STATS 101 Project

Sherry Shen, Lisa Zhong, Selina Zhang

```r
set.seed(123)     #setting seed
knitr::opts_chunk$set(echo = TRUE, message=FALSE, warning = FALSE)
```

##Data preprocessing

```r
data <- read.csv("ames2000_NAfix.csv", stringsAsFactors = TRUE)
data$MS.SubClass = factor(data$MS.SubClass)
```

```r
Convert_ordinal <- function(dict, row) {
  data[, row] = as.character(data[, row])
  for (i in 1:2000) {
     data[i, row] = dict[data[i, row]]
  }
  return (as.integer(data[, row])) # ordinal to their values
}

Lot.Shape.Order = c("IR3" = "1", "IR2" = "2", "IR1" = "3", "Reg" = "4")
data[, 7] <- Convert_ordinal(Lot.Shape.Order, 7)

Land.Slope.Order = c("Sev" = "1", "Mod" = "2", "Gtl" = "3")
data[, 11] <- Convert_ordinal(Land.Slope.Order, 11)

# data$Overall.Qual = factor(data$Overall.Qual)
# data$Overall.Cond = factor(data$Overall.Cond)

Qual.Order = c("Po" = "1", "Fa" = "2", "TA" = "3", "Gd" = "4", "Ex" = "5")
data[, 27] <- Convert_ordinal(Qual.Order, 27) # Exter.Qual to numerical order
data[, 28] <- Convert_ordinal(Qual.Order, 28) # Exter.Cond to numerical order
data[, 30] <- Convert_ordinal(Qual.Order, 30) # Bsmt.Qual to numerical order
data[, 31] <- Convert_ordinal(Qual.Order, 31) # Bsmt.Cond to numerical order
data[, 40] <- Convert_ordinal(Qual.Order, 40) # HeatingQC to numerical order
data[, 53] <- Convert_ordinal(Qual.Order, 53) # KitchenQual to numerical order
data[, 57] <- Convert_ordinal(Qual.Order, 57) # FireplaceQu to numerical order
data[, 63] <- Convert_ordinal(Qual.Order, 63) # Garage.Qual to numerical order
data[, 64] <- Convert_ordinal(Qual.Order, 64) # Garage.Cond to numerical order
data[, 72] <- Convert_ordinal(Qual.Order, 72) # Pool.QC to numerical order

Bsmt.Exposure.Order = c("No" = "1", "Mn" = "2", "Av" = "3", "Gd" = "4")
data[, 32] <- Convert_ordinal(Bsmt.Exposure.Order, 32) #Bsmt.Exposure to numerical order

BsmtFin.Order = c("Unf" = "1", "LwQ" = "2", "Rec" = "3", "BLQ" = "4", "ALQ", "5", "GLQ" = "6")
data[, 33] <- Convert_ordinal(BsmtFin.Order, 33) # BsmtFin.Type.1 to numerical order
data[, 35] <- Convert_ordinal(BsmtFin.Order, 35) # BsmtFin.Type.2 to numerical order
```

```r
Electrical.Order = c("Mix" = "1", "FuseP" = "2", "FuseF" = "3", "FuseA" = "4", "SBrkr" = "5")
data[, 42] <- Convert_ordinal(Electrical.Order, 42) # Electrical to numerical order

Functional.Order = c("Sal" = "1", "Sev" = "2", "Maj2" = "3", "Maj1" = "4", "Mod" = "5", "Min2" = "6", "
data[, 55] <- Convert_ordinal(Functional.Order, 55) # Functional to numerical order

Garage.Finish.Order = c("Unf" = "1", "RFn" = "2", "Fin" = "3")
data[, 60] <- Convert_ordinal(Garage.Finish.Order, 60) # Garage Finish to numerical order

Paved.Drive.Order = c("N" = "1", "P" = "2", "Y" = "3")
data[, 65] <- Convert_ordinal(Paved.Drive.Order, 65) # Paved Drive to numerical order

Fence.Order = c("MnWw" = "1", "GdWo" = "2", "MnPrv" = "3", "GdPrv" = "4")
data[, 73] <- Convert_ordinal(Fence.Order, 73) # Fence to numerical order
```

```r
# Bsmt.Full.Bath and the following variables should be factored or kept as numerical?

data$Lot.Frontage = as.integer(data$Lot.Frontage)
data$Mas.Vnr.Area = as.integer(data$Mas.Vnr.Area)
data$BsmtFin.SF.1 = as.integer(data$BsmtFin.SF.1)
data$BsmtFin.SF.2 = as.integer(data$BsmtFin.SF.2)
data$Bsmt.Unf.SF = as.integer(data$Bsmt.Unf.SF)
data$Total.Bsmt.SF = as.integer(data$Total.Bsmt.SF)
data$Garage.Yr.Blt = as.integer(data$Garage.Yr.Blt)
data$Garage.Area = as.integer(data$Garage.Area)
```

```r
for (y in 1:ncol(data)) {
  for (x in 1:nrow(data)) {
    if (!is.na(data[x,y])){
      if (as.character(data[x, y]) == "None") {
        data[x, y] = NA
      }
    }
  }
}
```

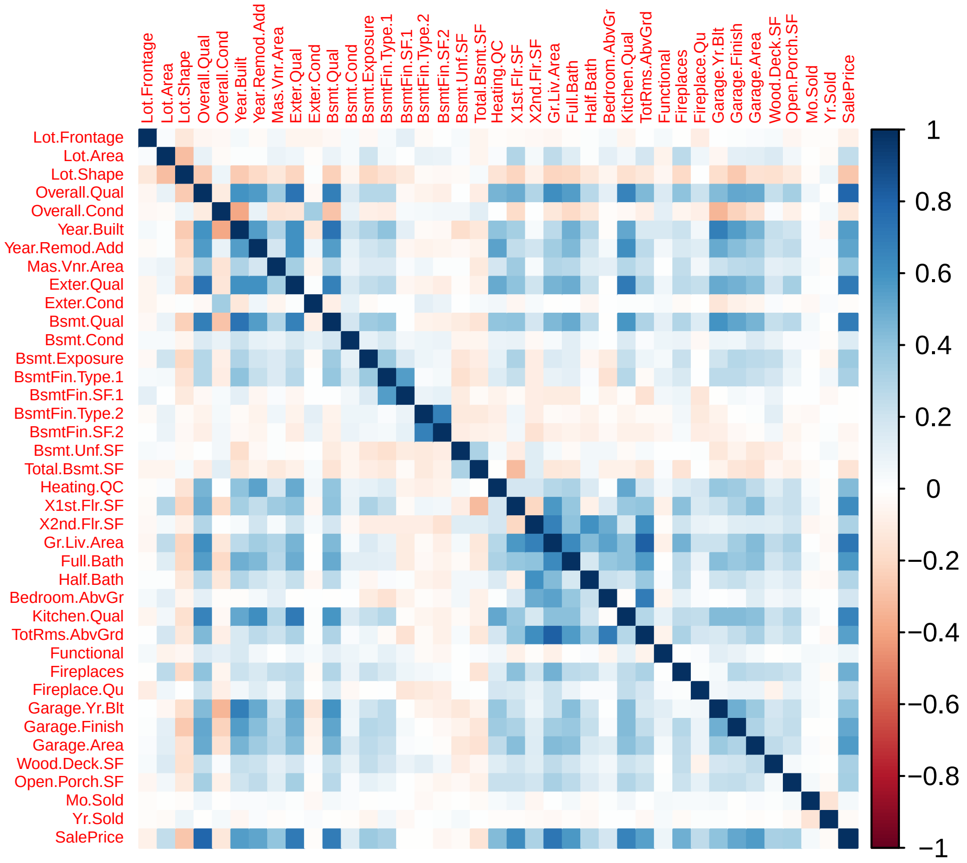
```r
sale_price <- data[, 80] # extracting sale price (y)

to_remove = c(5, 6, 8, 9, 11, 14, 22, 39, 41, 42, 45, 48, 52, 63, 64, 65, 68, 69, 70, 71, 72, 73, 74, 75

data <- data[,-to_remove] # removing the ones we don't want
```

##Splitting the data

```r
smp_size = floor(0.5 * nrow(data))
train_Index = sample(seq_len(nrow(data)), size = smp_size)
trainSale <- sale_price[train_Index]
testSale <- sale_price[-train_Index]
trainData <- data[train_Index, ]
testData <- data[-train_Index, ]

treeTrainData = trainData
treeTestData = testData
```

##Corrplot

```
require(corrplot)
cont_index <- c()
for (i in 1:ncol(trainData)) {
  if (!is.factor(data[, i])) {
    cont_index = c(cont_index, i)
  }
}

for (y in cont_index) {
  for (x in 1:length(trainData[, y])) {
    if (is.na(trainData[x, y])){
      trainData[x, y] = mean(trainData[, y], na.rm = TRUE)
    }
  }
}

correlation <- cor(trainData[cont_index])
corrplot(correlation, method = "color", tl.cex = 0.5)
```

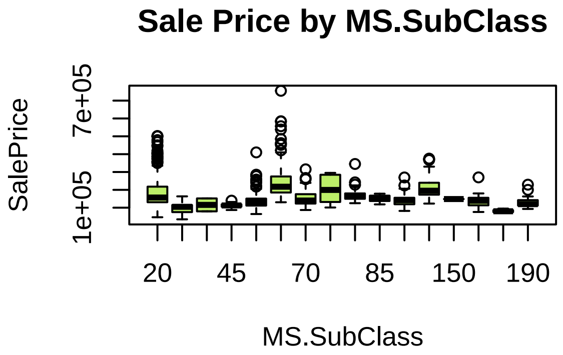


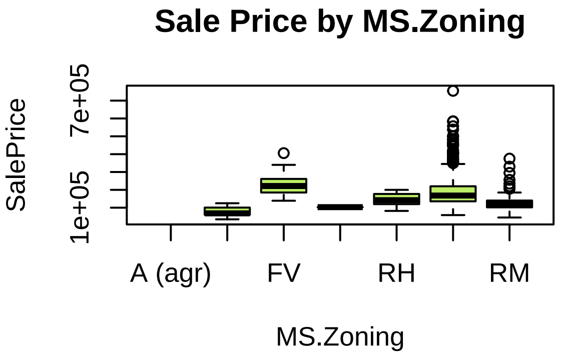
Removing insig. correlation cont. var

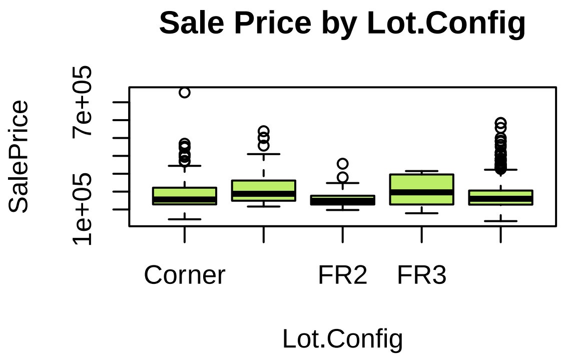
```r
library('psych')
corr_p = c()
remove_p = c()
for (i in cont_index[1:length(cont_index)-1]) {
  corr_p = corr.test(trainSale, trainData[, i], method = "pearson", alpha = 0.05)
  if (corr_p$p.adj > 0.05) {
    remove_p = c(remove_p, i)
  }
}

trainData = trainData[-remove_p]
```

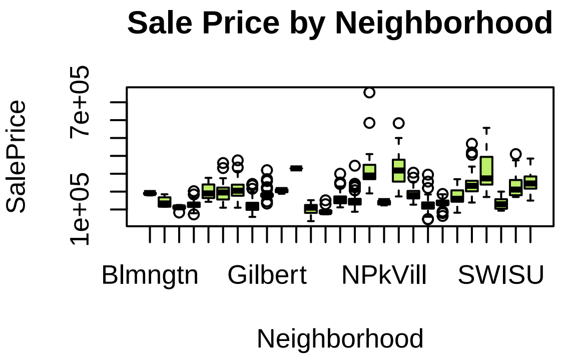
##Evaluating spread for cat. features

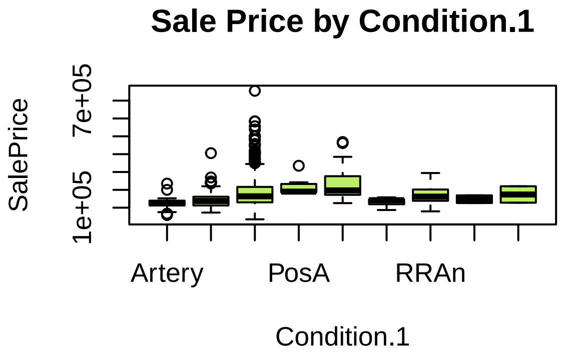
```r
factors = c()
for(j in 1:ncol(trainData)){
  if(is.factor(trainData[,j])) {
    factors = c(factors, j)
  }
}


par(mfrow=c(2,2))
for (j in factors) {
  boxplot(SalePrice ~ trainData[, j], data = trainData,
          main = paste("Sale Price by", names(trainData)[j]),
          xlab = names(trainData)[j],
          col = "darkolivegreen2")
}
```


Sale Price by MS.SubClass

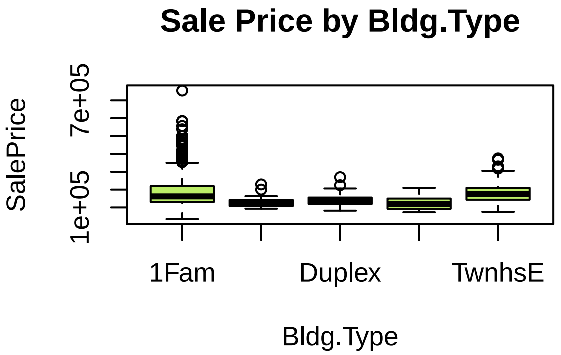

Sale Price by MS.Zoning


Sale Price by Lot.Config

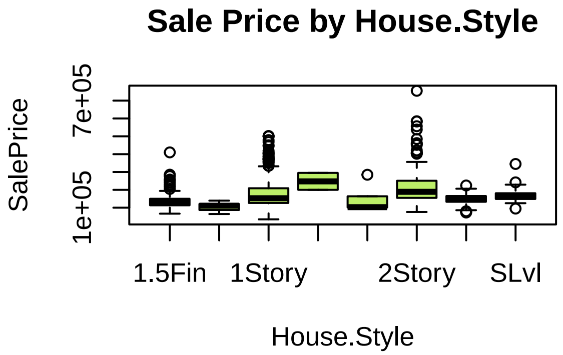

Sale Price by Neighborhood



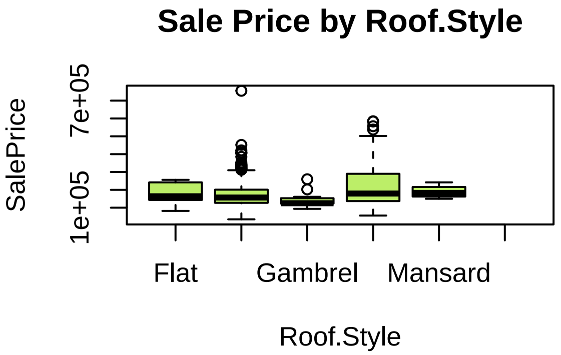
Sale Price by Condition.1



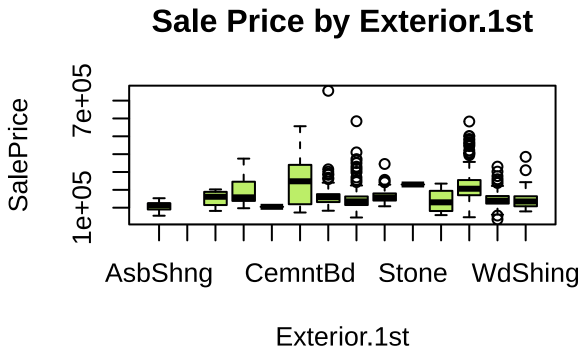
Sale Price by Bldg.Type

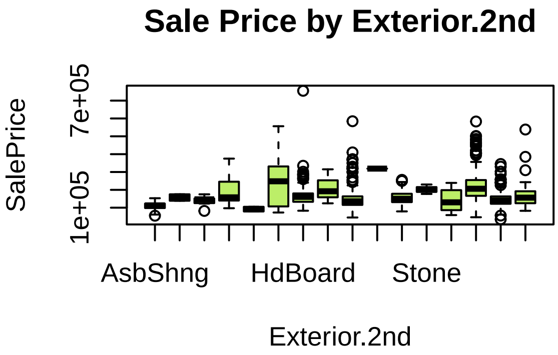


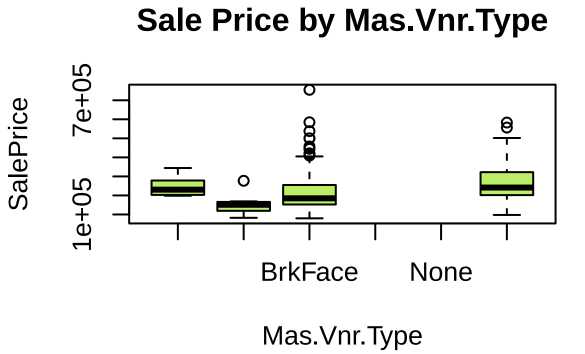
Sale Price by House.Style

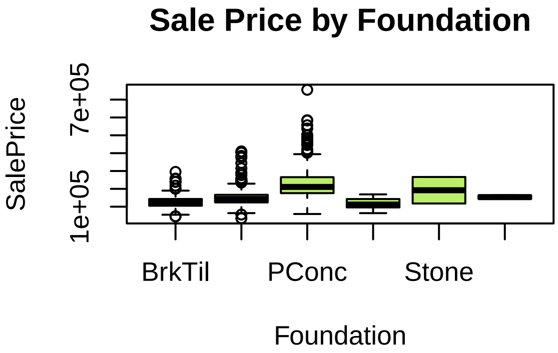


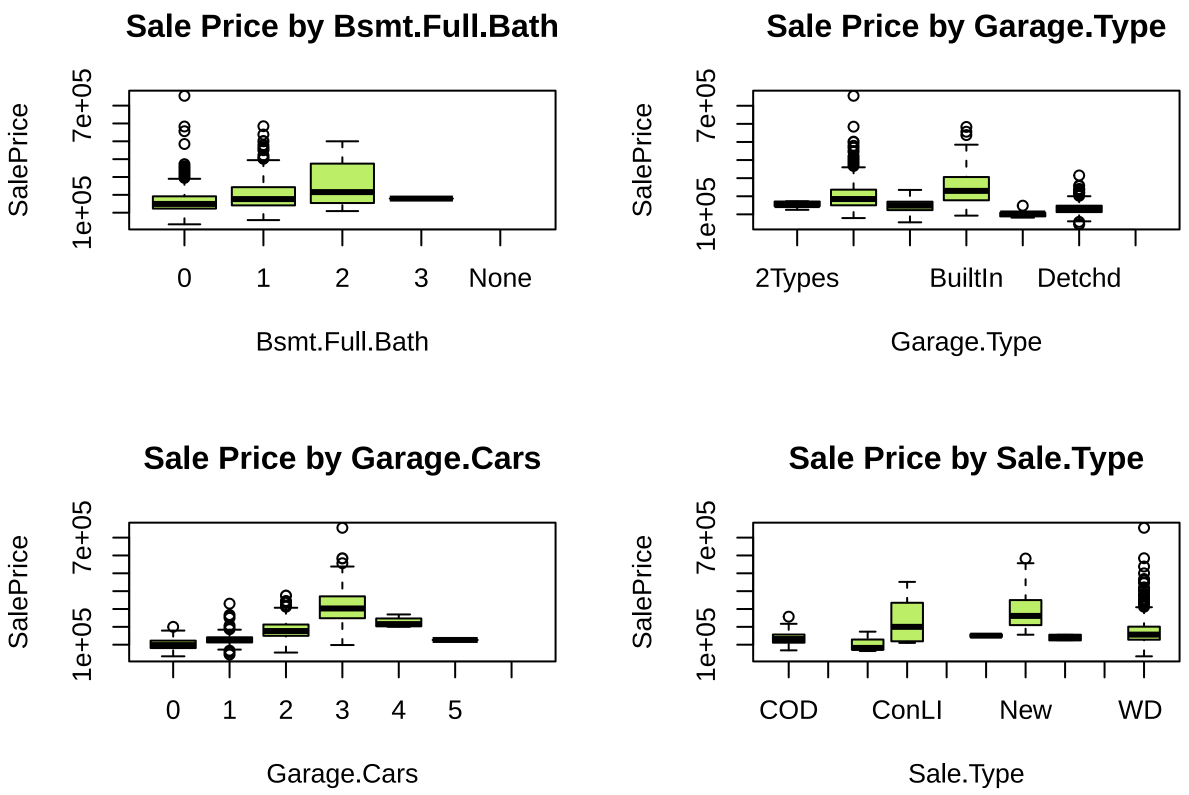
Sale Price by Roof.Style


Sale Price by Exterior.1st
SalePrice
1e+05
7e+05
AsbShng
CemntBd
Stone
WdShing
Exterior.1st

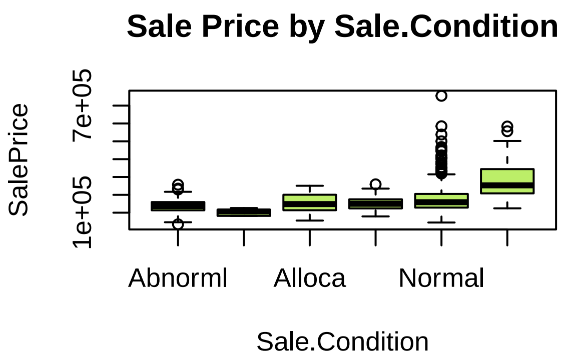

Sale Price by Exterior.2nd
SalePrice
1e+05
7e+05
AsbShng
HdBoard
Stone
Exterior.2nd


Sale Price by Mas.Vnr.Type
SalePrice
1e+05
7e+05
BrkFace
None
Mas.Vnr.Type


Sale Price by Foundation
SalePrice
1e+05
7e+05
BrkTil
PConc
Stone
Foundation

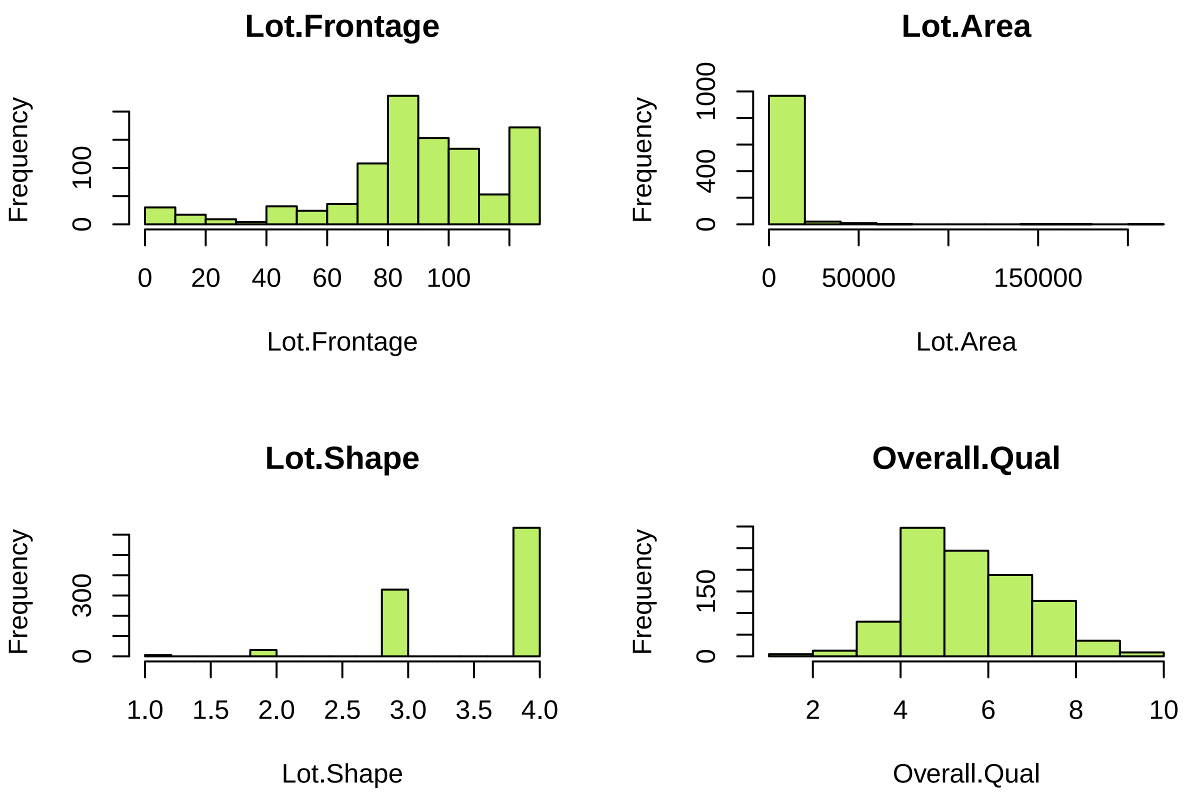


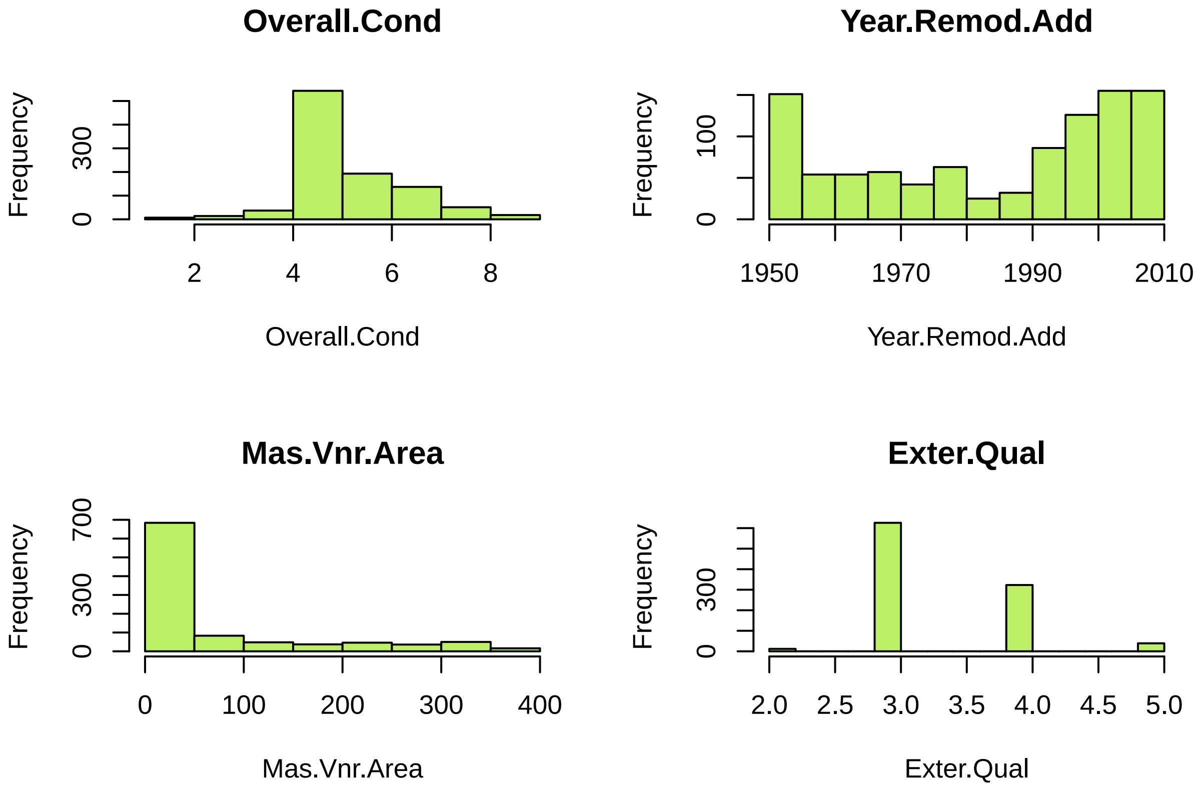
```
remove_cat = c(6, 7, 8, 13, 17)
trainData = trainData[-remove_cat]
```

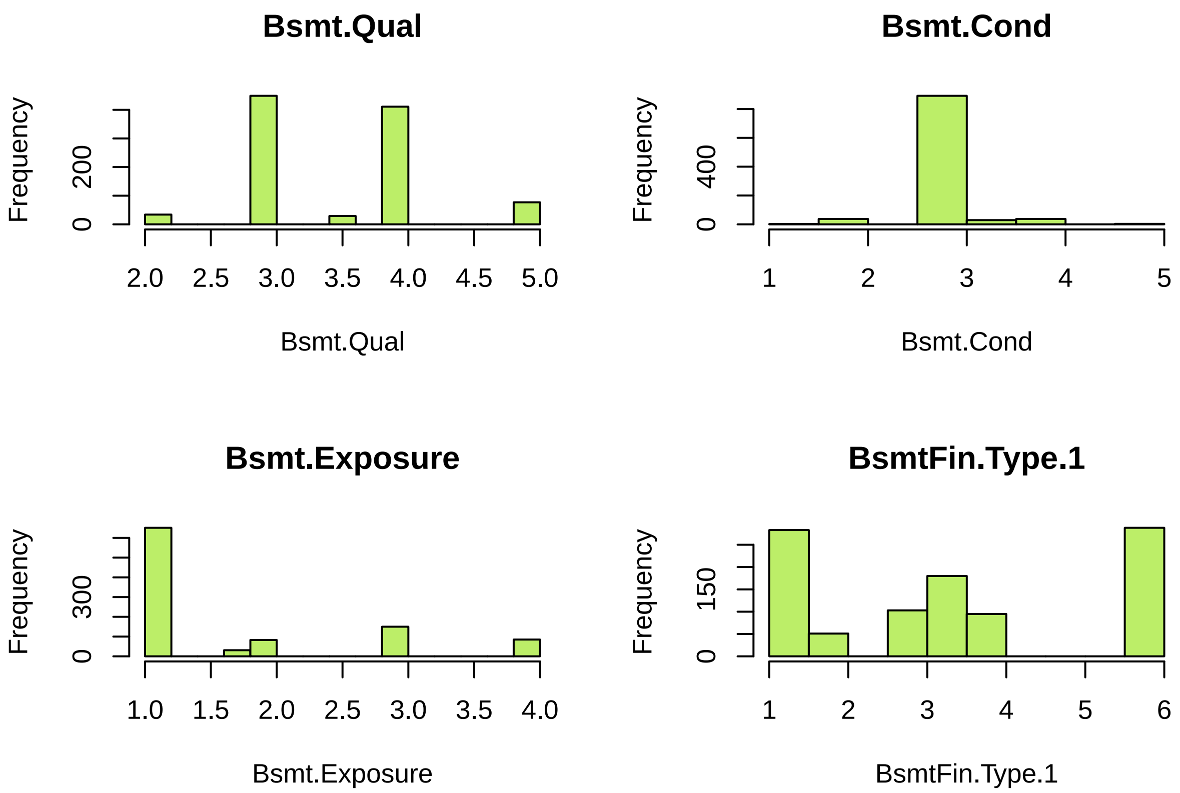


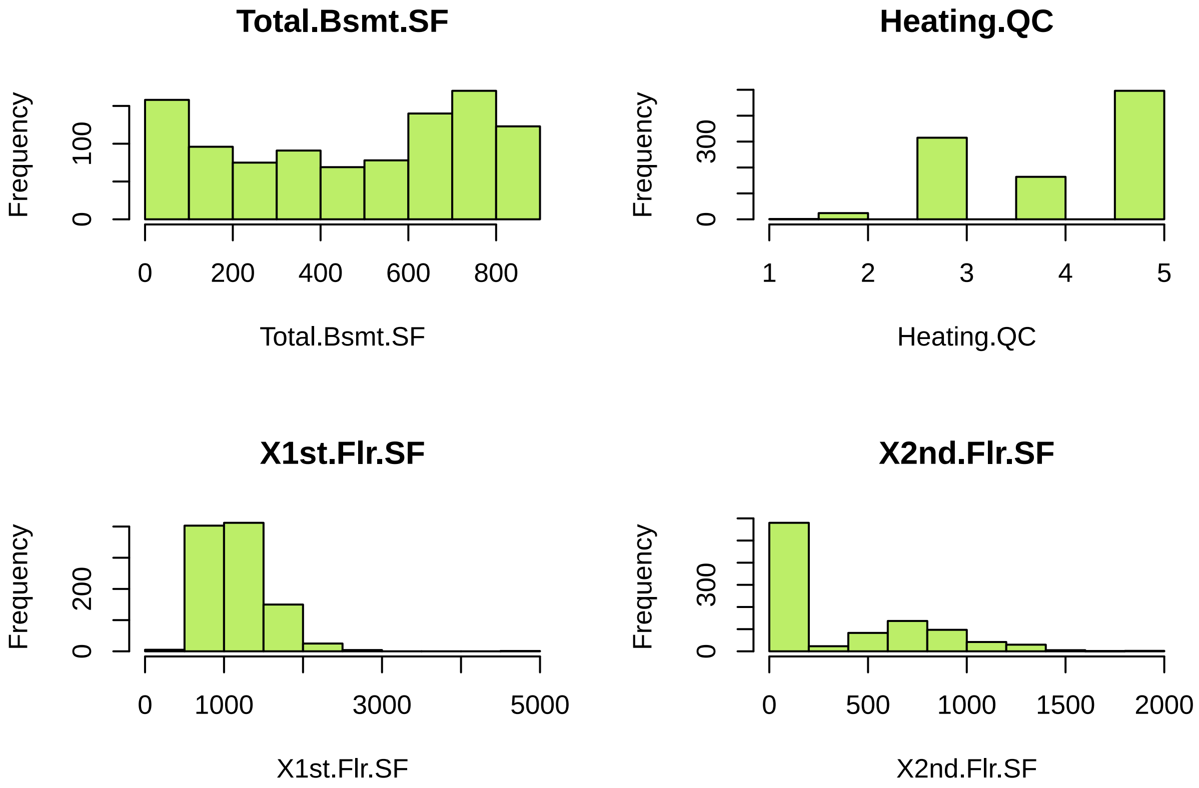
##Evaluating spread for cont. features

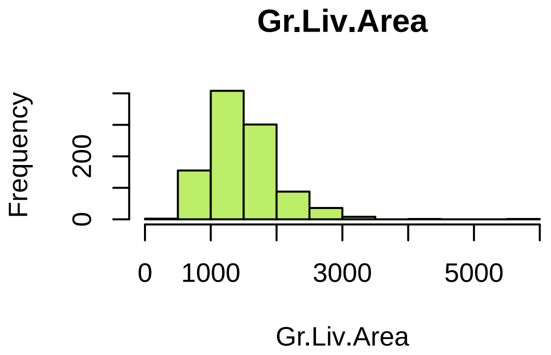
```
cont = c()
for(j in 1:ncol(trainData)){
  if(is.numeric(trainData[,j])) {
    cont = c(cont, j)
  }
}

par(mfrow = c(2,2))
names = colnames(trainData)
for (i in cont) {
  hist(trainData[,i], main = names[i],
       xlab = names[i],
       col = "darkolivegreen2")
}
```

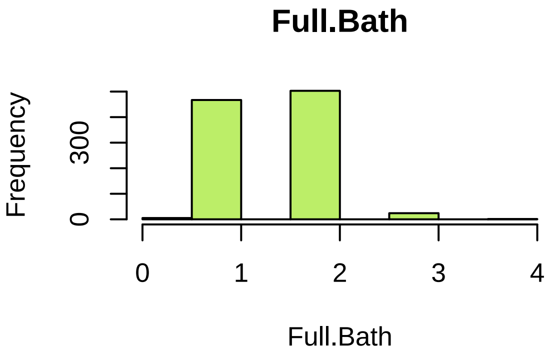

Lot.Frontage
Frequency
0
100
0
20
40
60
80
100
Lot.Frontage
Lot.Area
Frequency
0
400
1000
0
50000
150000
Lot.Area
Lot.Shape
Frequency
0
300
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Lot.Shape
Overall.Qual
Frequency
0
150
2
4
6
8
10
Overall.Qual

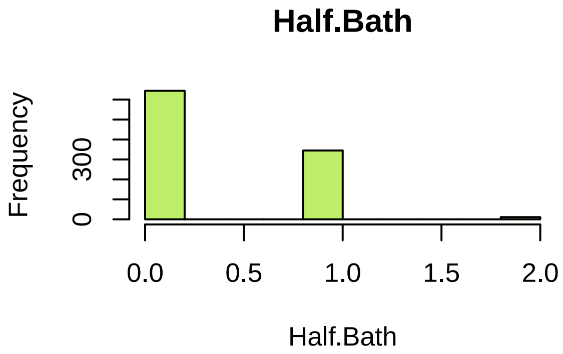

Overall.Cond
Frequency
0
300
2
4
6
8
Overall.Cond
Year.Remod.Add
Frequency
0
100
1950
1970
1990
2010
Year.Remod.Add
Mas.Vnr.Area
Frequency
0
300
700
0
100
200
300
400
Mas.Vnr.Area
Exter.Qual
Frequency
0
300
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Exter.Qual

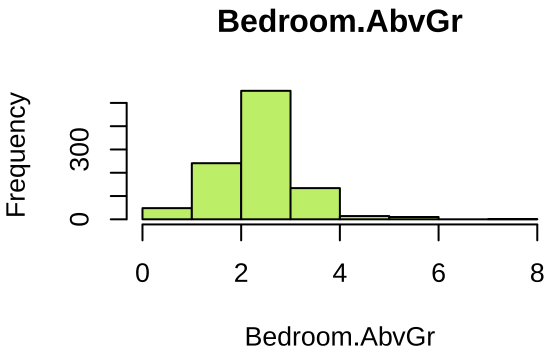

Bsmt.Qual
Frequency
0
200
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Bsmt.Qual
Bsmt.Cond
Frequency
0
400
1
2
3
4
5
Bsmt.Cond
Bsmt.Exposure
Frequency
0
300
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Bsmt.Exposure
BsmtFin.Type.1
Frequency
0
150
1
2
3
4
5
6
BsmtFin.Type.1

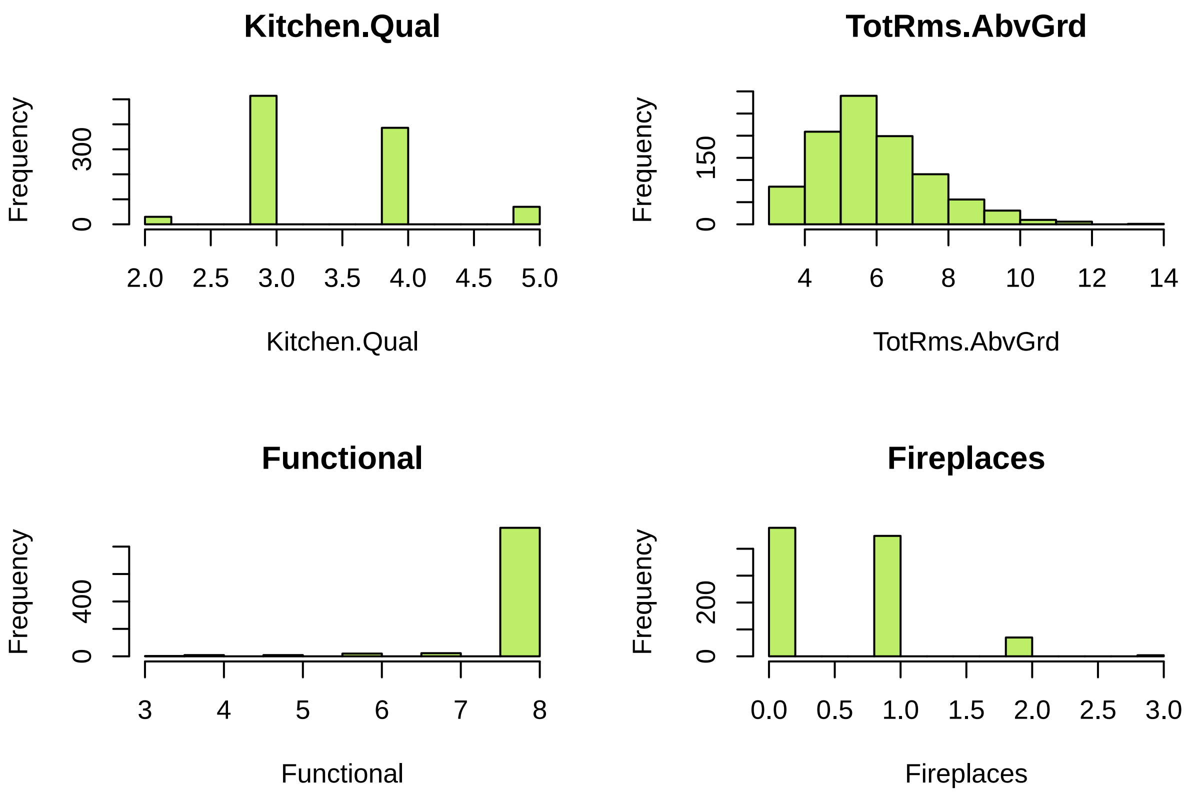

Total.Bsmt.SF
Frequency
0
100
0
200
400
600
800
Total.Bsmt.SF
Heating.QC
Frequency
0
300
1
2
3
4
5
Heating.QC
X1st.Flr.SF
Frequency
0
200
0
1000
3000
5000
X1st.Flr.SF
X2nd.Flr.SF
Frequency
0
300
0
500
1000
1500
2000
X2nd.Flr.SF

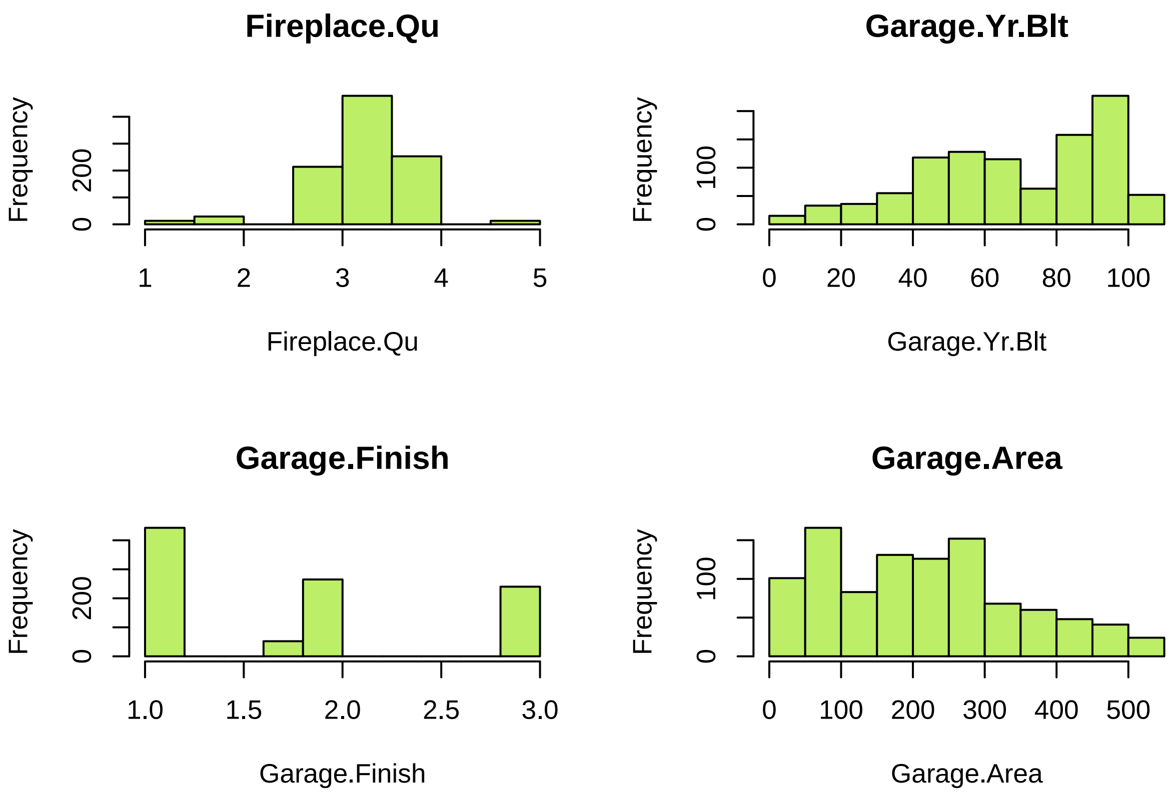

Gr.Liv.Area
Frequency
0
200
0
1000
3000
5000
Gr.Liv.Area


Full.Bath
Frequency
0
300
0
1
2
3
4
Full.Bath

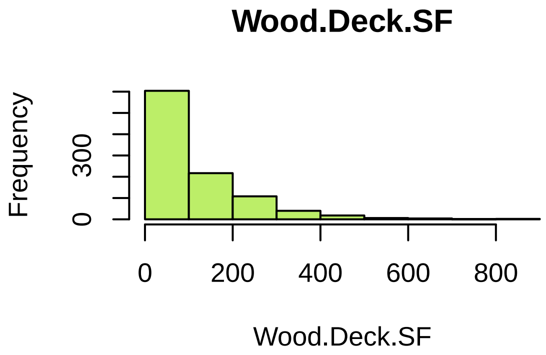

Half.Bath
Frequency
0
300
0.0
0.5
1.0
1.5
2.0
Half.Bath

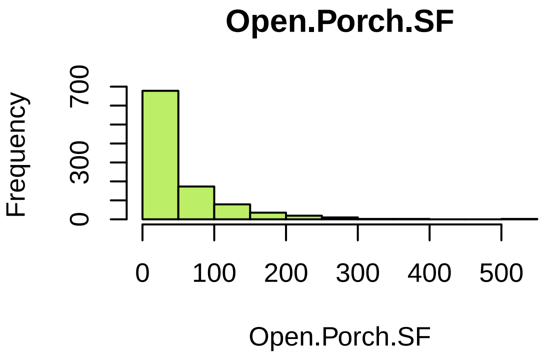

Bedroom.AbvGr
Frequency
0
300
0
2
4
6
8
Bedroom.AbvGr

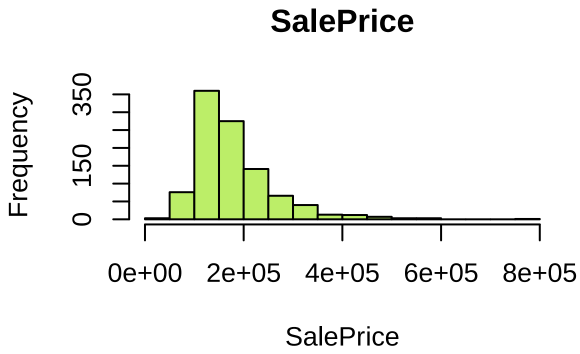

Kitchen.Qual
Frequency
0
300
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Kitchen.Qual
TotRms.AbvGrd
Frequency
0
150
4
6
8
10
12
14
TotRms.AbvGrd
Functional
Frequency
0
400
3
4
5
6
7
8
Functional
Fireplaces
Frequency
0
200
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Fireplaces



```
remove_con = c(2, 7, 8, 9, 10, 11, 16, 19, 23, 25, 26, 27, 28)
trainData = trainData[-remove_con]
```







##Initial linear model

```
initial_lm <- lm(SalePrice ~ ., data = trainData)
summary(initial_lm)
```

```
##
## Call:
## lm(formula = SalePrice ~ ., data = trainData)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -192723  -22679   -1671   20245  262480
##
## Coefficients: (3 not defined because of singularities)
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -1.253e+05  8.929e+04  -1.403 0.161591
## MS.SubClass30         -3.726e+04  4.669e+04  -0.798 0.425511
## MS.SubClass50         -1.915e+03  1.977e+04  -0.097 0.922898
## MS.SubClass60         -4.209e+04  1.573e+04  -2.676 0.007859 **
## MS.SubClass75          6.490e+03  4.733e+04   0.137 0.891020
## MS.SubClass80         -1.439e+04  1.170e+04  -1.229 0.219947
## MS.SubClass85         -2.514e+04  1.825e+04  -1.377 0.169488
## MS.SubClass90         -3.883e+04  2.276e+04  -1.706 0.089013 .
## MS.SubClass120        -4.955e+03  1.156e+04  -0.429 0.668570
## MS.SubClass160        -5.554e+04  2.770e+04  -2.005 0.045867 *
## Lot.Frontage          -3.538e+02  8.288e+01  -4.269 2.61e-05 ***
## Lot.Area               8.854e-02  3.115e-01   0.284 0.776412    
## Lot.Shape             -3.727e+03  4.490e+03  -0.830 0.407101    
## Bldg.TypeDuplex               NA         NA      NA       NA    
## Bldg.TypeTwnhs        -1.170e+04  2.436e+04  -0.480 0.631416    
## Bldg.TypeTwnhsE               NA         NA      NA       NA    
## Exterior.1stCemntBd    3.929e+04  5.109e+04   0.769 0.442400    
## Exterior.1stHdBoard   -5.111e+03  4.891e+04  -0.105 0.916841    
## Exterior.1stMetalSd    4.879e+03  4.926e+04   0.099 0.921174    
## Exterior.1stPlywood   -2.334e+04  4.958e+04  -0.471 0.638131    
## Exterior.1stStucco    -1.088e+05  5.606e+04  -1.941 0.053192 .  
## Exterior.1stVinylSd   -1.513e+04  4.954e+04  -0.305 0.760226    
## Exterior.1stWd Sdng   -1.822e+04  4.811e+04  -0.379 0.705191    
## Exterior.1stWdShing   -3.862e+04  5.694e+04  -0.678 0.498112    
## Mas.Vnr.TypeBrkCmn     2.518e+02  2.559e+04   0.010 0.992153    
## Mas.Vnr.TypeBrkFace   -2.057e+03  1.846e+04  -0.111 0.911318    
## Mas.Vnr.TypeStone     -1.056e+03  1.879e+04  -0.056 0.955213    
## Mas.Vnr.Area           2.741e+01  2.372e+01   1.156 0.248597    
## Exter.Qual             2.088e+04  7.214e+03   2.895 0.004066 ** 
## Bsmt.Qual              2.649e+04  6.730e+03   3.937 0.000102 ***
## Bsmt.Cond              3.953e+03  8.818e+03   0.448 0.654304    
## BsmtFin.Type.1         4.092e+03  1.374e+03   2.977 0.003139 ** 
## Total.Bsmt.SF         -4.415e+00  1.014e+01  -0.435 0.663538    
## Heating.QC             3.697e+02  4.029e+03   0.092 0.926962    
## X2nd.Flr.SF            3.972e+01  1.624e+01   2.447 0.014976 *  
## Bedroom.AbvGr         -3.333e+03  5.041e+03  -0.661 0.508963    
## Kitchen.Qual           2.129e+04  6.209e+03   3.429 0.000689 ***
## TotRms.AbvGrd          1.251e+04  3.031e+03   4.127 4.74e-05 ***
## Functional             4.063e+03  5.284e+03   0.769 0.442493    
## Fireplaces             1.505e+04  4.965e+03   3.030 0.002652 ** 
## Fireplace.Qu          -2.348e+02  4.619e+03  -0.051 0.959498    
## Garage.TypeAttchd     -1.496e+04  3.076e+04  -0.486 0.627086    
## Garage.TypeBasment    -1.966e+04  3.574e+04  -0.550 0.582620    
## Garage.TypeBuiltIn    -1.545e+04  3.268e+04  -0.473 0.636624    
## Garage.TypeCarPort     1.871e+04  5.387e+04   0.347 0.728579    
## Garage.TypeDetchd     -1.404e+04  3.060e+04  -0.459 0.646668    
## Garage.Yr.Blt         -4.770e+02  3.323e+02  -1.436 0.152134    
## Garage.Finish          1.503e+03  4.206e+03   0.357 0.721162    
## Garage.Cars2           1.077e+04  9.813e+03   1.097 0.273433    
## Garage.Cars3           5.587e+04  1.489e+04   3.751 0.000210 ***
## Garage.Cars4           2.464e+04  3.801e+04   0.648 0.517311    
## Garage.Area            1.539e+01  3.046e+01   0.505 0.613782    
## Wood.Deck.SF           6.913e+01  2.033e+01   3.400 0.000762 ***
## Open.Porch.SF          5.661e+01  4.830e+01   1.172 0.242102    
## Sale.TypeConLD        -7.291e+04  6.283e+04  -1.160 0.246808    
## Sale.TypeConLI        -1.444e+04  3.827e+04  -0.377 0.706304    
## Sale.TypeCWD          -2.151e+04  3.724e+04  -0.578 0.563932    
## Sale.TypeNew           1.863e+04  1.945e+04   0.958 0.338977    
## Sale.TypeWD           -6.456e+03  1.845e+04  -0.350 0.726590    
## Sale.ConditionAlloca   2.459e+04  3.497e+04   0.703 0.482473    
## Sale.ConditionFamily  -1.003e+04  2.143e+04  -0.468 0.639947    
## Sale.ConditionNormal   1.540e+04  1.533e+04   1.005 0.315878    
## Sale.ConditionPartial         NA         NA      NA       NA    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

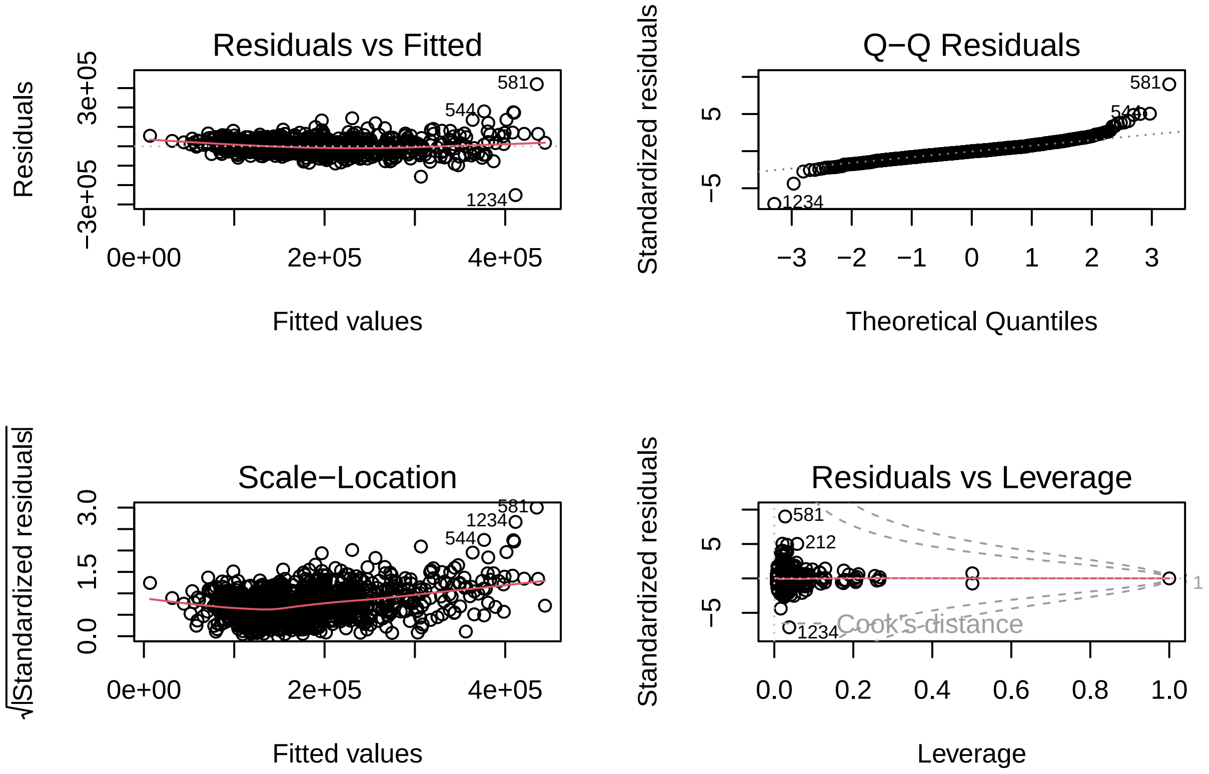
```
## 
## Residual standard error: 42890 on 308 degrees of freedom
##   (632 observations deleted due to missingness)
## Multiple R-squared:  0.829,  Adjusted R-squared:  0.7963 
## F-statistic: 25.31 on 59 and 308 DF,  p-value: < 2.2e-16
```

##Initial linear model

```
newTrainData = subset(trainData, select = c(MS.SubClass, Lot.Frontage, Exter.Qual, Bsmt.Qual, BsmtFin.Ty

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

```
## 
## Call:
## lm(formula = SalePrice ~ ., data = newTrainData)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -251427  -19740   -1370   16941  320101 
## 
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)    
## (Intercept)    -1.467e+05  1.270e+04 -11.548  < 2e-16 ***
## MS.SubClass30  -1.490e+04  6.333e+03  -2.353 0.018810 *  
## MS.SubClass40  -8.036e+00  2.575e+04   0.000 0.999751    
## MS.SubClass45  -1.303e+04  1.233e+04  -1.057 0.290800    
## MS.SubClass50  -9.199e+03  4.311e+03  -2.134 0.033097 *  
## MS.SubClass60  -8.822e+03  3.761e+03  -2.345 0.019203 *  
## MS.SubClass70  -1.383e+04  5.957e+03  -2.322 0.020458 *  
## MS.SubClass75   4.102e+03  1.503e+04   0.273 0.785019    
## MS.SubClass80  -1.132e+04  6.006e+03  -1.885 0.059791 .  
## MS.SubClass85  -1.348e+04  8.435e+03  -1.598 0.110459    
## MS.SubClass90  -4.089e+04  7.052e+03  -5.799 9.02e-09 ***
## MS.SubClass120 -5.340e+03  5.715e+03  -0.934 0.350379    
## MS.SubClass150 -6.663e+04  3.636e+04  -1.832 0.067184 .  
## MS.SubClass160 -3.412e+04  6.626e+03  -5.150 3.16e-07 ***
## MS.SubClass180 -5.105e+04  1.688e+04  -3.024 0.002563 ** 
## MS.SubClass190 -1.994e+04  9.843e+03  -2.026 0.043077 *  
## Lot.Frontage   -1.972e+02  4.458e+01  -4.424 1.08e-05 ***
## Exter.Qual      2.491e+04  3.221e+03   7.734 2.60e-14 ***
## Bsmt.Qual       2.227e+04  2.695e+03   8.264 4.58e-16 ***
## BsmtFin.Type.1  3.512e+03  6.823e+02   5.148 3.19e-07 ***
## Fireplaces      1.972e+04  2.054e+03   9.600  < 2e-16 ***
## Kitchen.Qual    1.656e+04  2.535e+03   6.530 1.06e-10 ***
## TotRms.AbvGrd   1.268e+04  1.056e+03  12.012  < 2e-16 ***
## Garage.Cars1    1.367e+04  5.642e+03   2.422 0.015612 *  
## Garage.Cars2    1.801e+04  5.699e+03   3.160 0.001628 ** 
## Garage.Cars3    7.089e+04  7.148e+03   9.918  < 2e-16 ***
## Garage.Cars4    6.610e+04  1.975e+04   3.346 0.000851 ***
## Garage.Cars5    4.048e+04  3.677e+04   1.101 0.271247    
## Wood.Deck.SF    3.805e+01  9.467e+00   4.020 6.28e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 36060 on 971 degrees of freedom
## Multiple R-squared:  0.8019, Adjusted R-squared:  0.7962
## F-statistic: 140.4 on 28 and 971 DF,  p-value: < 2.2e-16
```

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



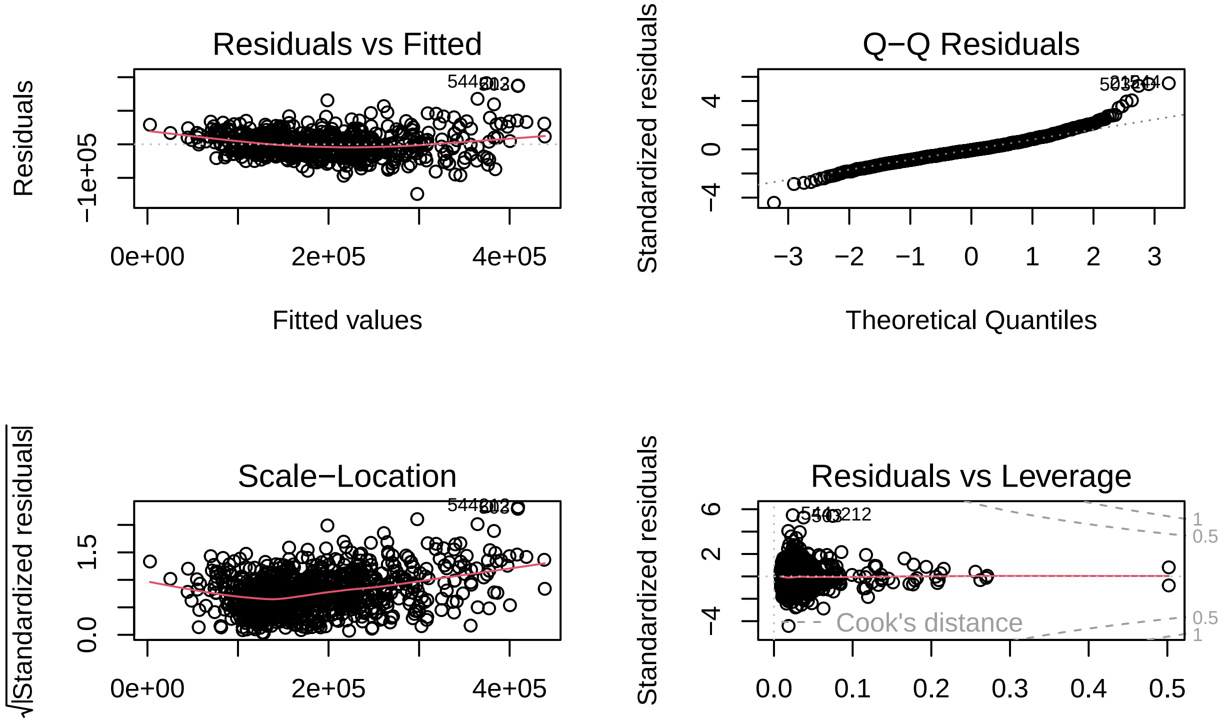
```
#remove 1234 because it's all na
train_Index = train_Index[-which(train_Index == 1234 | train_Index == 581)]
newTrainData = subset(data[train_Index, ], select = c(MS.SubClass, Lot.Frontage, Exter.Qual, Bsmt.Qual,
```

```
lm2 <- lm(SalePrice ~ ., data = newTrainData)
summary(lm2)
```

```
##
## Call:
## lm(formula = SalePrice ~ ., data = newTrainData)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -147945  -19210   -2259   17765  182202
##
```

```
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -166876.25   13331.76 -12.517  < 2e-16 ***
## MS.SubClass30   -10785.68    6548.62  -1.647 0.099953 .
## MS.SubClass40      450.75   24133.69   0.019 0.985103
## MS.SubClass45   -12725.84   12248.22  -1.039 0.299125
## MS.SubClass50    -8129.60    4427.76  -1.836 0.066726 .
## MS.SubClass60   -14499.67    3869.82  -3.747 0.000192 ***
## MS.SubClass70   -12046.15    6002.22  -2.007 0.045096 *
## MS.SubClass75     4254.39   14128.45   0.301 0.763400
## MS.SubClass80   -19404.27    6579.35  -2.949 0.003279 **
## MS.SubClass85   -11986.17    9482.22  -1.264 0.206579
## MS.SubClass90   -34581.65    8182.21  -4.226 2.65e-05 ***
## MS.SubClass120   -6960.39    5662.84  -1.229 0.219388
## MS.SubClass150  -73382.01   34069.30  -2.154 0.031550 *
## MS.SubClass160  -31735.43    7049.03  -4.502 7.74e-06 ***
## MS.SubClass180  -54226.86   15980.09  -3.393 0.000725 ***
## MS.SubClass190  -26719.84   11410.87  -2.342 0.019449 *
## Lot.Frontage      -198.84      45.35  -4.385 1.32e-05 ***
## Exter.Qual       25137.02    3390.17   7.415 3.15e-13 ***
## Bsmt.Qual        28235.44    2863.07   9.862  < 2e-16 ***
## BsmtFin.Type.1    3352.81     649.95   5.159 3.15e-07 ***
## Fireplaces       18534.36    2158.77   8.586  < 2e-16 ***
## Kitchen.Qual     15308.70    2648.63   5.780 1.08e-08 ***
## TotRms.AbvGrd    13435.06    1106.41  12.143  < 2e-16 ***
## Garage.Cars1     16114.20    5933.89   2.716 0.006760 **
## Garage.Cars2     18118.23    6030.50   3.004 0.002745 **
## Garage.Cars3     63342.49    7382.01   8.581  < 2e-16 ***
## Garage.Cars4     53568.05   19356.28   2.767 0.005782 **
## Garage.Cars5     38355.25   34503.54   1.112 0.266635
## Wood.Deck.SF        41.05      10.34   3.971 7.83e-05 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 33730 on 789 degrees of freedom
##   (180 observations deleted due to missingness)
## Multiple R-squared:  0.8315, Adjusted R-squared:  0.8255
## F-statistic: 139.1 on 28 and 789 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(lm2)
```



```r
library(car)
vif = vif(lm2)
vif
```

```
##                    GVIF Df GVIF^(1/(2*Df))
## MS.SubClass    5.860107 15        1.060710
## Lot.Frontage   1.125793  1        1.061034
## Exter.Qual     2.924727  1        1.710183
## Bsmt.Qual      2.959680  1        1.720372
## BsmtFin.Type.1 1.379340  1        1.174453
## Fireplaces     1.293933  1        1.137512
## Kitchen.Qual   2.359997  1        1.536228
## TotRms.AbvGrd  2.300215  1        1.516646
## Garage.Cars    3.246142  5        1.124959
## Wood.Deck.SF   1.169616  1        1.081488
```

##testing the linear model on the test dataset

```r
#making sure newTestData is formatted correctly, replacing nas with means of column
newTestData = subset(testData, select = c(MS.SubClass, Lot.Frontage, Exter.Qual, Bsmt.Qual, BsmtFin.Typ

cont_index_test <- c()
for (i in 1:ncol(newTestData)) {
  if (!is.factor(newTestData[, i])) {
    cont_index_test = c(cont_index_test, i)
  }
}

for (y in cont_index_test) {
  for (x in 1:length(newTestData[, y])) {
    if (is.na(newTestData[x, y])){
      newTestData[x, y] = mean(newTestData[, y], na.rm = TRUE)
    }
  }
}


predictions <- predict(lm2, newdata = newTestData)

#calculating R^2
actuals <- testData$SalePrice
m_actuals <- mean(actuals)
ss_total <- sum((actuals - m_actuals)^2)
ss_residual <- sum((actuals - predictions)^2, na.rm = TRUE)
rsquared <- 1 - (ss_residual / ss_total)
rsquared
```

```
## [1] 0.7773455
```

##confint and prediction interval for linear model

```
confint(lm2, level = 0.95)
```

```
##                        2.5 %        97.5 %
## (Intercept)    -193046.1706 -140706.33134
## MS.SubClass30   -23640.4690    2069.10298
## MS.SubClass40   -46923.0794   47824.57805
## MS.SubClass45   -36768.7973   11317.10971
## MS.SubClass50   -16821.1749     561.97708
## MS.SubClass60   -22096.0267   -6903.31679
## MS.SubClass70   -23828.3577    -263.95030
## MS.SubClass75   -23479.3940   31988.18235
## MS.SubClass80   -32319.3582   -6489.17461
## MS.SubClass85   -30599.5203    6627.18821
## MS.SubClass90   -50643.1182  -18520.17463
## MS.SubClass120  -18076.3968    4155.61486
## MS.SubClass150 -140259.2025   -6504.82035
## MS.SubClass160  -45572.4891  -17898.36578
## MS.SubClass180  -85595.3716  -22858.34490
## MS.SubClass190  -49119.0855   -4320.59303
## Lot.Frontage      -287.8580    -109.82352
## Exter.Qual       18482.1956   31791.83978
## Bsmt.Qual        22615.3024   33855.56971
## BsmtFin.Type.1    2076.9846    4628.63724
## Fireplaces       14296.7463   22771.98220
## Kitchen.Qual     10109.5041   20507.89410
## TotRms.AbvGrd    11263.1960   15606.92400
## Garage.Cars1      4466.1230   27762.27291
## Garage.Cars2      6280.4988   29955.95322
## Garage.Cars3     48851.7957   77833.18918
## Garage.Cars4     15572.1540   91563.94235
## Garage.Cars5    -29374.3468  106084.84308
## Wood.Deck.SF        20.7542      61.34048
```

```
predInt1 = predict(lm2, newdata = newTestData, interval = "predict")
```

```
log_lm <- lm(log(SalePrice) ~ ., data = trainData)
summary(log_lm)
```

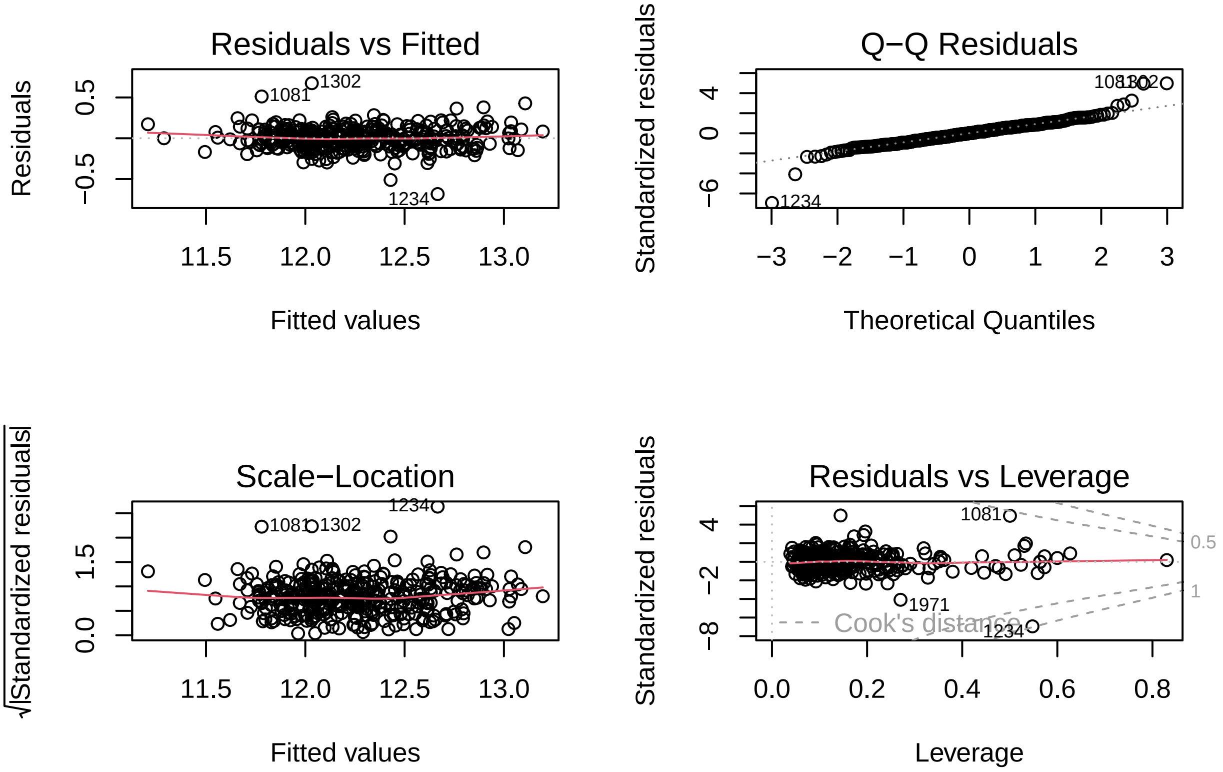
```
##
## Call:
## lm(formula = log(SalePrice) ~ ., data = trainData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.68342 -0.08074  0.00000  0.08189  0.67403
##
## Coefficients: (3 not defined because of singularities)
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            1.036e+01  3.046e-01  34.013  < 2e-16 ***
## MS.SubClass30         -4.374e-01  1.593e-01  -2.747 0.006377 **
## MS.SubClass50          1.054e-01  6.744e-02   1.564 0.118938
## MS.SubClass60         -9.565e-02  5.366e-02  -1.783 0.075636 .
## MS.SubClass75          1.313e-01  1.615e-01   0.813 0.416882
## MS.SubClass80         -2.146e-02  3.993e-02  -0.537 0.591377
## MS.SubClass85         -9.403e-02  6.227e-02  -1.510 0.132024
## MS.SubClass90         -2.078e-01  7.764e-02  -2.676 0.007847 **
## MS.SubClass120        -2.718e-02  3.945e-02  -0.689 0.491297
## MS.SubClass160        -1.424e-01  9.450e-02  -1.506 0.132974
## Lot.Frontage          -6.730e-04  2.827e-04  -2.381 0.017895 *
## Lot.Area               8.736e-08  1.063e-06   0.082 0.934528
## Lot.Shape             -1.669e-02  1.532e-02  -1.090 0.276781
## Bldg.TypeDuplex               NA         NA      NA       NA
## Bldg.TypeTwnhs        -6.898e-02  8.310e-02  -0.830 0.407120
## Bldg.TypeTwnhsE               NA         NA      NA       NA
## Exterior.1stCemntBd    2.334e-01  1.743e-01   1.339 0.181451
## Exterior.1stHdBoard    5.663e-02  1.669e-01   0.339 0.734553
## Exterior.1stMetalSd    1.369e-01  1.681e-01   0.815 0.415796
## Exterior.1stPlywood    2.160e-02  1.691e-01   0.128 0.898466
## Exterior.1stStucco    -3.506e-01  1.912e-01  -1.833 0.067695 .
## Exterior.1stVinylSd    6.152e-02  1.690e-01   0.364 0.716090
## Exterior.1stWd Sdng    5.721e-02  1.641e-01   0.349 0.727638
## Exterior.1stWdShing   -6.957e-03  1.942e-01  -0.036 0.971449
## Mas.Vnr.TypeBrkCmn    -2.221e-02  8.728e-02  -0.255 0.799278
## Mas.Vnr.TypeBrkFace   -1.678e-02  6.296e-02  -0.266 0.790048
## Mas.Vnr.TypeStone     -1.865e-02  6.410e-02  -0.291 0.771261
## Mas.Vnr.Area           9.597e-05  8.090e-05   1.186 0.236425
## Exter.Qual             7.034e-02  2.461e-02   2.858 0.004548 **
## Bsmt.Qual              9.420e-02  2.296e-02   4.103 5.22e-05 ***
## Bsmt.Cond              2.180e-02  3.008e-02   0.725 0.469089
## BsmtFin.Type.1         1.254e-02  4.688e-03   2.675 0.007875 **
## Total.Bsmt.SF         -4.224e-05  3.459e-05  -1.221 0.222942
## Heating.QC             1.212e-02  1.374e-02   0.881 0.378765
## X2nd.Flr.SF            8.060e-05  5.538e-05   1.455 0.146627
## Bedroom.AbvGr          8.918e-03  1.720e-02   0.519 0.604374
## Kitchen.Qual           7.772e-02  2.118e-02   3.670 0.000286 ***
## TotRms.AbvGrd          4.278e-02  1.034e-02   4.137 4.54e-05 ***
## Functional             3.032e-02  1.802e-02   1.682 0.093554 .
## Fireplaces             6.895e-02  1.694e-02   4.071 5.97e-05 ***
## Fireplace.Qu          -6.616e-03  1.576e-02  -0.420 0.674867
## Garage.TypeAttchd     -6.085e-03  1.049e-01  -0.058 0.953789
## Garage.TypeBasment     6.260e-02  1.219e-01   0.513 0.607978
## Garage.TypeBuiltIn    -2.043e-02  1.115e-01  -0.183 0.854747
## Garage.TypeCarPort     1.161e-01  1.838e-01   0.632 0.527974
## Garage.TypeDetchd     -5.614e-02  1.044e-01  -0.538 0.591139
## Garage.Yr.Blt          1.944e-04  1.134e-03   0.171 0.863965
## Garage.Finish          5.550e-03  1.435e-02   0.387 0.699169
## Garage.Cars2           9.463e-02  3.347e-02   2.827 0.005007 **
## Garage.Cars3           2.174e-01  5.081e-02   4.278 2.52e-05 ***
## Garage.Cars4           2.403e-01  1.297e-01   1.853 0.064810 .
## Garage.Area            1.876e-04  1.039e-04   1.806 0.071956 .
## Wood.Deck.SF           2.307e-04  6.935e-05   3.327 0.000986 ***
## Open.Porch.SF          2.906e-04  1.648e-04   1.764 0.078778 .
## Sale.TypeConLD        -1.337e-01  2.143e-01  -0.624 0.533246
## Sale.TypeConLI        -1.134e-01  1.306e-01  -0.868 0.385945
## Sale.TypeCWD          -9.540e-02  1.270e-01  -0.751 0.453236
## Sale.TypeNew           9.635e-02  6.635e-02   1.452 0.147477
## Sale.TypeWD           -3.445e-02  6.293e-02  -0.547 0.584473
## Sale.ConditionAlloca   1.536e-01  1.193e-01   1.288 0.198854
## Sale.ConditionFamily  -6.625e-03  7.310e-02  -0.091 0.927854
## Sale.ConditionNormal   1.003e-01  5.230e-02   1.918 0.055980 .
## Sale.ConditionPartial         NA         NA      NA       NA
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.1463 on 308 degrees of freedom
##   (632 observations deleted due to missingness)
## Multiple R-squared:  0.8716, Adjusted R-squared:  0.847
## F-statistic: 35.43 on 59 and 308 DF,  p-value: < 2.2e-16
```

```
newTrainData = subset(data[train_Index, ], select = c(MS.SubClass, Exter.Qual, Bsmt.Qual, BsmtFin.Type.

log_lm2 <- lm(log(SalePrice) ~ ., data = newTrainData)
summary(log_lm2)
```

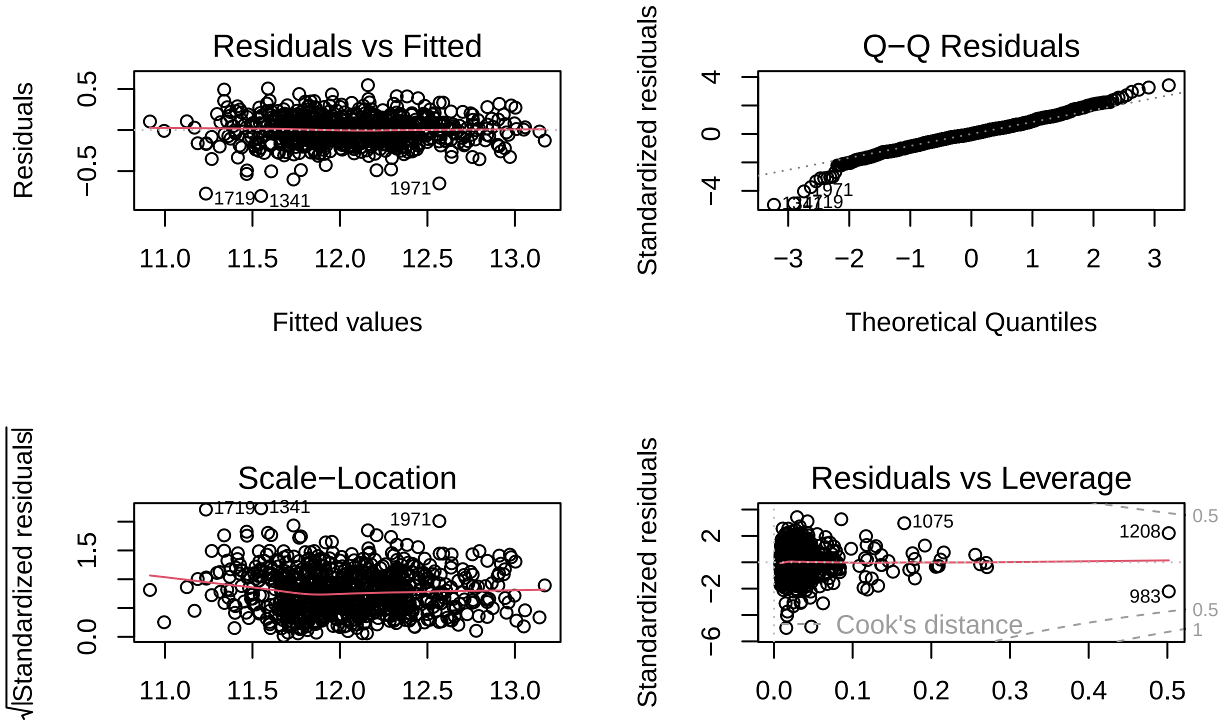
```
##
## Call:
## lm(formula = log(SalePrice) ~ ., data = newTrainData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.80204 -0.09078 -0.00121  0.09078  0.54686
##
## Coefficients:
```

```
##                   Estimate Std. Error t value Pr(>|t|)    
## (Intercept)      1.015e+01  5.991e-02 169.494  < 2e-16 ***
## MS.SubClass30   -2.249e-01  3.143e-02  -7.156 1.90e-12 ***
## MS.SubClass40   -6.305e-02  1.160e-01  -0.544 0.586793    
## MS.SubClass45   -1.288e-01  5.881e-02  -2.191 0.028767 *  
## MS.SubClass50   -6.047e-02  2.123e-02  -2.848 0.004513 ** 
## MS.SubClass60   -5.750e-02  1.862e-02  -3.088 0.002083 ** 
## MS.SubClass70   -4.261e-02  2.887e-02  -1.476 0.140284    
## MS.SubClass75    3.962e-02  6.794e-02   0.583 0.559960    
## MS.SubClass80   -8.094e-02  3.160e-02  -2.561 0.010620 *  
## MS.SubClass85   -5.212e-02  4.561e-02  -1.143 0.253455    
## MS.SubClass90   -1.390e-01  3.936e-02  -3.531 0.000438 ***
## MS.SubClass120 -1.608e-02  2.713e-02  -0.593 0.553606    
## MS.SubClass150 -3.257e-01  1.638e-01  -1.989 0.047047 *  
## MS.SubClass160 -1.752e-01  3.305e-02  -5.301 1.50e-07 ***
## MS.SubClass180 -4.110e-01  7.616e-02  -5.397 8.97e-08 ***
## MS.SubClass190 -1.169e-01  5.476e-02  -2.136 0.033002 *  
## Exter.Qual       1.324e-01  1.631e-02   8.122 1.76e-15 ***
## Bsmt.Qual        1.393e-01  1.377e-02  10.110  < 2e-16 ***
## BsmtFin.Type.1   1.510e-02  3.125e-03   4.831 1.63e-06 ***
## Fireplaces       1.024e-01  1.038e-02   9.867  < 2e-16 ***
## Kitchen.Qual     6.318e-02  1.273e-02   4.963 8.50e-07 ***
## TotRms.AbvGrd    6.360e-02  5.321e-03  11.952  < 2e-16 ***
## Garage.Cars1     1.710e-01  2.853e-02   5.993 3.12e-09 ***
## Garage.Cars2     2.341e-01  2.896e-02   8.081 2.40e-15 ***
## Garage.Cars3     3.422e-01  3.550e-02   9.638  < 2e-16 ***
## Garage.Cars4     3.990e-01  9.298e-02   4.291 2.00e-05 ***
## Garage.Cars5     4.655e-01  1.660e-01   2.804 0.005164 ** 
## Wood.Deck.SF     1.731e-04  4.973e-05   3.480 0.000528 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.1623 on 790 degrees of freedom
##   (180 observations deleted due to missingness)
## Multiple R-squared:  0.8438, Adjusted R-squared:  0.8384 
## F-statistic:   158 on 27 and 790 DF,  p-value: < 2.2e-16
```

```r
par(mfrow=c(2,2))
plot(log_lm)
```



```
par(mfrow=c(2,2))
plot(log_lm2)
```



```
MSE.log <- mean(log_lm2$residuals^2)
print(MSE.log)
```

```
## [1] 0.02542539
```

```
MSE.linear <- mean(lm2$residuals^2)
print(MSE.linear)
```

```
## [1] 1097099195
```

```
#testing
newTestData = subset(testData, select = c(MS.SubClass, Exter.Qual, Bsmt.Qual, BsmtFin.Type.1, Fireplace

cont_index_test <- c()
for (i in 1:ncol(newTestData)) {
  if (!is.factor(newTestData[, i])) {
    cont_index_test = c(cont_index_test, i)
  }
}

for (y in cont_index_test) {
  for (x in 1:length(newTestData[, y])) {
    if (is.na(newTestData[x, y])){
      newTestData[x, y] = mean(newTestData[, y], na.rm = TRUE)
    }
  }
}

predictions_log <- predict(log_lm2, newdata = newTestData)


#calculating R^2
actualsLG <- log(newTestData$SalePrice)
m_actualsLG <- mean(actualsLG)
ss_total <- sum((actualsLG - m_actualsLG)^2)
ss_residual <- sum((actualsLG - predictions_log)^2, na.rm = TRUE)
rsquared <- 1 - (ss_residual / ss_total)
rsquared
```

```
## [1] 0.7609686
```

```
predInt2= predict(log_lm2, newdata = newTestData, interval = "predict")
```

```
squared_lm <- lm((SalePrice)^2 ~ ., data = trainData)
summary(squared_lm)
```

```
##
## Call:
## lm(formula = (SalePrice)^2 ~ ., data = trainData)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -1.209e+11 -1.570e+10 -8.263e+08  1.185e+10  3.145e+11
##
## Coefficients: (3 not defined because of singularities)
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -7.166e+10  7.129e+10  -1.005  0.31562
## MS.SubClass30          -8.507e+09  3.728e+10  -0.228  0.81963
## MS.SubClass50          -2.233e+10  1.578e+10  -1.415  0.15809
## MS.SubClass60          -4.024e+10  1.256e+10  -3.204  0.00150 **
## MS.SubClass75          -2.330e+10  3.779e+10  -0.616  0.53803
## MS.SubClass80          -1.307e+10  9.345e+09  -1.398  0.16308
## MS.SubClass85          -1.529e+10  1.457e+10  -1.049  0.29485
## MS.SubClass90          -1.822e+10  1.817e+10  -1.003  0.31688
## MS.SubClass120         -3.935e+09  9.232e+09  -0.426  0.67026
## MS.SubClass160         -4.813e+10  2.212e+10  -2.176  0.03032 *
## Lot.Frontage           -3.386e+08  6.617e+07  -5.118 5.46e-07 ***
## Lot.Area                9.390e+04  2.487e+05   0.378  0.70601
## Lot.Shape              -2.250e+09  3.585e+09  -0.628  0.53075
## Bldg.TypeDuplex                NA         NA      NA       NA
## Bldg.TypeTwnhs         -8.366e+09  1.945e+10  -0.430  0.66739
## Bldg.TypeTwnhsE                NA         NA      NA       NA
## Exterior.1stCemntBd     1.087e+10  4.079e+10   0.266  0.79006
## Exterior.1stHdBoard    -1.144e+10  3.905e+10  -0.293  0.76984
## Exterior.1stMetalSd    -1.258e+10  3.933e+10  -0.320  0.74929
## Exterior.1stPlywood    -2.769e+10  3.958e+10  -0.700  0.48470
## Exterior.1stStucco    -7.786e+10  4.476e+10  -1.740  0.08293 .
## Exterior.1stVinylSd   -2.581e+10  3.955e+10  -0.653  0.51453
## Exterior.1stWd Sdng   -2.736e+10  3.841e+10  -0.712  0.47689
## Exterior.1stWdShing   -4.096e+10  4.546e+10  -0.901  0.36823
## Mas.Vnr.TypeBrkCmn     2.375e+09  2.043e+10   0.116  0.90753
## Mas.Vnr.TypeBrkFace   -7.423e+08  1.474e+10  -0.050  0.95986
## Mas.Vnr.TypeStone      1.881e+09  1.500e+10   0.125  0.90029
## Mas.Vnr.Area           9.695e+06  1.893e+07   0.512  0.60899
## Exter.Qual             1.497e+10  5.760e+09   2.600  0.00978 **
## Bsmt.Qual              1.506e+10  5.373e+09   2.802  0.00540 **
## Bsmt.Cond              2.366e+09  7.040e+09   0.336  0.73706
## BsmtFin.Type.1         2.749e+09  1.097e+09   2.505  0.01275 *
## Total.Bsmt.SF          1.704e+06  8.096e+06   0.210  0.83346
## Heating.QC            -1.040e+09  3.217e+09  -0.323  0.74675
## X2nd.Flr.SF            4.106e+07  1.296e+07   3.167  0.00169 **
## Bedroom.AbvGr         -4.936e+09  4.025e+09  -1.227  0.22092
## Kitchen.Qual           1.394e+10  4.957e+09   2.813  0.00523 **
## TotRms.AbvGrd          8.021e+09  2.420e+09   3.314  0.00103 **
## Functional             3.652e+08  4.218e+09   0.087  0.93107
## Fireplaces             7.943e+09  3.964e+09   2.004  0.04598 *
## Fireplace.Qu           9.885e+08  3.688e+09   0.268  0.78885
## Garage.TypeAttchd     -1.910e+10  2.456e+10  -0.778  0.43731
## Garage.TypeBasment    -3.239e+10  2.853e+10  -1.135  0.25717
## Garage.TypeBuiltIn    -1.979e+10  2.609e+10  -0.758  0.44873
## Garage.TypeCarPort     6.446e+09  4.301e+10   0.150  0.88096
## Garage.TypeDetchd     -1.195e+10  2.443e+10  -0.489  0.62509
## Garage.Yr.Blt         -5.511e+08  2.653e+08  -2.077  0.03859 *
## Garage.Finish          5.288e+08  3.358e+09   0.157  0.87498
## Garage.Cars2           2.330e+09  7.834e+09   0.297  0.76640
## Garage.Cars3           3.442e+10  1.189e+10   2.895  0.00407 **
## Garage.Cars4          -2.323e+09  3.035e+10  -0.077  0.93904
## Garage.Area           -1.550e+07  2.432e+07  -0.637  0.52433
## Wood.Deck.SF           4.928e+07  1.623e+07   3.036  0.00260 **
## Open.Porch.SF          1.678e+07  3.856e+07   0.435  0.66381
## Sale.TypeConLD        -5.901e+10  5.017e+10  -1.176  0.24038
## Sale.TypeConLI        -2.677e+08  3.056e+10  -0.009  0.99302
## Sale.TypeCWD          -9.434e+09  2.973e+10  -0.317  0.75123
## Sale.TypeNew           8.649e+09  1.553e+10   0.557  0.57800
## Sale.TypeWD           -1.946e+09  1.473e+10  -0.132  0.89498
## Sale.ConditionAlloca   1.205e+10  2.792e+10   0.432  0.66634
## Sale.ConditionFamily  -8.148e+09  1.711e+10  -0.476  0.63424
## Sale.ConditionNormal   4.407e+09  1.224e+10   0.360  0.71912
## Sale.ConditionPartial         NA         NA      NA       NA
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 3.424e+10 on 308 degrees of freedom
##   (632 observations deleted due to missingness)
## Multiple R-squared:  0.7244, Adjusted R-squared:  0.6716
## F-statistic: 13.72 on 59 and 308 DF,  p-value: < 2.2e-16
```

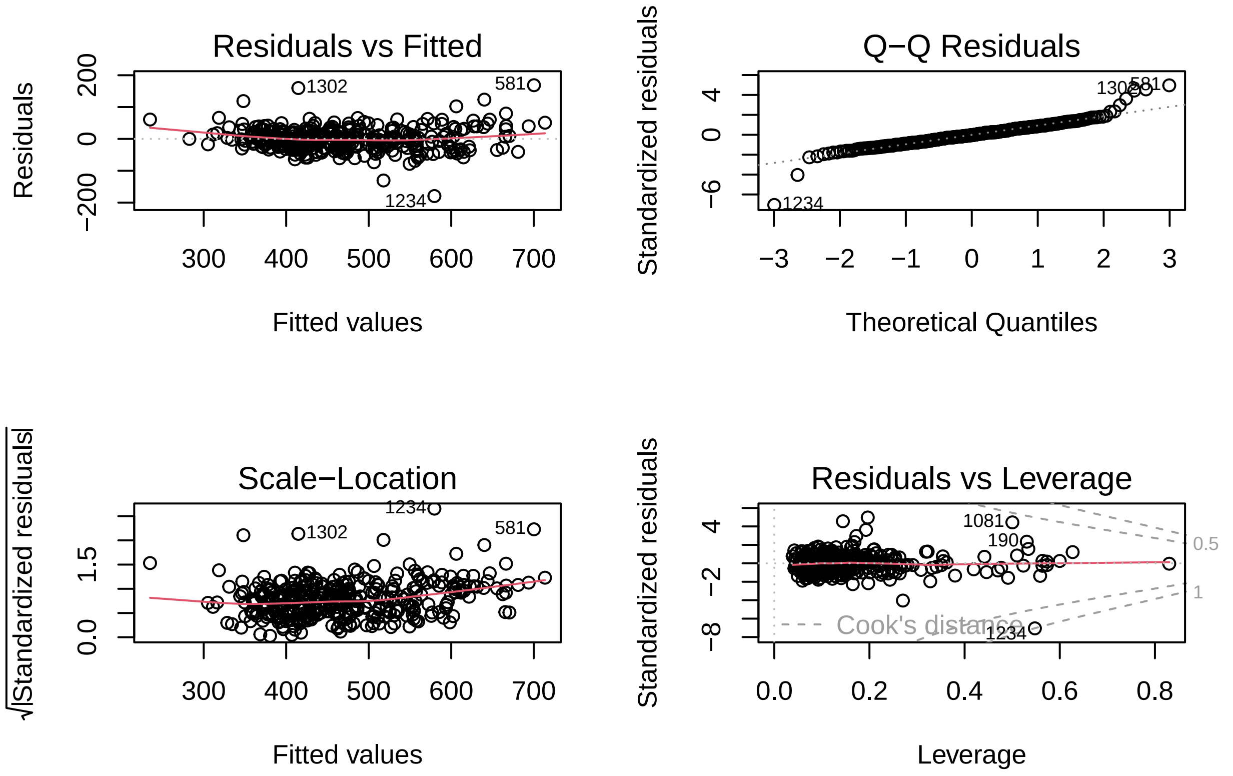
```
sqrt_lm <- lm(sqrt(SalePrice) ~ ., data = trainData)
summary(sqrt_lm)
```

```
##
## Call:
## lm(formula = sqrt(SalePrice) ~ ., data = trainData)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -179.62  -21.61   -1.02   20.94  168.61
##
## Coefficients: (3 not defined because of singularities)
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             6.010e+01  7.883e+01   0.762 0.446398
## MS.SubClass30          -6.427e+01  4.122e+01  -1.559 0.119965
## MS.SubClass50           1.281e+01  1.745e+01   0.734 0.463632
## MS.SubClass60          -3.150e+01  1.389e+01  -2.268 0.023991 *
## MS.SubClass75           2.167e+01  4.179e+01   0.519 0.604452
## MS.SubClass80          -9.878e+00  1.033e+01  -0.956 0.339871
## MS.SubClass85          -2.400e+01  1.612e+01  -1.489 0.137417
## MS.SubClass90          -4.381e+01  2.009e+01  -2.180 0.029983 *
## MS.SubClass120         -5.411e+00  1.021e+01  -0.530 0.596476
## MS.SubClass160         -4.400e+01  2.446e+01  -1.799 0.072979 .
## Lot.Frontage           -2.497e-01  7.317e-02  -3.413 0.000728 ***
## Lot.Area                5.389e-05  2.750e-04   0.196 0.844757
## Lot.Shape              -3.800e+00  3.964e+00  -0.959 0.338475
## Bldg.TypeDuplex                NA         NA      NA       NA
## Bldg.TypeTwnhs         -1.290e+01  2.151e+01  -0.600 0.548962
## Bldg.TypeTwnhsE                NA         NA      NA       NA
## Exterior.1stCemntBd     4.806e+01  4.510e+01   1.065 0.287487
## Exterior.1stHdBoard     4.082e+00  4.318e+01   0.095 0.924758
## Exterior.1stMetalSd     1.901e+01  4.349e+01   0.437 0.662288
## Exterior.1stPlywood    -9.147e+00  4.377e+01  -0.209 0.834602
## Exterior.1stStucco     -9.526e+01  4.950e+01  -1.925 0.055195 .
## Exterior.1stVinylSd     1.803e-01  4.374e+01   0.004 0.996714
## Exterior.1stWd Sdng    -2.142e+00  4.248e+01  -0.050 0.959819
## Exterior.1stWdShing    -2.021e+01  5.027e+01  -0.402 0.687983
## Mas.Vnr.TypeBrkCmn     -2.483e+00  2.259e+01  -0.110 0.912560
## Mas.Vnr.TypeBrkFace    -3.043e+00  1.630e+01  -0.187 0.851980
## Mas.Vnr.TypeStone      -3.037e+00  1.659e+01  -0.183 0.854886
## Mas.Vnr.Area            2.668e-02  2.094e-02   1.274 0.203529
## Exter.Qual              1.876e+01  6.369e+00   2.946 0.003469 **
## Bsmt.Qual               2.497e+01  5.942e+00   4.203 3.46e-05 ***
## Bsmt.Cond               4.382e+00  7.785e+00   0.563 0.573936
## BsmtFin.Type.1          3.575e+00  1.213e+00   2.947 0.003459 **
## Total.Bsmt.SF          -7.580e-03  8.952e-03  -0.847 0.397816
## Heating.QC              1.626e+00  3.557e+00   0.457 0.647925
## X2nd.Flr.SF             2.833e-02  1.433e-02   1.976 0.049039 *
## Bedroom.AbvGr          -5.106e-01  4.450e+00  -0.115 0.908733
## Kitchen.Qual            1.987e+01  5.482e+00   3.625 0.000338 ***
## TotRms.AbvGrd           1.138e+01  2.676e+00   4.253 2.80e-05 ***
## Functional              5.766e+00  4.665e+00   1.236 0.217396
## Fireplaces              1.577e+01  4.384e+00   3.597 0.000374 ***
## Fireplace.Qu           -9.514e-01  4.078e+00  -0.233 0.815676
## Garage.TypeAttchd      -7.483e+00  2.715e+01  -0.276 0.783046
## Garage.TypeBasment     -1.597e+00  3.155e+01  -0.051 0.959661
## Garage.TypeBuiltIn     -9.178e+00  2.885e+01  -0.318 0.750641
## Garage.TypeCarPort      2.298e+01  4.756e+01   0.483 0.629285
## Garage.TypeDetchd      -1.301e+01  2.702e+01  -0.481 0.630584
## Garage.Yr.Blt          -2.132e-01  2.934e-01  -0.727 0.467839
## Garage.Finish           1.535e+00  3.714e+00   0.413 0.679640
## Garage.Cars2            1.615e+01  8.663e+00   1.864 0.063244 .
## Garage.Cars3            5.373e+01  1.315e+01   4.086 5.60e-05 ***
## Garage.Cars4            4.052e+01  3.356e+01   1.207 0.228177
## Garage.Area             3.235e-02  2.689e-02   1.203 0.229845
## Wood.Deck.SF            6.184e-02  1.795e-02   3.445 0.000650 ***
## Open.Porch.SF           6.480e-02  4.264e-02   1.519 0.129663
## Sale.TypeConLD         -5.326e+01  5.547e+01  -0.960 0.337775
## Sale.TypeConLI         -2.107e+01  3.379e+01  -0.624 0.533411
## Sale.TypeCWD           -2.260e+01  3.288e+01  -0.687 0.492386
## Sale.TypeNew            2.083e+01  1.717e+01   1.213 0.226003
## Sale.TypeWD            -7.491e+00  1.629e+01  -0.460 0.645857
## Sale.ConditionAlloca    2.996e+01  3.087e+01   0.970 0.332657
## Sale.ConditionFamily   -6.278e+00  1.892e+01  -0.332 0.740247
## Sale.ConditionNormal    1.972e+01  1.354e+01   1.457 0.146130
## Sale.ConditionPartial          NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 37.86 on 308 degrees of freedom
##   (632 observations deleted due to missingness)
## Multiple R-squared:  0.8574, Adjusted R-squared:  0.8301
## F-statistic: 31.39 on 59 and 308 DF,  p-value: < 2.2e-16
```

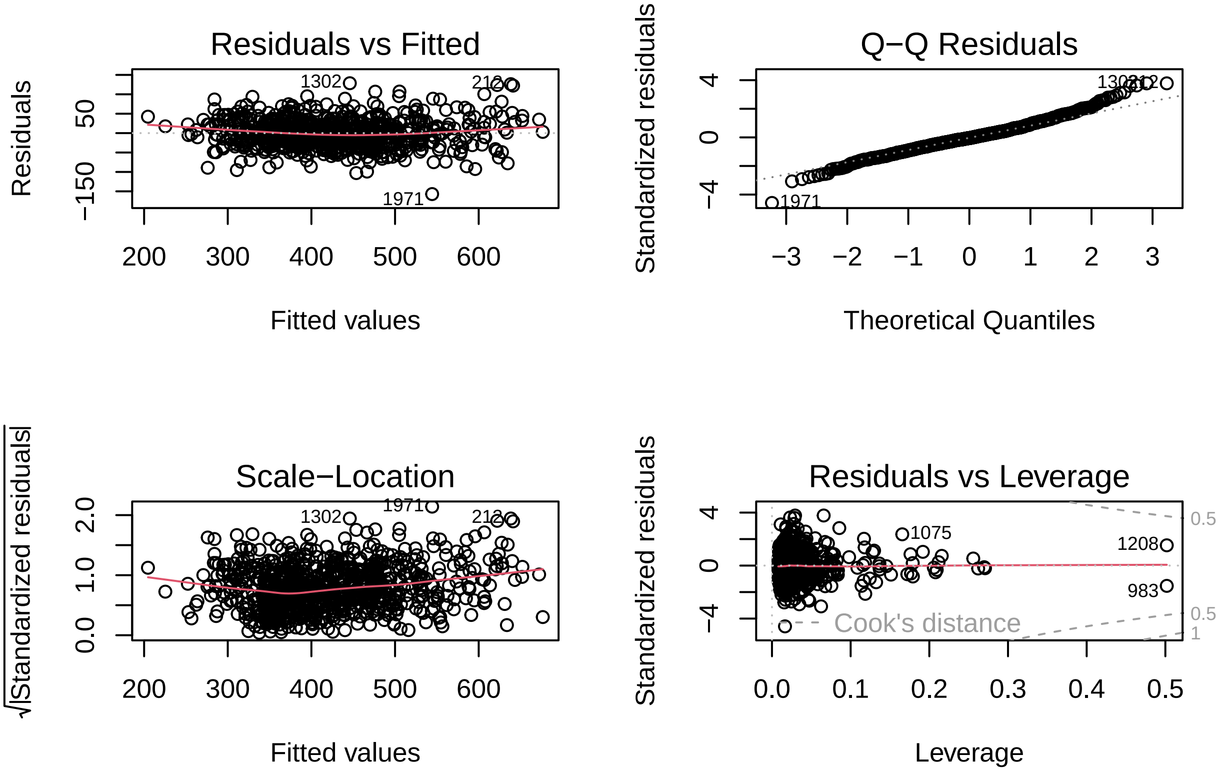
```
#train_Index = train_Index[- which(train_Index == 1208 | train_Index == 983)]

newTrainData = subset(data[train_Index, ], select = c(MS.SubClass, Exter.Qual, Bsmt.Qual, BsmtFin.Type.

sqrt_lm2 <- lm(sqrt(SalePrice) ~ ., data = newTrainData)
summary(sqrt_lm2)
```

```
##
## Call:
## lm(formula = sqrt(SalePrice) ~ ., data = newTrainData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -157.038  -20.513   -1.501   18.431  128.523
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      11.47869   12.75007   0.900 0.368243
## MS.SubClass30   -27.56043    6.68779  -4.121 4.17e-05 ***
## MS.SubClass40    -2.81238   24.68032  -0.114 0.909305
## MS.SubClass45   -19.44256   12.51498  -1.554 0.120694
## MS.SubClass50   -10.14344    4.51849  -2.245 0.025052 *
## MS.SubClass60   -13.84692    3.96212  -3.495 0.000501 ***
## MS.SubClass70   -10.79453    6.14301  -1.757 0.079270 .
## MS.SubClass75     8.56815   14.45885   0.593 0.553626
## MS.SubClass80   -20.56642    6.72581  -3.058 0.002305 **
## MS.SubClass85   -13.59336    9.70606  -1.401 0.161755
## MS.SubClass90  -34.31964    8.37589  -4.097 4.61e-05 ***
## MS.SubClass120  -4.28270    5.77400  -0.742 0.458477    
## MS.SubClass150 -79.47756   34.85232  -2.280 0.022849 *  
## MS.SubClass160 -32.86804    7.03300  -4.673 3.48e-06 ***
## MS.SubClass180 -68.18860   16.20728  -4.207 2.88e-05 ***
## MS.SubClass190 -25.79802   11.65284  -2.214 0.027122 *  
## Exter.Qual      28.61472    3.46999   8.246 6.79e-16 ***
## Bsmt.Qual       30.71444    2.93139  10.478  < 2e-16 ***
## BsmtFin.Type.1   3.46519    0.66516   5.210 2.42e-07 ***
## Fireplaces      21.64886    2.20879   9.801  < 2e-16 ***
## Kitchen.Qual    15.58006    2.70912   5.751 1.27e-08 ***
## TotRms.AbvGrd   14.25549    1.13242  12.588  < 2e-16 ***
## Garage.Cars1    25.38306    6.07271   4.180 3.24e-05 ***
## Garage.Cars2    33.44457    6.16386   5.426 7.67e-08 ***
## Garage.Cars3    70.64844    7.55526   9.351  < 2e-16 ***
## Garage.Cars4    70.87162   19.78700   3.582 0.000362 ***
## Garage.Cars5    68.81458   35.32506   1.948 0.051764 .  
## Wood.Deck.SF     0.04057    0.01058   3.833 0.000137 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 34.53 on 790 degrees of freedom
##   (180 observations deleted due to missingness)
## Multiple R-squared:  0.8489, Adjusted R-squared:  0.8437
## F-statistic: 164.3 on 27 and 790 DF,  p-value: < 2.2e-16
```

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



```r
par(mfrow=c(2,2))
plot(sqrt_lm2)
```



```
MSE.log <- mean(log_lm2$residuals^2)
print(MSE.log)
```

```
## [1] 0.02542539
```

```
MSE.linear <- mean(lm2$residuals^2)
print(MSE.linear)
```

```
## [1] 1097099195
```

```
#testing
newTestData = subset(testData, select = c(MS.SubClass, Exter.Qual, Bsmt.Qual, BsmtFin.Type.1, Fireplace

cont_index_test <- c()
for (i in 1:ncol(newTestData)) {
  if (!is.factor(newTestData[, i])) {
    cont_index_test = c(cont_index_test, i)
  }
}

for (y in cont_index_test) {
  for (x in 1:length(newTestData[, y])) {
    if (is.na(newTestData[x, y])){
      newTestData[x, y] = mean(newTestData[, y], na.rm = TRUE)
    }
  }
}

predictions_sqrt <- predict(sqrt_lm2, newdata = newTestData)


#calculating R^2
actualsSQRT <- sqrt(newTestData$SalePrice)
m_actualsSQRT <- mean(actualsSQRT)
ss_total <- sum((actualsSQRT - m_actualsSQRT)^2)
ss_residual <- sum((actualsSQRT - predictions_sqrt)^2, na.rm = TRUE)
rsquared <- 1 - (ss_residual / ss_total)
rsquared
```

```
## [1] 0.7926801
```

```
confint(sqrt_lm2, level = 0.95)
```

```
##                        2.5 %       97.5 %
## (Intercept)     -13.54932791  36.50671707
## MS.SubClass30   -40.68837430 -14.43248336
## MS.SubClass40   -51.25913054  45.63437605
## MS.SubClass45   -44.00909772   5.12397891
## MS.SubClass50   -19.01311409  -1.27376727
## MS.SubClass60   -21.62445290  -6.06938178
## MS.SubClass70   -22.85308746   1.26403527
## MS.SubClass75   -19.81416537  36.95046020
## MS.SubClass80   -33.76899943  -7.36384340
## MS.SubClass85   -32.64607536   5.45935195
## MS.SubClass90   -50.76127997 -17.87800220
## MS.SubClass120  -15.61690295   7.05150574
## MS.SubClass150 -147.89167081 -11.06344834
## MS.SubClass160  -46.67360850 -19.06246833
## MS.SubClass180 -100.00302585 -36.37417044
## MS.SubClass190  -48.67220204  -2.92383776
## Exter.Qual       21.80323673  35.42621217
## Bsmt.Qual        24.96019859  36.46868274
## BsmtFin.Type.1    2.15950576   4.77087638
## Fireplaces       17.31306438  25.98464604
## Kitchen.Qual     10.26213114  20.89799363
## TotRms.AbvGrd    12.03258115  16.47840841
## Garage.Cars1     13.46250452  37.30362460
## Garage.Cars2     21.34507951  45.54405359
## Garage.Cars3     55.81768469  85.47919452
## Garage.Cars4     32.03029874 109.71293846
## Garage.Cars5     -0.52750765 138.15666884
## Wood.Deck.SF      0.01979261   0.06134022
```

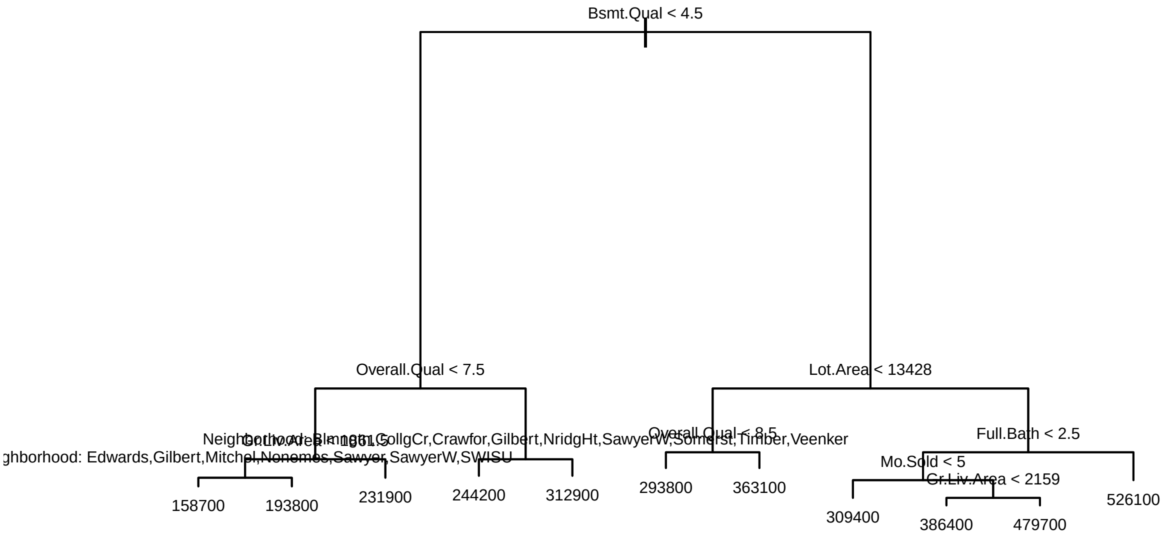
```
predInt3 = predict(sqrt_lm2, newdata = newTestData, interval = "predict")
```

##Decision tree model

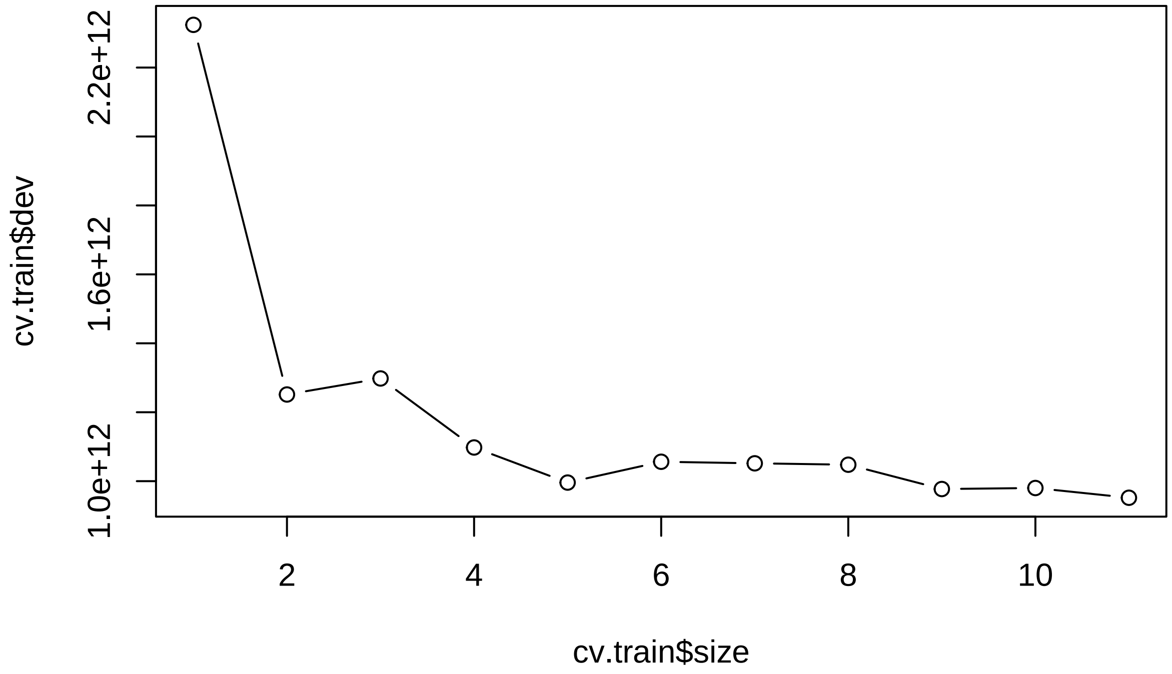
```r
require(tree)
trainTree <- tree(SalePrice ~., treeTrainData)
summary(trainTree)
```

```
##
## Regression tree:
## tree(formula = SalePrice ~ ., data = treeTrainData)
## Variables actually used in tree construction:
## [1] "Bsmt.Qual"    "Overall.Qual" "Gr.Liv.Area"  "Neighborhood" "Lot.Area"
## [6] "Full.Bath"    "Mo.Sold"
## Number of terminal nodes:  11
## Residual mean deviance:  1.726e+09 = 3.728e+11 / 216
## Distribution of residuals:
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -200800.0  -19770.0     801.1       0.0   20380.0  228900.0
```

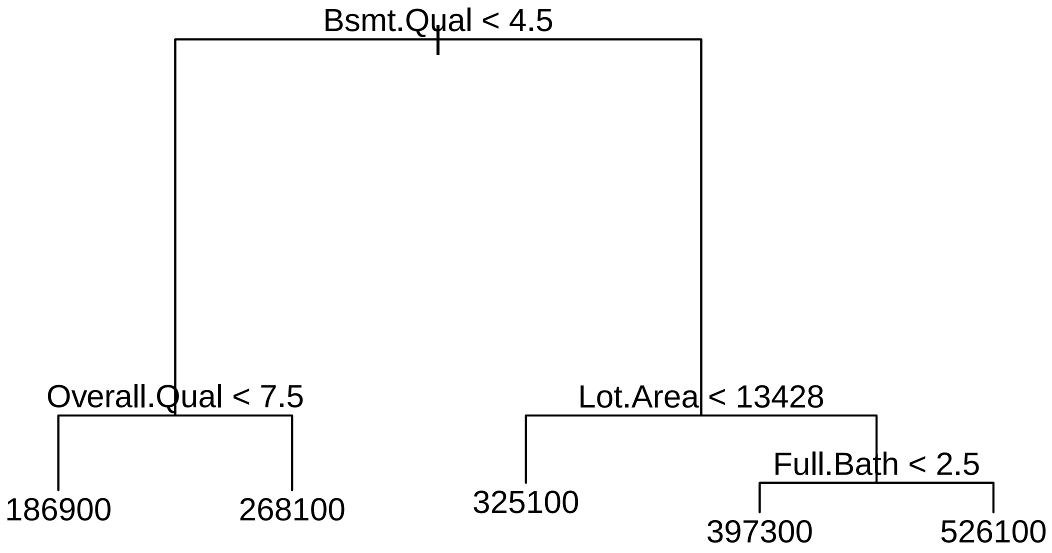
```r
plot(trainTree)
text(trainTree, pretty = 0, cex=0.5)
```



```r
predictionsTree = predict(trainTree, treeTestData)

#pruning
cv.train <- cv.tree(trainTree)
plot(cv.train$size, cv.train$dev, type = "b")
```



```
pruneTrain <- prune.tree(trainTree, best = 5)
plot(pruneTrain)
text(pruneTrain, pretty = 0)
```

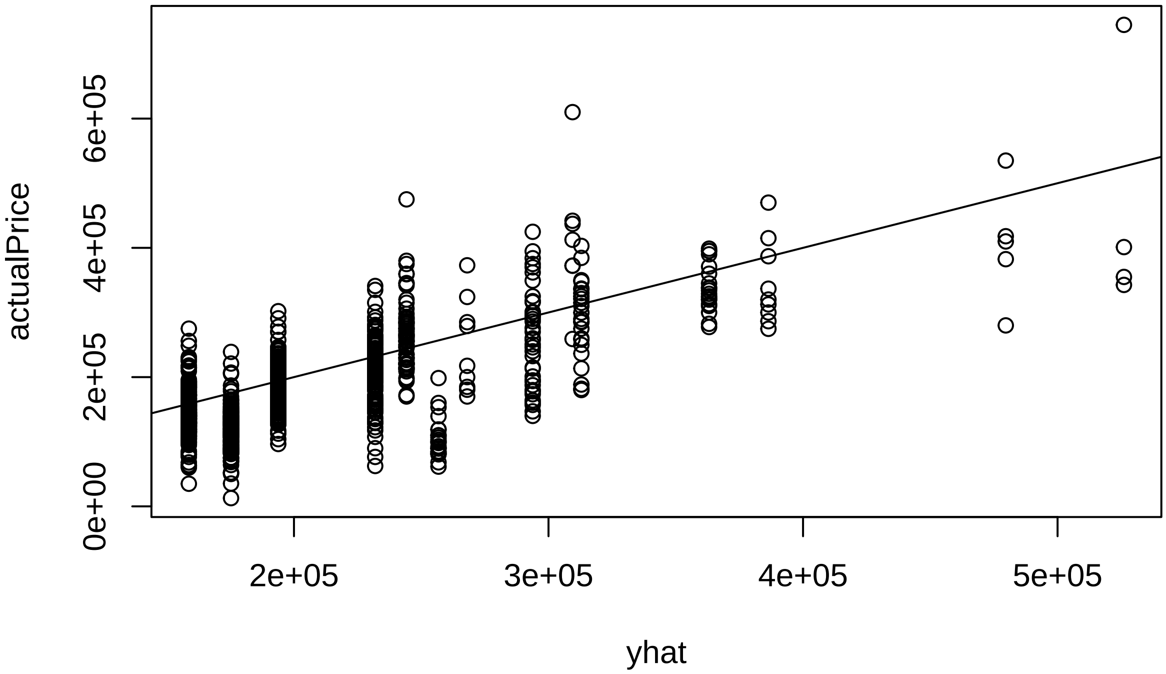


##Calculating R^2 + MSE for unpruned

```
#computing R^2
actualsT <- treeTestData$SalePrice
m_actuals <- mean(actualsT)
ss_total <- sum((actualsT - m_actuals)^2)
ss_residual <- sum((actualsT - predictionsTree)^2, na.rm = TRUE)
rsquared <- 1 - (ss_residual / ss_total)
rsquared
```

```
## [1] 0.4292883
```

```
#computing test mse
yhat <- predict(trainTree, newdata = treeTestData[1:55])
actualPrice <- treeTestData$SalePrice
plot(yhat, actualPrice)
abline(0, 1)
```



```
test.mse = mean((yhat - actualPrice)^2)
rmse = sqrt(test.mse)
print(test.mse)
```

```
## [1] 3385109322
```

```
print(rmse)
```

```
## [1] 58181.69
```

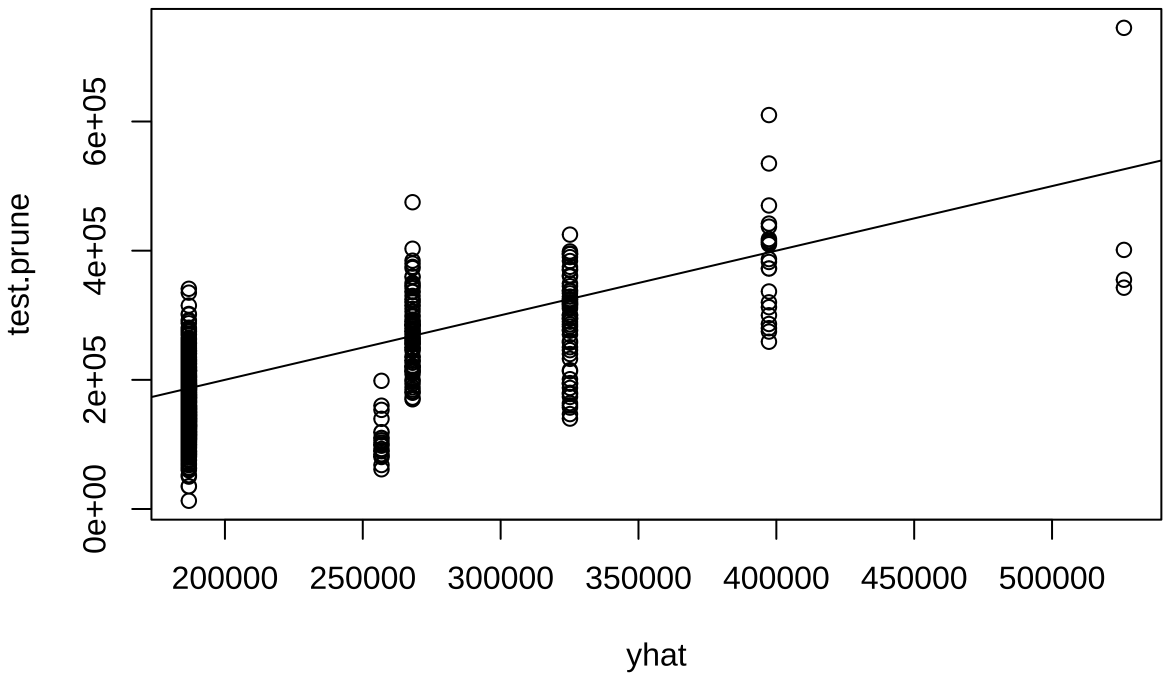
```
#this model leads to test predictions that are (on average) within approximately $58181.69$ of the true

predInt4 = predict(trainTree, newdata = treeTestData, interval = "predict")
```

##Calculating R^2 + MSE for pruned

```
yhat <- predict(pruneTrain, newdata = treeTestData[1:55])

actualsTP <- treeTestData$SalePrice
m_actuals <- mean(actualsTP)
ss_total <- sum((actualsTP - m_actuals)^2)
ss_residual <- sum((actualsTP - yhat)^2, na.rm = TRUE)
rsquared <- 1 - (ss_residual / ss_total)
rsquared
```

```
## [1] 0.3081712
```

```
test.prune <- treeTestData$SalePrice
plot(yhat, test.prune)
abline(0, 1)
```



```
test.mse.prune = mean((yhat - test.prune)^2)
rmse.prune = sqrt(test.mse.prune)
print(test.mse.prune)
```

```
## [1] 4103501361
```

```
print(rmse.prune)
```

```
## [1] 64058.58
```

```
#this model leads to test predictions that are (on average) within approximately $64058.58$ of the true

predInt5 = predict(pruneTrain, newdata = treeTestData, interval = "predict")
```

#Comparing prediction intervals between models

```
paste("Pred. int for lm model: ", mean(predInt1[,1], na.rm = TRUE), "to", mean(predInt1[,3], na.rm = TRU
```

```
## [1] "Pred. int for lm model:  182188.477389569 to 249547.931072331"
```

```
paste("Pred. int for log. lm model: ", mean(predInt2[,1], na.rm = TRUE), "to", mean(predInt2[,3], na.rm
```

```
## [1] "Pred. int for log. lm model:  12.0312306482992 to 12.3551154214485"
```

```
paste("Pred. int for sqrt. model: ", mean(predInt3[,1], na.rm = TRUE), "to", mean(predInt3[,3], na.rm =
```

```
## [1] "Pred. int for sqrt. model:  418.092002635025 to 487.020212450143"
```