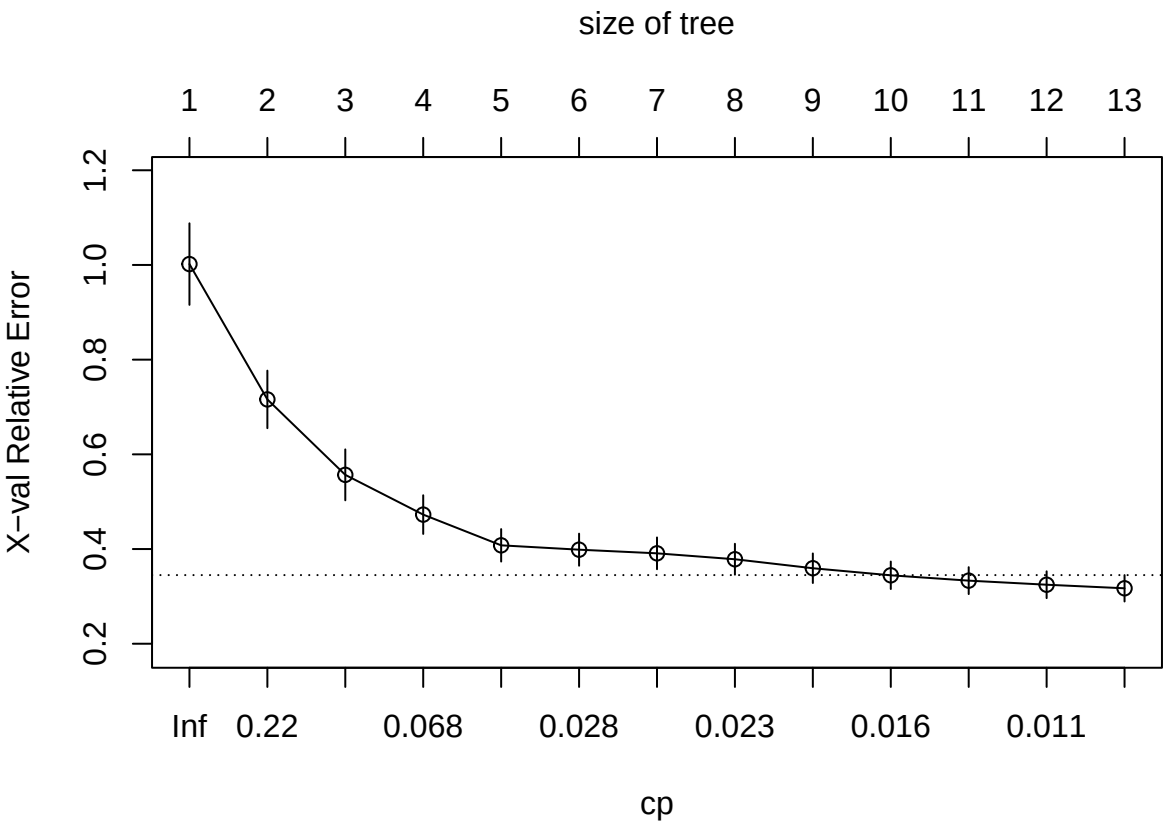
```
ames.tree
```

```
## n= 849
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 849 4699010000000 178141.4
##    2) area< 1799 676 1688466000000 155913.1
##      4) Total.Bsmt.SF< 1292.5 534  623299500000 139965.8
##        8) Overall.Qual>=3.5 280  212239100000 121917.1 *
##        9) Overall.Qual< 3.5 254  219299300000 159862.1
##         18) Garage.Area< 385 77   49857640000 134526.5 *
##         19) Garage.Area>=385 177   98514570000 170883.8 *
##      5) Total.Bsmt.SF>=1292.5 142  418657700000 215884.0
##       10) Bsmt.Qual=0 30   28922040000 156948.3 *
##       11) Bsmt.Qual=1,2 112  257621900000 231670.3
##         22) Overall.Qual< 5.5 102  150609800000 222510.3 *
##         23) Overall.Qual>=5.5 10   11157170000 325102.9 *
##    3) area>=1799 173 1371399000000 264998.4
##      6) Bsmt.Qual=0,1,3 135  510311500000 235734.6
##       12) Kitchen.Qual=0,3 46  127089600000 193273.4
##         24) Fireplaces=1,2 34   49647700000 172841.2 *
##         25) Fireplaces=0 12   23030900000 251164.7 *
##       13) Kitchen.Qual=1,2 89  257420300000 257680.8
##         26) Total.Bsmt.SF< 1027 37   81904000000 222575.9 *
##         27) Total.Bsmt.SF>=1027 52   97475130000 282659.3 *
##      7) Bsmt.Qual=2 38  334756400000 368962.1
##       14) Mas.Vnr.Area< 390 26  105859000000 328730.6
##         28) Neighborhood>=6.5 7    3417505000 254955.1 *
##         29) Neighborhood< 6.5 19   50305050000 355911.0 *
##       15) Mas.Vnr.Area>=390 12   95635050000 456130.2 *
```

```
ames.tree$cptable
```

```
##            CP nsplit rel error    xerror       xstd
## 1  0.34882785      0 1.0000000 1.0018570 0.08614343
## 2  0.13758397      1 0.6511721 0.7160321 0.06078854
## 3  0.11200886      2 0.5135882 0.5567037 0.05379602
## 4  0.04080882      3 0.4015793 0.4727173 0.04086431
## 5  0.02835968      4 0.3607705 0.4078483 0.03429484
## 6  0.02811522      5 0.3324108 0.3986344 0.03388843
## 7  0.02677195      6 0.3042956 0.3909261 0.03356184
## 8  0.02039896      7 0.2775237 0.3784389 0.03267146
## 9  0.01660800      8 0.2571247 0.3594739 0.03124418
## 10 0.01509406      9 0.2405167 0.3445364 0.02902458
## 11 0.01157924     10 0.2254226 0.3332687 0.02842201
## 12 0.01109520     11 0.2138434 0.3246472 0.02834828
## 13 0.01000000     12 0.2027482 0.3170865 0.02788692
```

```
plotcp(ames.tree)
```




```
set.seed(567)
PredictedTest2 <- predict(ames.tree,ames.test_data.df)
ModelTest2.2 <- data.frame(obs = ames.test_data.df$price, pred=PredictedTest2)
defaultSummary(ModelTest2.2)
```

```
##          RMSE      Rsquared           MAE
## 33498.6894389     0.7972518 24653.4045984
```

11. Based on the performances on both train and test data sets, determine your primary (champion) model and the other model which would be your benchmark model.

Transformed Model result using test set

```
# Summary for transformed linear model using train set
PredictedTest1 <- predict(transformed_model,ames.train_data.df)
ModelTest1 <- data.frame(obs = ames.train_data.df$price, pred=PredictedTest1)
defaultSummary(ModelTest1)
```

```
##           RMSE       Rsquared            MAE
## 200170.0996838      0.9012827 183527.3078814
```

```
# Build full model with all predictors
fm <- lm(price^(lambda) ~ MS.Zoning + Lot.Area + Land.Contour + Lot.Config +
    Land.Slope + Neighborhood + Condition.1 + Bldg.Type + Overall.Qual +
    Overall.Cond + Year.Built + Year.Remod.Add + Roof.Matl +
    Mas.Vnr.Type + Mas.Vnr.Area + Exter.Qual + Bsmt.Qual + Bsmt.Exposure +
    BsmtFin.SF.1 + BsmtFin.Type.2 + BsmtFin.SF.2 + Bsmt.Unf.SF +
    Central.Air + X1st.Flr.SF + X2nd.Flr.SF + Full.Bath + Half.Bath +
    Bedroom.AbvGr + Kitchen.AbvGr + Kitchen.Qual + Functional +
    Fireplaces + Garage.Area + Wood.Deck.SF + Screen.Porch +
    Sale.Condition, data = ames.test_data.df)
# Build null model with no predictors
nm <- lm(price^(lambda) ~ 1, data = ames.test_data.df)

# Stepwise selection using both AIC and BIC
sm_both_aic <- step(fm, direction = "both", scope = list(lower = nm, upper = fm), trace = FALSE, k = 2)
```

```
PredictedTest1.2 <- predict(sm_both_aic,ames.test_data.df)
ModelTest1.2 <- data.frame(obs = ames.test_data.df$price, pred=PredictedTest1.2)
defaultSummary(ModelTest1.2)
```

```
##           RMSE       Rsquared            MAE
## 192908.4779778      0.8805362 177994.6637578
```

```
out <- rbind(defaultSummary(ModelTest1),defaultSummary(ModelTest1.2),defaultSummary(ModelTest2),defaultSummary
dimnames(out)[[1]] <- c("Best Linear (Train)","Best Linear (Test)","Best Regression Tree (Train)","Best Regres
out
```

```
##                                  RMSE  Rsquared      MAE
## Best Linear (Train)          200170.10 0.9012827 183527.3
## Best Linear (Test)           192908.48 0.8805362 177994.7
## Best Regression Tree (Train)  37052.28 0.7851344  27149.9
## Best Regression Tree (Test)   33498.69 0.7972518  24653.4
```

Based on the observations above, the champion model should be the Linear Regression Model.

12. Create a model development document that describes the model following this template, input the name of the authors, Harvard IDs, the name of the Group, all of your code and calculations, etc.: be sure you populate all the sections of this template.

13. Each student must submit the files on Canvas to get the full credit.

This data was taken from the Kaggle competition please click on the link below for details: https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques/rules

Table 2: ***Executive Summary***

This section will describe the model usage, your conclusions and any regulatory and internal requirements. In a real world scenario, this section is for senior management who do not need to know the details. They need to know high level (the purpose of the model, limitations of the model and any issues).

I. Introduction (5 points)

This section needs to introduce the reader to the problem to be resolved, the purpose, and the scope of the statistical testing applied. What you are doing with your prediction? What is the purpose of the model? What methods were trained on the data, how large is the test sample, and how did you build the model?

Based on the ames housing dataset provided, we have to build a model that can predict housing prices based on 79 explanotory variables. The scope of our testing is limited to the dataset provided. The test sample is about 879 observations whereas the training model is about 2050 observations.

The model is trained on various regression methods such as stepwise regression and regression tree model.

I. Description of the data and quality (15 points)

Here you need to include your review of data, the statistical test applied to understand the predictors and the response and how are they correlated. Extensive graph analysis is recommended. Is the data continuous, or categorical, do any transformation needed? Do you need dummies?

The data contains 43 categorical variables and 36 continuous variables. Some columns also have missing values. Our first data cleaning task was to figure out which columns had a lot of missing values. We dropped those columns. We then cleaned data for the columns that have very few missing values. We also looked at categorical columns that have more than 5 unique values. We treat such columns as continuous variables. The other categorical variables (that have less than or equal to 5 unique values) are marked as factors so that the model training algorithms process them as categorical variables. We also removed rows that had houses with more than 4000 square feet based on the recommendation provided on the official website for the dataset. A chi-squared test to estimate the significance of each categorical variable (post missing values removal) with respect to the response variable was done to eliminate some more categorical columns from the data.

III. Model Development Process (15 points)

Build a regression model to predict price. And of course, create the train data set which contains 70% of the data and use set.seed (1023). The remaining 30% will be your test data set. Investigate the data and combine the level of categorical variables if needed and drop variables. For example, you can drop id, Latitude, Longitude, etc.

We used the stepwise selection method to determine which are the best predictors in the data. Once we derived a model, we checked for the normality assumptions and transformed the response variable using box cox method. Since the lambda was close to 0, we used log transformation. We also carried out extensive analysis of looking into outliers and influentials. We removed all the influentials using a recursive function. The data contains a lot of outliers. We tried out removing all the outliers using a recursive function, but that removed a lot of rows and felt like a case of overfitting. We also looked at RMSE,Rsquared,MAE and diagonal yhat values. Further we also carried out Robust regression as a remedial measure.

IV. Model Performance Testing (15 points)

Use the test data set to assess the model performances. Here, build the best multiple linear models by using the stepwise both ways selection method. Compare the performance of the best two linear models. Make sure that model assumption(s) are checked for the final linear model. Apply remedy measures (transformation, etc.) that helps satisfy the assumptions. In particular you must deeply investigate unequal variances and multicollinearity. If necessary, apply remedial methods (WLS, Ridge, Elastic Net, Lasso, etc.).

We carried out performance test on the model chosen from stepwise method using the test data. We checked for the normality assumptions and transformed the response variable using box cox method. We also carried out extensive analysis of looking into outliers and influentials. We removed all the influentials using a recursive function. The data contains a lot of outliers. We tried out removing all the outliers using a recursive function, but that removed a lot of rows and felt like a case of overfitting. We also looked at RMSE,Rsquared,MAE and diagonal yhat values. Further we also carried out Robust regression as a remedial measure.

V. Challenger Models (15 points)

Build an alternative model based on one of the following approaches to predict price: regression tree, NN, or SVM or regression model with alternative variables. Always check the applicable model assumptions. Apply in-sample and out-of-sample testing, back testing and review the comparative goodness of fit of the candidate models. Describe step by step your procedure to get to the best model and why you believe it is fit for purpose.

We tried out Nueral network model and regression tree model as alternatives. Ultimately, we decided to keep the regression tree model as our challenger as it seemed to do better based than the NN model.

We carried out basic checks on model assumptions. We also did testing using the train and test data to compare performance of the model.

VI. Model Limitation and Assumptions (15 points)

Based on the performances on both train and test data sets, determine your primary (champion) model and the other model which would be your benchmark model. Validate your models using the test sample. Do the residuals look normal? Does it matter given your technique? How is the prediction performance using Pseudo R^2, SSE, ***RMSE****? Benchmark the model against alternatives. How good is the relative fit? Are there any serious violations of the model assumptions? Has the model had issues or limitations that the user must know? (Which assumptions are needed to support the Champion model?)*

Based on the benchmarking done on both the models, we think that the original model derived using stepwise method performed better and hence we choose that as our primary model. The primary limitation we have faced with the model is that the data had too many categorical values and better domain knowledge could have perhaps helped us clean it better. Also lack of time and relatively less experience with cleaning such datasets, we feel like our models could be skewed.

VII. Ongoing Model Monitoring Plan (5 points)

How would you picture the model needing to be monitored, which quantitative thresholds and triggers would you set to decide when the model needs to be replaced? What are the assumptions that the model must comply with for its continuous use?

Once deployed to production, the model will need to be continuosly monitored for performance on new data in production. We can setup on going or a set frequency based monitoring to derive various metrics such as model assumptions, outliers, null values and other business KPIs that are important to the use case at hand. If the model starts underperforming we would want to recalibrate it by performing more iterations of model refinement. Also, we would have to consider various domain

factors as the model ages. If with time, certain factors that we initially excluded from the model become more relevant we might have to include them back.

VIII. Conclusion (5 points)

Summarize your results here. What is the best model for the data and why?

Based on our analysis the model we derived from stepwise regression turned out to be the better one. It contains the below predictors -

```
MS.Zoning + Lot.Area + Land.Contour + Lot.Config +
Land.Slope + Neighborhood + Condition.1 + Bldg.Type + Overall.Qual +
Overall.Cond + Year.Built + Year.Remod.Add + Roof.Matl +
Mas.Vnr.Type + Mas.Vnr.Area + Exter.Qual + Bsmt.Qual + Bsmt.Exposure +
BsmtFin.SF.1 + BsmtFin.Type.2 + BsmtFin.SF.2 + Bsmt.Unf.SF +
Central.Air + X1st.Flr.SF + X2nd.Flr.SF + Full.Bath + Half.Bath +
Bedroom.AbvGr + Kitchen.AbvGr + Kitchen.Qual + Functional +
Fireplaces + Garage.Area + Wood.Deck.SF + Screen.Porch +
Sale.Condition, data = ames.test_data.df
```

Bibliography (7 points)

Please include all references, articles and papers in this section.

Data Set Information 1

Data Set Information 2

R basics

R resources 1

R resources 2

R resources 3

R resources 4

Categorical variables - reference 1

Categorical variables - reference 2

Data cleaning on housing price data set

Kaggle forums on housing price competition

Appendix (3 points)

Please add any additional supporting graphs, plots and data analysis.