2023

AMES Housing Kaggle

Erica Brooks and Steven Cox

Statistics I Project

8/1/2023

Introduction

The real estate market is a complex web of factors that influence property prices. In this paper, we delve into the Kaggle Housing Prices challenge to analyze and understand the relationship between house sale prices and various attributes in the context of Ames, Iowa. Our investigation focuses on two distinct analyses that provide valuable insights to Century 21 Ames, a prominent real estate company operating in the area.

The first analysis we will be working with Century 21 to answer a critical question pertaining to their business. Specifically, they want to determine the relationship between the sale price of a house and the square footage of its living area. Additionally, they are keen to identify if this relationship varies depending on the neighborhood in which the house is located. The neighborhoods of interest North Ames, Edwards, and Brookside.

In the second analysis, our focus will be to use four predictive models (Forward Selection, Backward Elimination, Stepwise Selection, and a unique model of our own) to predict housing costs in all the given neighborhoods. We will provide a description of each model, its strengths, weaknesses, and ultimately compare each of the models using the Adjusted R-squared and PRESS (Predicted Residual Sum of Squares) values to predict how well the model will perform. As the final test, Kaggle has provided a test data set that we will use to predict the Sales Costs. The overall Kaggle score will determine which model performed better.

Data Description

The Ames Housing dataset at is divided into a training set with 1,460 observations and a test set with 1,459 observations, comprising residential housing information. Spanning 79 explanatory variables, the data encapsulates a comprehensive range of features such as zoning type, lot area, building structure, roof material, heating systems, garage details, and various other attributes related to the property's condition and appearance.

The complexity and breadth of the data provide an opportunity to delve into intricate patterns and relationships in residential housing. For Analysis I, we will be focusing on three specific neighborhoods North Ames, Edwards, and Brookside and categories Sale Cost and General Living Area. For Analysis II, the entire data set will be used with specific modifications that will be discussed later. For more information regarding this data set, please refer to Kaggle's House Prices - Advanced Regression Techniques challenge (link provided in the appendix)

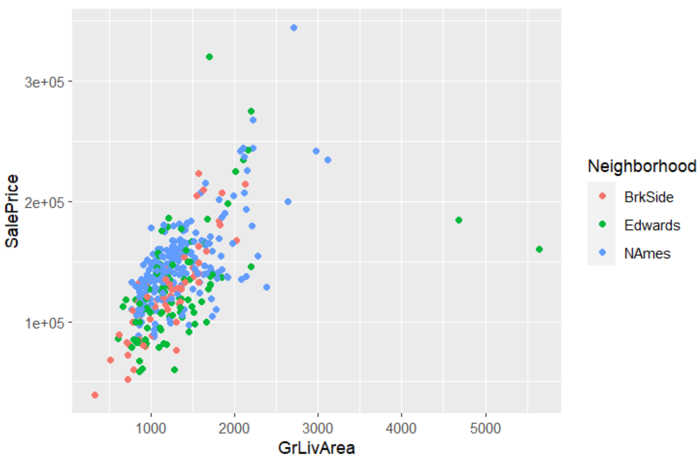
Analysis I Century 21 Ames Real Estate

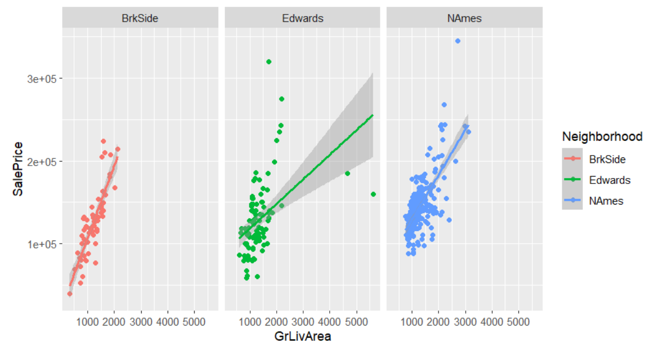
Problem

Century 21 Ames a real estate company sells houses in the North Ames, Edwards, and Brookside neighborhoods in Ames, Iowa. They would like to know if the sales price of a house is related to the square footage of the living area of the house, and whether or not the sales price depends on which neighborhood the house is located in.

Exploratory Data Analysis

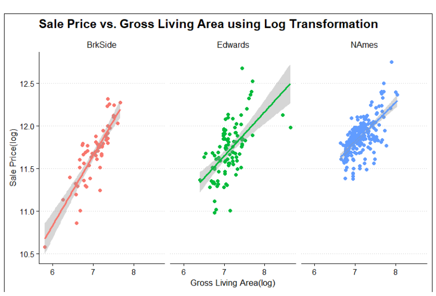
This is the data that records the sale price of a house based on the living area of the house (increments of 100 sq. ft.) and the neighborhood that the house is located in. The neighborhoods of interest are North Ames, Edwards, and Brookside. Based on the initial scatter plots we can see that there appears to be a positive linear relationship between the sale price and total living area.

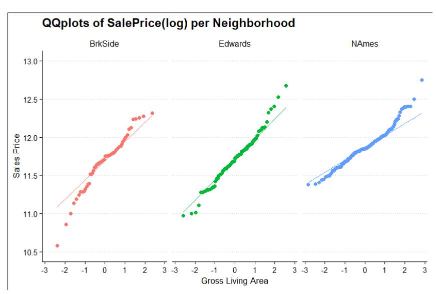


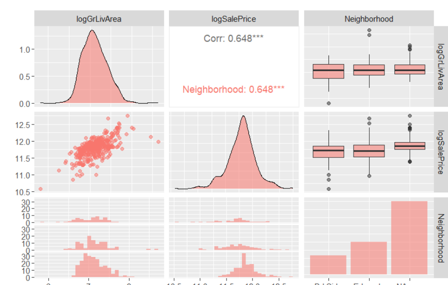


We will proceed with cleaning and transforming the data and constructing a multiple linear regression model in which the sale price depends linearly on the square footage of the house in each neighborhood. We will allow for possible different slopes and intercepts.

A Log-Log Transformation was done on the sale price, and the living area







Judging from the scatter plots, Q-Q plot, and plot matrix there is some evidence of outliers but nothing strong enough to prove against the data does not follow a normal distribution plus based on CLT we have more than enough data points. We will assume that the observations are independent.

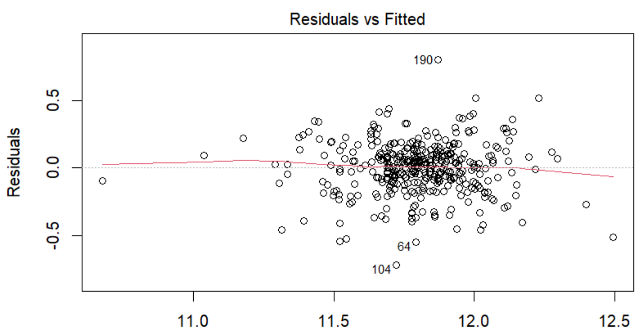
Building the Model

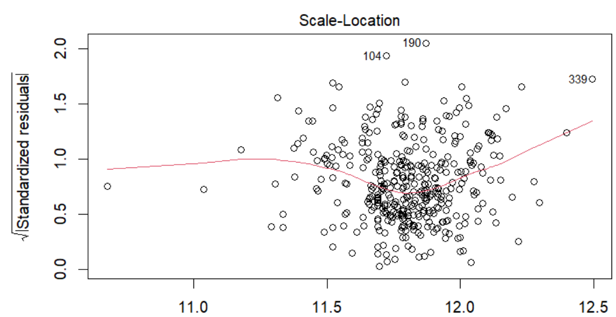
Model: logSalePrice = $\beta_0 + \beta_1 logGrLivArea + \beta_2 Neighborhood + \beta_3 logGrLivArea * Neighborhood$

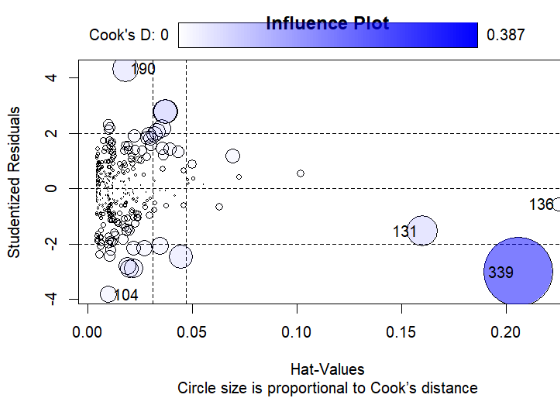
According to the model we can see that the intercepts and slopes are significantly different for each neighborhood based on the p-values at a .05 significance level.

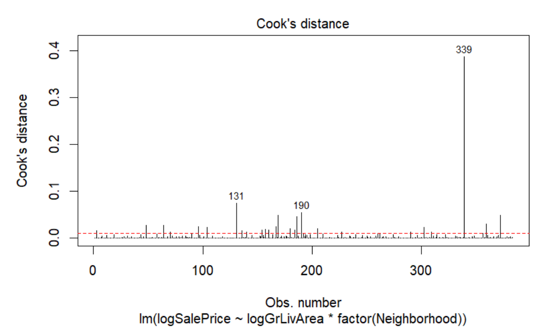
```
Call:
lm(formula = logSalePrice ~ logGrLivArea * factor(Neighborhood),
    data = century21)

Residuals:
     Min       1Q   Median       3Q      Max
-0.72080 -0.10353  0.02184  0.10586  0.80470

Coefficients:
                                         Estimate Std. Error t value Pr(>|t|)
(Intercept)                               5.91292    0.50459  11.718  < 2e-16 ***
logGrLivArea                              0.81965    0.07163  11.443  < 2e-16 ***
factor(Neighborhood)Edwards               2.09359    0.64589   3.241   0.0013 **
factor(Neighborhood)NAmes                 2.57981    0.59988   4.301 2.17e-05 ***
logGrLivArea:factor(Neighborhood)Edwards -0.29998    0.09122  -3.289   0.0011 **
logGrLivArea:factor(Neighborhood)NAmes   -0.34662    0.08482  -4.087 5.35e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1923 on 377 degrees of freedom
Multiple R-squared:  0.5121,	Adjusted R-squared:  0.5056
F-statistic: 79.14 on 5 and 377 DF,  p-value: < 2.2e-16
```

Checking Assumptions









There is visual evidence that we have a few extremely influential points within our data. Upon further investigation, we concluded that these outliers may have some influence on the outcome of our model. We made the decision to remove these outliers and re-run our model also taking into account varying y intercepts and slopes for each of the neighborhoods.

Model2: logSalePrice = $\beta_0 + \beta_1 logGrLivArea + \beta_2 Neighborhood + \beta_3 logGrLivArea * Neighborhood$

```
Call:
lm(formula = logSalePrice ~ logGrLivArea * factor(Neighborhood),
    data = century21_rmOutliers)

Residuals:
     Min       1Q   Median       3Q      Max
-0.73636 -0.10922  0.02052  0.10528  0.74523

Coefficients:
                                     Estimate Std. Error t value Pr(>|t|)
(Intercept)                           6.06485    0.56120  10.807  < 2e-16 ***
logGrLivArea                          0.79836    0.07944  10.050  < 2e-16 ***
factor(Neighborhood)Edwards           0.85824    0.74594   1.151 0.250652
factor(Neighborhood)NAmes             2.42788    0.64574   3.760 0.000197 ***
logGrLivArea:factor(Neighborhood)Edwards -0.12502    0.10531  -1.187 0.235924
logGrLivArea:factor(Neighborhood)NAmes   -0.32534    0.09117  -3.568 0.000406 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1893 on 374 degrees of freedom
Multiple R-squared:  0.5023,	Adjusted R-squared:  0.4956
F-statistic: 75.49 on 5 and 374 DF,  p-value: < 2.2e-16
```

```
> confint(fit3, level=0.95)
                                              2.5 %      97.5 %
(Intercept)                               4.9613550  7.16834164
logGrLivArea                              0.6421531  0.95456935
factor(Neighborhood)Edwards              -0.6085209  2.32501004
factor(Neighborhood)NAmes                 1.1581366  3.69762199
logGrLivArea:factor(Neighborhood)Edwards -0.3321064  0.08205915
logGrLivArea:factor(Neighborhood)NAmes   -0.5046076 -0.14606762
```

Parameters

A change in sales price from Brookside to Edwards neighborhoods based on square footage does not appear to be significantly different, although with a change in Brookside to North Ames neighborhood there appears to be a significant change in the sale price based on square footage.

Our best estimate is a change in sale price from Brookside to North Ames neighborhoods. For every 100 sq. ft. in living area, the sale price for a house in North Ames will increase by \$1133.50($e^{2.4278}$). A 95% confidence interval for an increase in sales price per 100 sq. ft is (\$319.00, \$4045.00) ($e^{1.16}$, $e^{3.70}$). Since the slopes are different this change in sale price can vary. This data can be rerun by grouping North Ames and Edwards neighborhoods.

Conclusion

There is a relationship between the sale price and the square footage of a house. An increase in square footage results in an increase in house prices. It also appears that the sale price of a house does depend on the neighborhood. Brookside and Edwards appear to have houses similar in price while North Ames neighborhood in more expensive.

For an interactive experience comparing the Sales Cost and Cost per square foot of the housing market in Ames, Iowa, please visit our R Shiny app located at https://scox97.shinyapps.io/ames_neighborhood.

Analysis Question 2

Problem Statement

To assess the accuracy and effectiveness of various predictive models, our task involves analyzing a dataset containing 79 explanatory variables pertaining to housing characteristics. Specifically, we aim to evaluate four specific models - Forward Selection, Backward Elimination, Stepwise Selection, and an internally developed model. Our focus is on the Ames area in Iowa as we strive to determine which of these models provides the most precise and efficient predictions for the Sales Costs of houses within this region.

Exploratory Data Analysis

Of the 79 explanatory variables, we had 15424 missing values, most of which were in four categories with missing percentages of Pool Quality (99.7%), Miscellaneous Features (96.4%) , Alley (93.2%), and Fence (80.4%). These variables were simply removed from the data set as we suspect they would have little effect on the modeling. Of the remaining categories, it was evident that there wasn't a distinction of the house not having the feature, and therefore most of the remaining NAs are now interpreted as None for categorical and 0 for numerical.

Checking Assumptions

Similarly, to Analysis I, using residual and qq plots, we found that performing a Log-Log transformation on the sale price, general living area, and lot area had a significant improvement on normalizing those categories. However, with almost 3000, observations, the Central Limit Theorem (CLT) comes into play, ensuring that the distribution of the sum of independent variables will become more normalized. Please refer to the appendix to observe the residual and qq graphs.

Regarding independence between the variables while modeling with Forward, Backward, and Stepwise methods, we will assume independence between all of the variables. However, in our custom model, we decided to group some of the variables that we believed to not be as independent, along with reassigning some of the levels for a few variables.

Additionally, since all the data is bound to the Ames, Iowa region, we can only apply inference on houses within Ames.

Influential Points

In the same manner of identifying influential points from Analysis I, we had to broaden our categories to include all the neighborhoods and explanatory variables. Using Cook's D values and visualizing the data on the graph, we found three observations that had significant impact on the adjusted R squared value of the model. Therefore, we decided to leave these observations out of the analysis, however we suggest further investigation.

Model Selections

For analysis purposes, we decided to use the same parameters and methods for *Forward, Backward, and Stepwise*. In training, we opted for using Linear Modeling and Leave-One-Out Cross-Validation (LOOCV). For variable inclusion, we chose to use Akaike Information Criterion (AIC) as the criteria for variable predictors. After performing the modeling, we found that both Forward and Stepwise had the same adjusted R squared values, while Backward had a significantly lower score. Further analysis as to the deviation between these models is warranted. See the results below for actual scores.

Custom

In our custom model, we developed the formula below where we used predictors that we thought relevant to what a buyer would ask while categorizing what season the purchase would be made. With limiting the explanatory variables, we are curious if there was a significant impact on the overall performance between it and other models.

"`logSalePrice ~ GarageCars+OverallQual+TotalBathrooms+SeasonSold+logLotArea+logGrLivArea`"

Comparing Competing Models

Model Results

Model	Adjusted.R2	CV.PRESS	Kaggle Score
Forward Model:	0.9108818	20.26023	0.13898
Backward Model:	0.7840674	49.26270	0.20348
Stepwise Model:	0.9108818	20.26023	0.13898
Custom Model:	0.8382415	37.10588	0.17390

Upon analyzing the data, a clear pattern emerges among the four models assessed in forecasting housing costs in Ames, Iowa. The Forward Model and Stepwise Model both yielded an impressive Adjusted R2 value of 0.9108818, indicating a strong fit to the data. These models also showcased exceptional performance on the Kaggle test dataset with their lowest Kaggle Scores of 0.13898, emphasizing their superior predictive capabilities.

However, the Backward Model produced a higher Kaggle Score of 0.20348 while achieving an Adjusted R2 value of 0.7840674, suggesting a less accurate prediction compared to its counterparts. Sitting between these extremes was the Custom Model which achieved an Adjusted R2 value of 0.8382415 and a corresponding Kaggle Score of 0.17390.

Conclusion

The results demonstrate a consistent relationship between the values for Adjusted R2 and Kaggle Scores; where higher values for Adjusted R2 align with lower (and thus better) scores on Kaggle tests. This correlation highlights how effectively Adjusted R2 predicts model performance on unseen data as validated by the outcomes from the Kaggle competition.

In conclusion, both Forward and Stepwise Models emerged as highly promising approaches due to their robustness in fitting models as well as their superior predictive accuracy when compared to other alternatives analyzed during this study.

Links to External Code and Data

[Group Git Hub]

[RShiny App]

Appendix:
Case Study: House Prices and Regressions

DS 6371 Kaggle Project

Erica Brooks and Steven Cox

8/6/2023

Introduction

Data Description

A description of the data, where it came from, and what it contains.

Read the Data

```
ames_train <- read.csv("Data/train.csv")
ames_test <- read.csv("Data/test.csv")

#Merging the two datasets to have one complete set.  A new column indicating
Train/Test will be added
ames_train$train <- 'train'
ames_test$SalePrice <- NA # Needed to bind two sets together
ames_test$train <- 'test'
ames <- rbind(ames_train, ames_test)

# Verify data frame
str(ames)

## 'data.frame':    2919 obs. of  82 variables:
##  $ Id           : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ MSSubClass   : int  60 20 60 70 60 50 20 60 50 190 ...
##  $ MSZoning     : chr  "RL" "RL" "RL" "RL" ...
##  $ LotFrontage  : int  65 80 68 60 84 85 75 NA 51 50 ...
##  $ LotArea      : int  8450 9600 11250 9550 14260 14115 10084 10382 6120 7
420 ...
##  $ Street       : chr  "Pave" "Pave" "Pave" "Pave" ...
##  $ Alley        : chr  NA NA NA NA ...
##  $ LotShape     : chr  "Reg" "Reg" "IR1" "IR1" ...
##  $ LandContour  : chr  "Lvl" "Lvl" "Lvl" "Lvl" ...
##  $ Utilities    : chr  "AllPub" "AllPub" "AllPub" "AllPub" ...
##  $ LotConfig    : chr  "Inside" "FR2" "Inside" "Corner" ...
##  $ LandSlope    : chr  "Gtl" "Gtl" "Gtl" "Gtl" ...
##  $ Neighborhood : chr  "CollgCr" "Veenker" "CollgCr" "Crawfor" ...
##  $ Condition1   : chr  "Norm" "Feedr" "Norm" "Norm" ...
##  $ Condition2   : chr  "Norm" "Norm" "Norm" "Norm" ...
##  $ BldgType     : chr  "1Fam" "1Fam" "1Fam" "1Fam" ...
##  $ HouseStyle   : chr  "2Story" "1Story" "2Story" "2Story" ...
##  $ OverallQual  : int  7 6 7 7 8 5 8 7 7 5 ...
```

```
##  $ OverallCond  : int  5 8 5 5 5 5 5 6 5 6 ...
##  $ YearBuilt    : int  2003 1976 2001 1915 2000 1993 2004 1973 1931 1939 .
..
##  $ YearRemodAdd : int  2003 1976 2002 1970 2000 1995 2005 1973 1950 1950 .
..
##  $ RoofStyle    : chr  "Gable" "Gable" "Gable" "Gable" ...
##  $ RoofMatl     : chr  "CompShg" "CompShg" "CompShg" "CompShg" ...
##  $ Exterior1st  : chr  "VinylSd" "MetalSd" "VinylSd" "Wd Sdng" ...
##  $ Exterior2nd  : chr  "VinylSd" "MetalSd" "VinylSd" "Wd Shng" ...
##  $ MasVnrType   : chr  "BrkFace" "None" "BrkFace" "None" ...
##  $ MasVnrArea   : int  196 0 162 0 350 0 186 240 0 0 ...
##  $ ExterQual    : chr  "Gd" "TA" "Gd" "TA" ...
##  $ ExterCond    : chr  "TA" "TA" "TA" "TA" ...
##  $ Foundation   : chr  "PConc" "CBlock" "PConc" "BrkTil" ...
##  $ BsmtQual     : chr  "Gd" "Gd" "Gd" "TA" ...
##  $ BsmtCond     : chr  "TA" "TA" "TA" "Gd" ...
##  $ BsmtExposure : chr  "No" "Gd" "Mn" "No" ...
##  $ BsmtFinType1 : chr  "GLQ" "ALQ" "GLQ" "ALQ" ...
##  $ BsmtFinSF1   : int  706 978 486 216 655 732 1369 859 0 851 ...
##  $ BsmtFinType2 : chr  "Unf" "Unf" "Unf" "Unf" ...
##  $ BsmtFinSF2   : int  0 0 0 0 0 0 0 32 0 0 ...
##  $ BsmtUnfSF    : int  150 284 434 540 490 64 317 216 952 140 ...
##  $ TotalBsmtSF  : int  856 1262 920 756 1145 796 1686 1107 952 991 ...
##  $ Heating      : chr  "GasA" "GasA" "GasA" "GasA" ...
##  $ HeatingQC    : chr  "Ex" "Ex" "Ex" "Gd" ...
##  $ CentralAir   : chr  "Y" "Y" "Y" "Y" ...
##  $ Electrical   : chr  "SBrkr" "SBrkr" "SBrkr" "SBrkr" ...
##  $ X1stFlrSF    : int  856 1262 920 961 1145 796 1694 1107 1022 1077 ...
##  $ X2ndFlrSF    : int  854 0 866 756 1053 566 0 983 752 0 ...
##  $ LowQualFinSF : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ GrLivArea    : int  1710 1262 1786 1717 2198 1362 1694 2090 1774 1077 .
..
##  $ BsmtFullBath : int  1 0 1 1 1 1 1 1 0 1 ...
##  $ BsmtHalfBath : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ FullBath     : int  2 2 2 1 2 1 2 2 2 1 ...
##  $ HalfBath     : int  1 0 1 0 1 1 0 1 0 0 ...
##  $ BedroomAbvGr : int  3 3 3 3 4 1 3 3 2 2 ...
##  $ KitchenAbvGr : int  1 1 1 1 1 1 1 1 2 2 ...
##  $ KitchenQual  : chr  "Gd" "TA" "Gd" "Gd" ...
##  $ TotRmsAbvGrd : int  8 6 6 7 9 5 7 7 8 5 ...
##  $ Functional   : chr  "Typ" "Typ" "Typ" "Typ" ...
##  $ Fireplaces   : int  0 1 1 1 1 0 1 2 2 2 ...
##  $ FireplaceQu  : chr  NA "TA" "TA" "Gd" ...
##  $ GarageType   : chr  "Attchd" "Attchd" "Attchd" "Detchd" ...
##  $ GarageYrBlt  : int  2003 1976 2001 1998 2000 1993 2004 1973 1931 1939 .
..
##  $ GarageFinish : chr  "RFn" "RFn" "RFn" "Unf" ...
##  $ GarageCars   : int  2 2 2 3 3 2 2 2 2 1 ...
##  $ GarageArea   : int  548 460 608 642 836 480 636 484 468 205 ...
##  $ GarageQual   : chr  "TA" "TA" "TA" "TA" ...
```

```
##  $ GarageCond   : chr  "TA" "TA" "TA" "TA" ...
##  $ PavedDrive   : chr  "Y" "Y" "Y" "Y" ...
##  $ WoodDeckSF   : int  0 298 0 0 192 40 255 235 90 0 ...
##  $ OpenPorchSF  : int  61 0 42 35 84 30 57 204 0 4 ...
##  $ EnclosedPorch: int  0 0 0 272 0 0 0 228 205 0 ...
##  $ X3SsnPorch   : int  0 0 0 0 0 320 0 0 0 0 ...
##  $ ScreenPorch  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PoolArea     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PoolQC       : chr  NA NA NA NA ...
##  $ Fence        : chr  NA NA NA NA ...
##  $ MiscFeature  : chr  NA NA NA NA ...
##  $ MiscVal      : int  0 0 0 0 0 700 0 350 0 0 ...
##  $ MoSold       : int  2 5 9 2 12 10 8 11 4 1 ...
##  $ YrSold       : int  2008 2007 2008 2006 2008 2009 2007 2009 2008 2008 .
..
##  $ SaleType     : chr  "WD" "WD" "WD" "WD" ...
##  $ SaleCondition: chr  "Normal" "Normal" "Normal" "Abnorml" ...
##  $ SalePrice    : int  208500 181500 223500 140000 250000 143000 307000 20
0000 129900 118000 ...
##  $ train        : chr  "train" "train" "train" "train" ...

# summary(ames)
# str(ames)
```

Exploratory Analysis

###Investigating Missing Variables

```
#Find number of missing observations in each column
missing_count <- ames %>% select(-SalePrice) %>% sapply(function(x) sum(is.na
(x)))
missing_table <- as.data.frame(missing_count) %>%
  filter(missing_count >= 1) %>% arrange(desc(missing_count))

# print(missing_table)
  #OR
# Print the table using kable with descending order and ignoring counts less
than 1
kable(missing_table, caption = "Missing Value Counts (Descending Order)", ali
gn = "c")
```

Missing Value Counts (Descending Order)

	missing_count
PoolQC	2909
MiscFeature	2814
Alley	2721
Fence	2348

	missing_count
FireplaceQu	1420
LotFrontage	486
GarageYrBlt	159
GarageFinish	159
GarageQual	159
GarageCond	159
GarageType	157
BsmtCond	82
BsmtExposure	82
BsmtQual	81
BsmtFinType2	80
BsmtFinType1	79
MasVnrType	24
MasVnrArea	23
MSZoning	4
Utilities	2
BsmtFullBath	2
BsmtHalfBath	2
Functional	2
Exterior1st	1
Exterior2nd	1
BsmtFinSF1	1
BsmtFinSF2	1
BsmtUnfSF	1
TotalBsmtSF	1
Electrical	1
KitchenQual	1
GarageCars	1
GarageArea	1
SaleType	1

```r
total_NAs <- sum(is.na(ames[, !(names(ames) %in% "SaleCost")]))
cat("Total number of missing variables:", total_NAs)

## Total number of missing variables: 15424
```

```r
# Creating a list of possible columns to remove based on percentage of NAs.
# Data containing more than 80% missing values will be removed.

# Function to calculate percentage of NA values in a column
na_percentage <- function(x) {
  return(round(sum(is.na(x)) / length(x) * 100, digits = 1))
}
# Find number of NAs in each column, excluding SalesPrice
na_percentages <- ames %>% select(-SalePrice) %>% sapply(na_percentage)
na_percentages <- data.frame(Column = names(na_percentages), Percentage_NA =
na_percentages)
# Sort the dataframe by the Percentage_NA column
na_percentages <- na_percentages[order(-na_percentages$Percentage_NA), ]
# Print sorted dataframe
head(na_percentages)
```

```
##                  Column Percentage_NA
## PoolQC           PoolQC          99.7
## MiscFeature MiscFeature          96.4
## Alley             Alley          93.2
## Fence             Fence          80.4
## FireplaceQu FireplaceQu          48.6
## LotFrontage LotFrontage          16.6
```

There are four parameters that have very high missing value percentages: PoolQC, MiscFeature, and Ally all have more than 90%. Fence is also missing slightly over 80% of it’s value. These categories will be removed as it would take considerable resources to accurately handle this data.

###Removing categories with high missing values

```r
high_na_columns <- na_percentages[na_percentages$Percentage_NA > 80, ]
ames_subset <- ames[, !(names(ames) %in% high_na_columns$Column)]
```

Cleaning remaining missing data

Note: It appears that most of the remaining missing values have logical explanations. For example, GarageType does not have any instances of no garage, and the associated variables for garage type are also missing. Therefore for missing data values that are numeric and have a “Type” variable associated with it will be replaced with a 0, while the “type” will be replaced with “None”.

```r
# Investigating additional missing values.
missing_count <- ames_subset %>% select(-SalePrice) %>% sapply(function(x) su
m(is.na(x)))
missing_table <- as.data.frame(missing_count) %>%
  filter(missing_count >= 1) %>%
  arrange(desc(missing_count))
```

```r
kable(missing_table, caption = "Missing Value Counts (Descending Order)", ali
gn = "c")
```

Missing Value Counts (Descending Order)

	missing_count
FireplaceQu	1420
LotFrontage	486
GarageYrBlt	159
GarageFinish	159
GarageQual	159
GarageCond	159
GarageType	157
BsmtCond	82
BsmtExposure	82
BsmtQual	81
BsmtFinType2	80
BsmtFinType1	79
MasVnrType	24
MasVnrArea	23
MSZoning	4
Utilities	2
BsmtFullBath	2
BsmtHalfBath	2
Functional	2
Exterior1st	1
Exterior2nd	1
BsmtFinSF1	1
BsmtFinSF2	1
BsmtUnfSF	1
TotalBsmtSF	1
Electrical	1
KitchenQual	1
GarageCars	1
GarageArea	1
SaleType	1

```r
## Assume NA means that the property does not have it and assign 0 as value.
ames_subset <- ames_subset %>%
```

```r
  mutate(
    PoolArea = ifelse(is.na(PoolArea), 0, PoolArea),
    GarageType = ifelse(is.na(GarageType), "None", GarageType),
    GarageYrBlt = ifelse(is.na(GarageYrBlt), 0, GarageYrBlt),
    GarageFinish = ifelse(is.na(GarageFinish), "None", GarageFinish),
    GarageCars = ifelse(is.na(GarageCars), 0, GarageCars),
    GarageArea = ifelse(is.na(GarageArea), 0, GarageArea),
    GarageQual = ifelse(is.na(GarageQual), "None", GarageQual),
    GarageCond = ifelse(is.na(GarageCond), "None", GarageCond),
    BsmtQual = ifelse(is.na(BsmtQual), "None", BsmtQual),
    BsmtCond = ifelse(is.na(BsmtCond), "None", BsmtCond),
    BsmtExposure = ifelse(is.na(BsmtExposure), "None", BsmtExposure),
    BsmtFinType1 = ifelse(is.na(BsmtFinType1), "None", BsmtFinType1),
    BsmtFinSF1 = ifelse(is.na(BsmtFinSF1), 0, BsmtFinSF1),
    BsmtFinType2 = ifelse(is.na(BsmtFinType2), "None", BsmtFinType2),
    BsmtFinSF2 = ifelse(is.na(BsmtFinSF2), 0, BsmtFinSF2),
    BsmtUnfSF = ifelse(is.na(BsmtUnfSF), 0, BsmtUnfSF),
    TotalBsmtSF = ifelse(is.na(TotalBsmtSF), 0, TotalBsmtSF),
    LotFrontage = ifelse(is.na(LotFrontage), 0, LotFrontage),
    MasVnrType = ifelse(is.na(MasVnrType), "None", MasVnrType),
    MasVnrArea = ifelse(is.na(MasVnrArea), 0, MasVnrArea),
    FireplaceQu = ifelse(is.na(FireplaceQu), "None", FireplaceQu),
    Electrical = ifelse(is.na(Electrical), "None", Electrical)
    )

# Create a table with the remaining missing values
missing_count <- ames_subset %>% select(-SalePrice) %>% sapply(function(x) su
m(is.na(x)))
missing_table <- as.data.frame(missing_count) %>%
  filter(missing_count >= 1) %>%
  arrange(desc(missing_count))

kable(missing_table, caption = "Missing Value Counts (Descending Order)", ali
gn = "c")
```

Missing Value Counts (Descending Order)

	missing_count
MSZoning	4
Utilities	2
BsmtFullBath	2
BsmtHalfBath	2
Functional	2
Exterior1st	1
Exterior2nd	1
KitchenQual	1

	missing_count
SaleType	1

```
# Use mutate_at to replace NA with 0 in the specified range of columns
# Define a custom function
replace_na_custom <- function(x) {
  if (is.numeric(x)) {
    return(replace_na(x, 0))
  } else {
    return(replace_na(x, "None"))
  }
}
# Specify the columns you want to exclude
cols_to_exclude <- c( "SalePrice")

# Replace NA with 0 for numeric columns and "None" for non-numeric columns
# SalePrice from mutation since the testing data is in this data set
ames_subset <- ames_subset %>%
  mutate(across(.cols = -all_of(cols_to_exclude), .fns = replace_na_custom))

# Summarize the amount of remaining NA's by column
missing_count <- ames_subset %>% select(-SalePrice) %>% sapply(function(x) su
m(is.na(x)))
missing_table <- as.data.frame(missing_count) %>%
  filter(missing_count >= 1) %>%
  arrange(desc(missing_count))
kable(missing_table, caption = "Missing Value Counts (Descending Order)", ali
gn = "c")
```

Missing Value Counts (Descending Order)

missing_count

NOTE: There are few missing values left, most of which are in the testing data set.

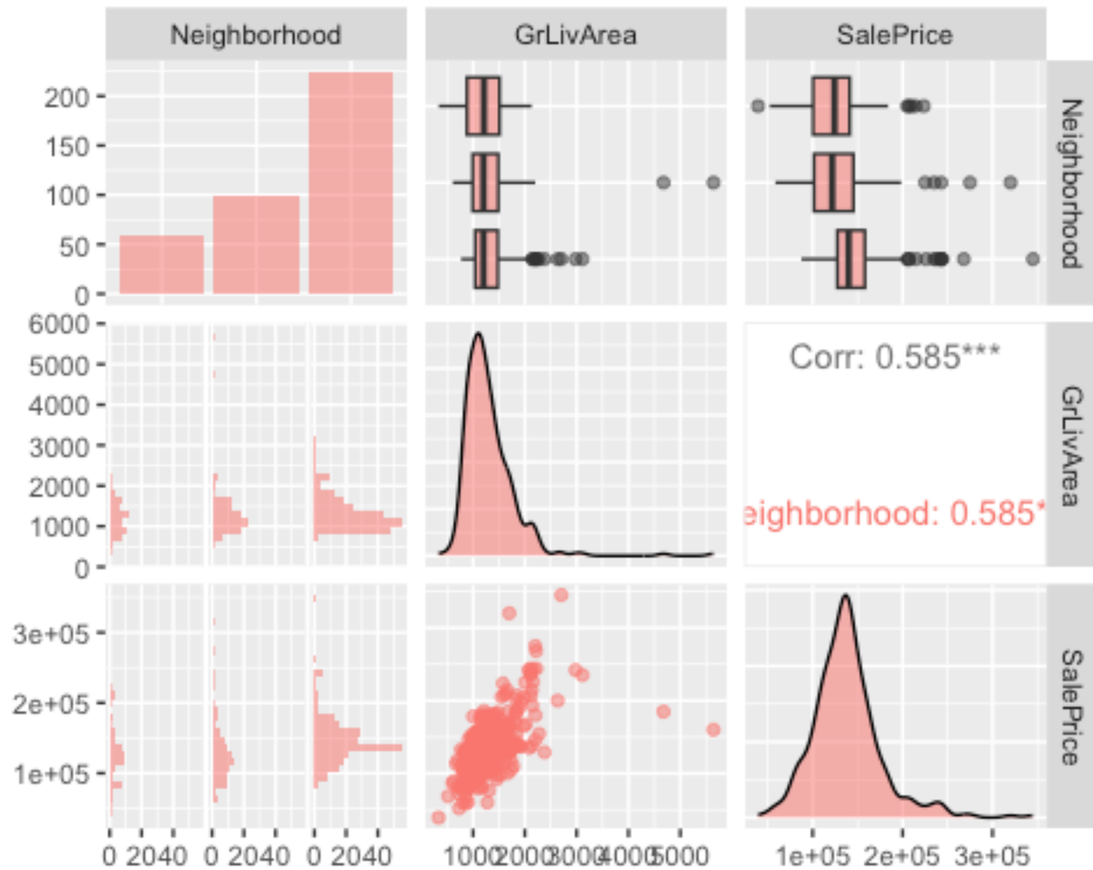
Analysis 1: Sale Price and Gross Living Area

Century 21 is interested if there is a correlation between the SalePrice of a house is its related to the square (GrLIvArea). In addition, determine if there is also a correlatoin between the SalePrice of a house and the neighborhood the house is located in. For this analysis, Centrury 21 is only insterested in their neighborhoods of NAmes, Edwards, and BrkSide.

Visualize Data

```
# Visualize dataset for SalePrice, GrLivArea by Neighborhood
century21 <- ames_subset %>% filter(train == "train") %>%
  select(Neighborhood, GrLivArea, SalePrice) %>%
  filter(Neighborhood == "NAmes" | Neighborhood == "BrkSide" | Neighborhood =
= "Edwards")
```

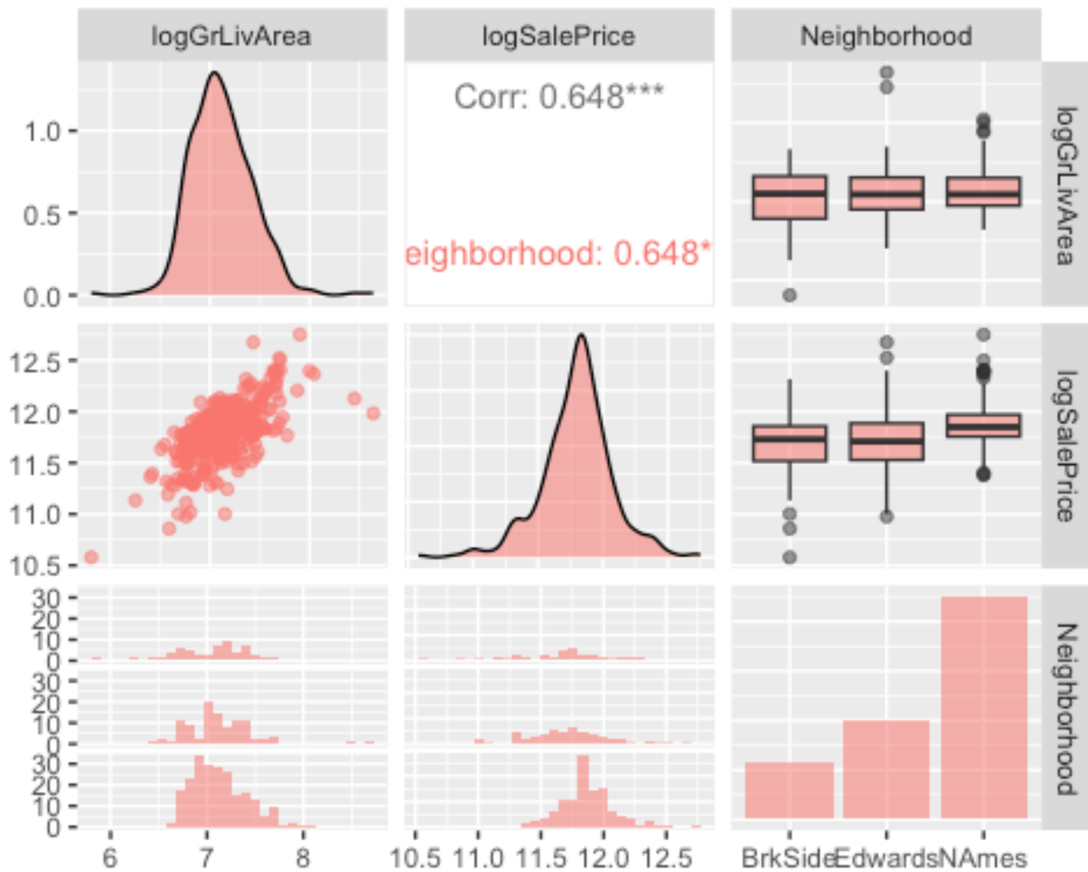
```r
##Inspect for missing values
colSums(is.na(century21))

## Neighborhood    GrLivArea    SalePrice
##            0            0            0

#Initial visualization
ggpairs(century21, mapping = aes(color = "Neighborhood", alpha = 0.5))
```



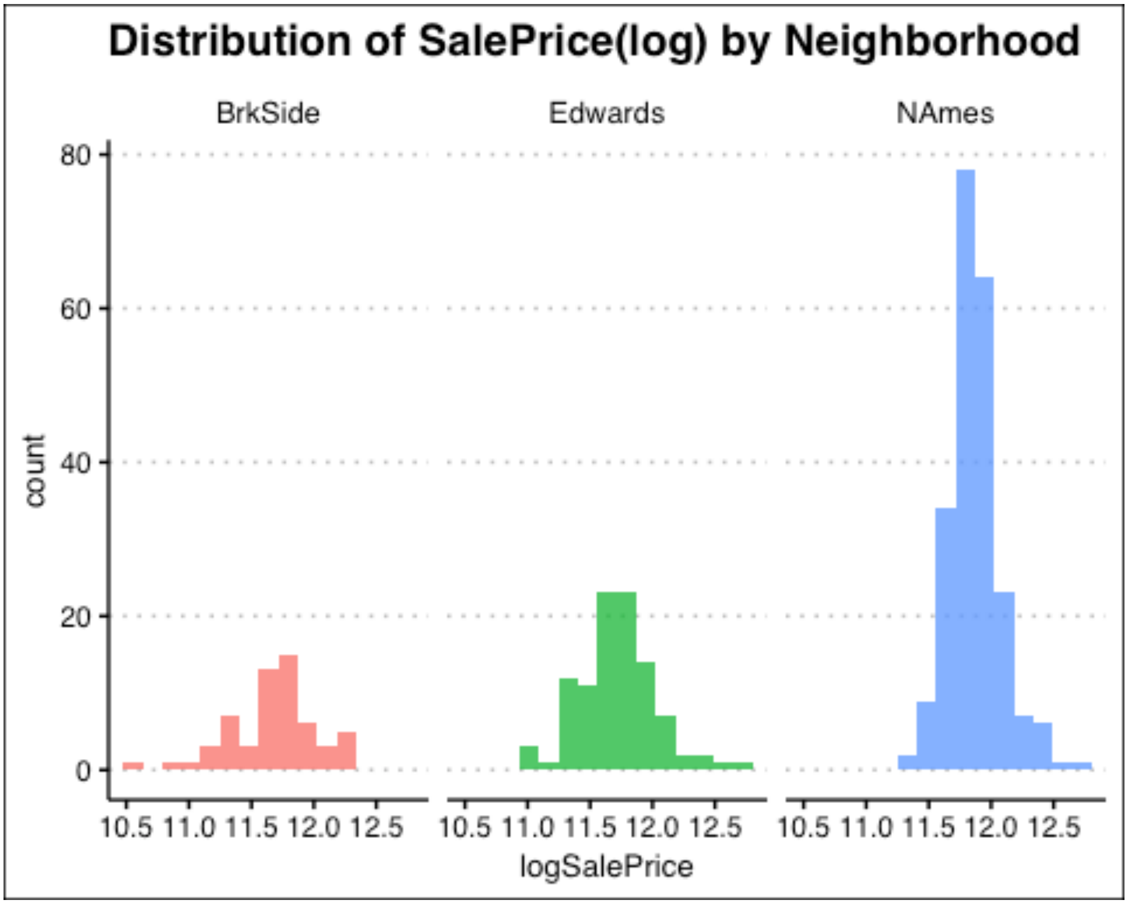
Check the Assumptions

Visually the observations do not show evidence of not being linear, however a log transformation may increase the correlarity.

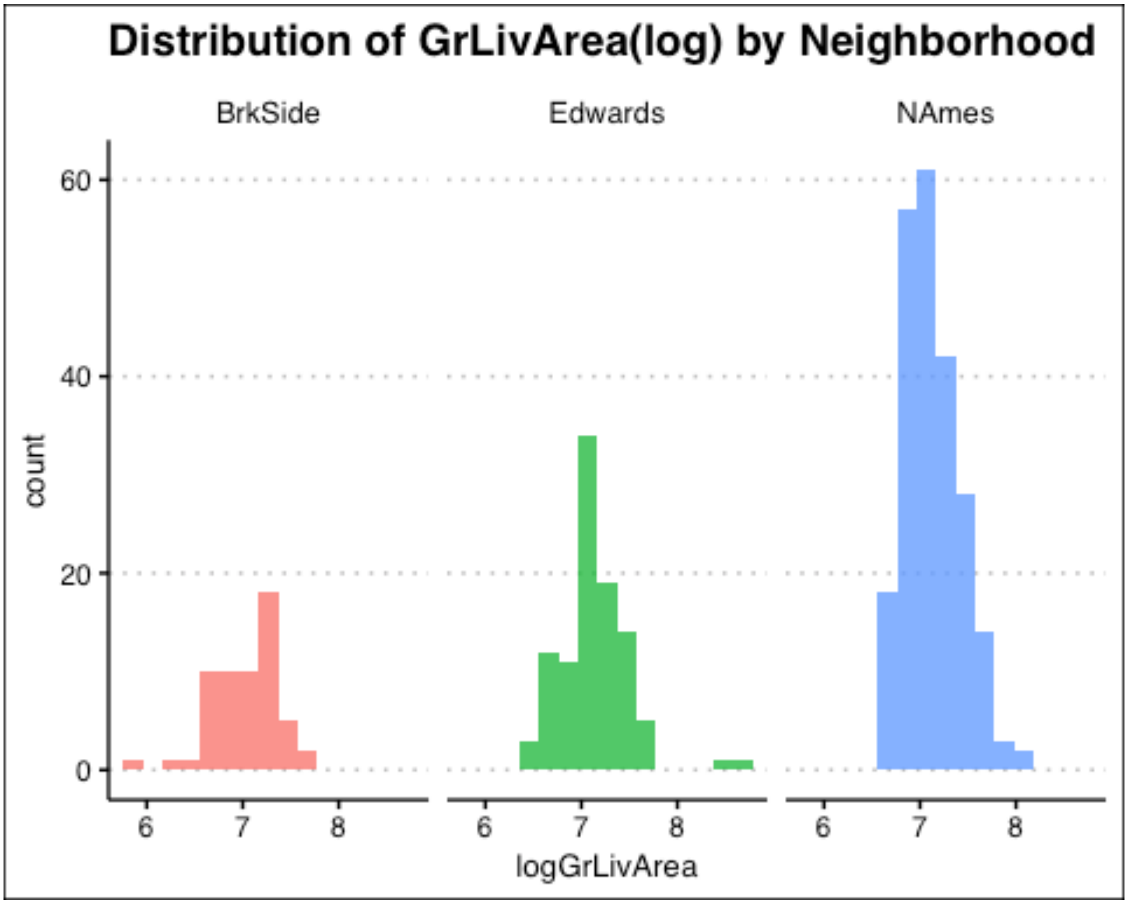
```r
# Visualize Log-Log Transformation
century21$logSalePrice <- log(century21$SalePrice)
century21$logGrLivArea <- log(century21$GrLivArea)

#Visualize only logged data by neighborhood
century21 %>%
  select(logGrLivArea,logSalePrice, Neighborhood ) %>%
  ggpairs( mapping=aes(color= "Neighborhood", alpha = 0.5))
```



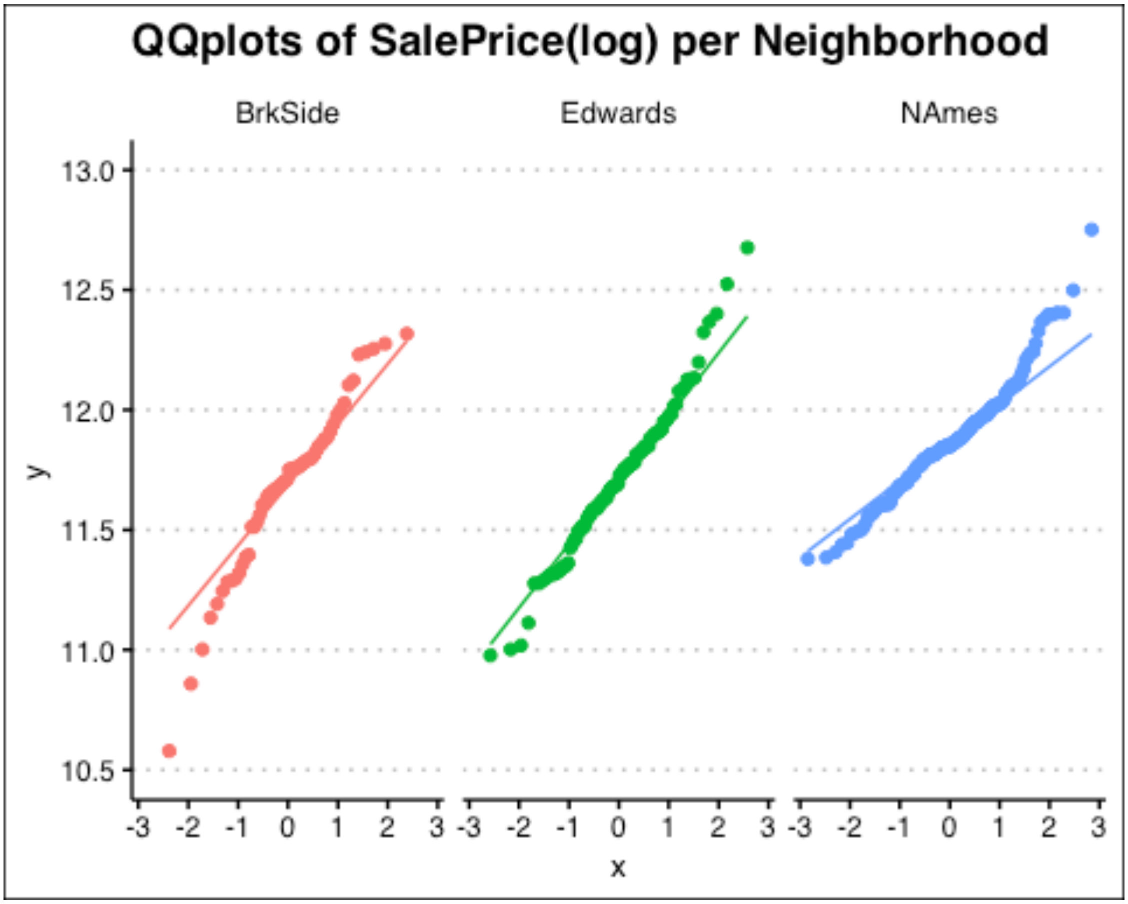

```r
ggplot(century21, aes(x = logSalePrice, fill = Neighborhood)) +
  geom_histogram(alpha = 0.8, position = "dodge", bins = 15) +
  ggtitle("Distribution of SalePrice(log) by Neighborhood") +
  facet_wrap(~Neighborhood) + theme_clean() + theme(legend.position = "none")
```



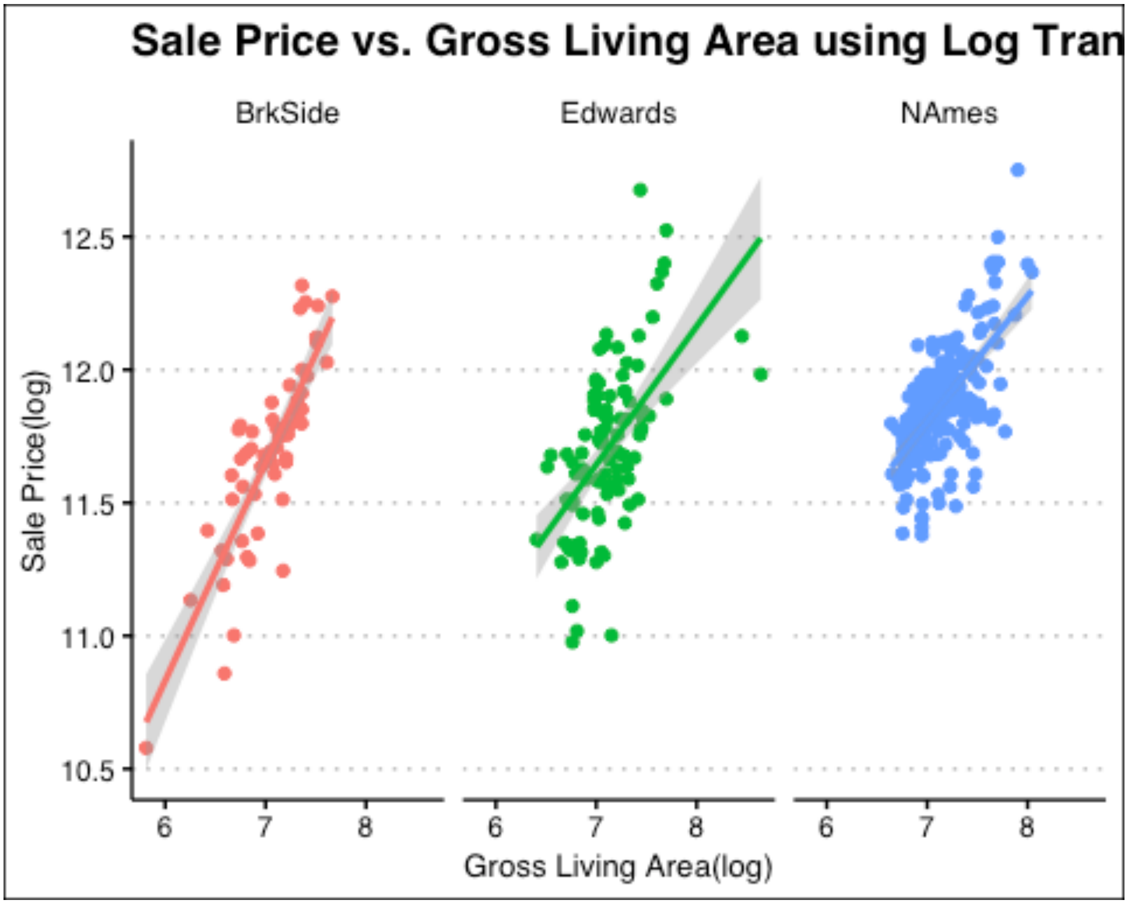

```r
ggplot(century21, aes(x = logGrLivArea, fill = Neighborhood)) +
  geom_histogram(alpha = 0.8, position = "dodge", bins = 15) +
  ggtitle("Distribution of GrLivArea(log) by Neighborhood") +
  facet_wrap(~Neighborhood) + theme_clean() + theme(legend.position = "none")
```




```r
# Visualize QQ plots
ggplot(century21, aes(sample = logSalePrice, color = Neighborhood)) +
  geom_qq() +
  geom_qq_line() +
  ggtitle("QQplots of SalePrice(log) per Neighborhood") +
  theme_clean() + theme(legend.position = "none") +
  facet_wrap(~Neighborhood) +
  scale_y_continuous(limits = c(10.5, 13), breaks = seq(10.5, 13, 0.5))
```




```
# Visualize Correlation with the linear model superposed
century21 %>% ggplot(aes(x = logGrLivArea, y = logSalePrice, color = Neighbor
hood))+
  geom_point()+
  geom_smooth(method = "lm") +
  facet_wrap(vars(Neighborhood)) +
  labs(
    title = "Sale Price vs. Gross Living Area using Log Transformation",
    x = "Gross Living Area(log)",
    y = "Sale Price(log)") + theme_clean() + theme(legend.position = "none")
```




```
#Check for equal variance
library(car)
leveneTest(century21$SalePrice, century21$GrLivArea) #Before

## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value Pr(>F)
## group 289  1.0604 0.3758
##        93

leveneTest(century21$logSalePrice, century21$logGrLivArea) #After

## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value Pr(>F)
## group 289  0.6793 0.9916
##        93
```

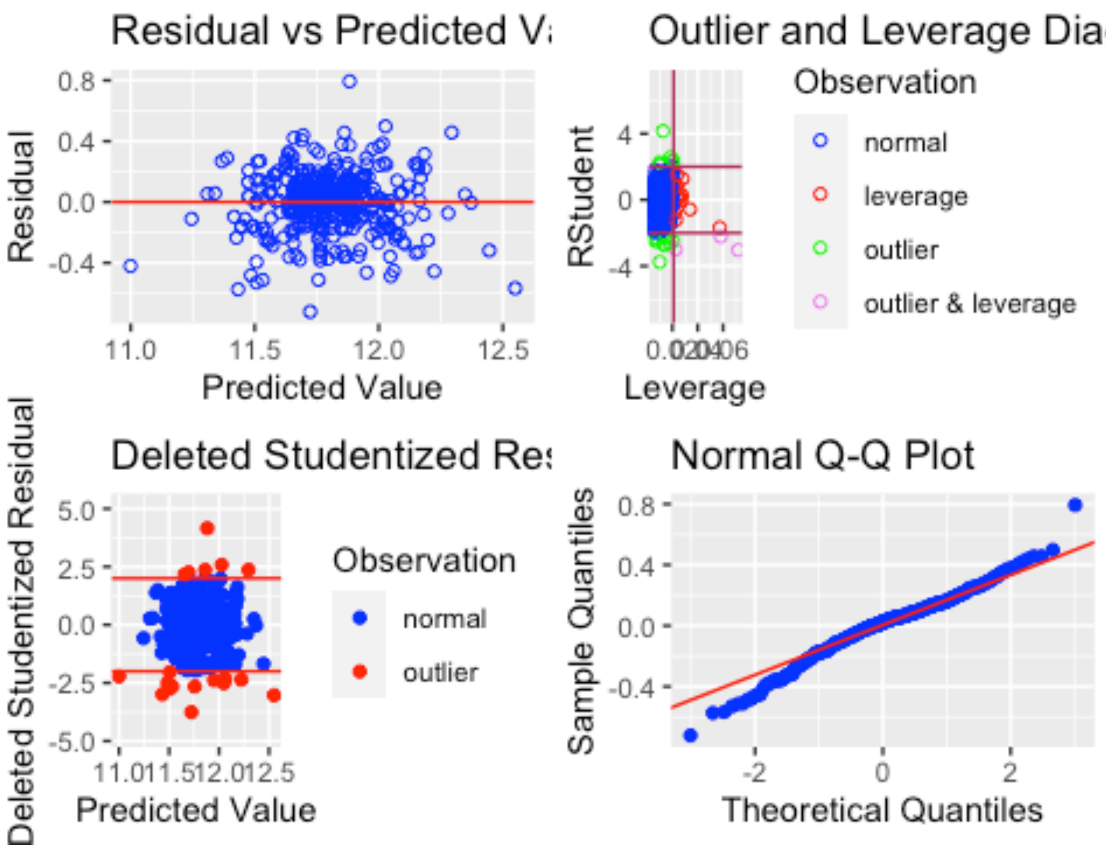
Assumptions: The Levene Test results imply no significant variance between the groups (homogeneity)

The distribution of both logGrLivArea and logSalePrice show evidence of normality, along with having the CLT apply due to number of observations.

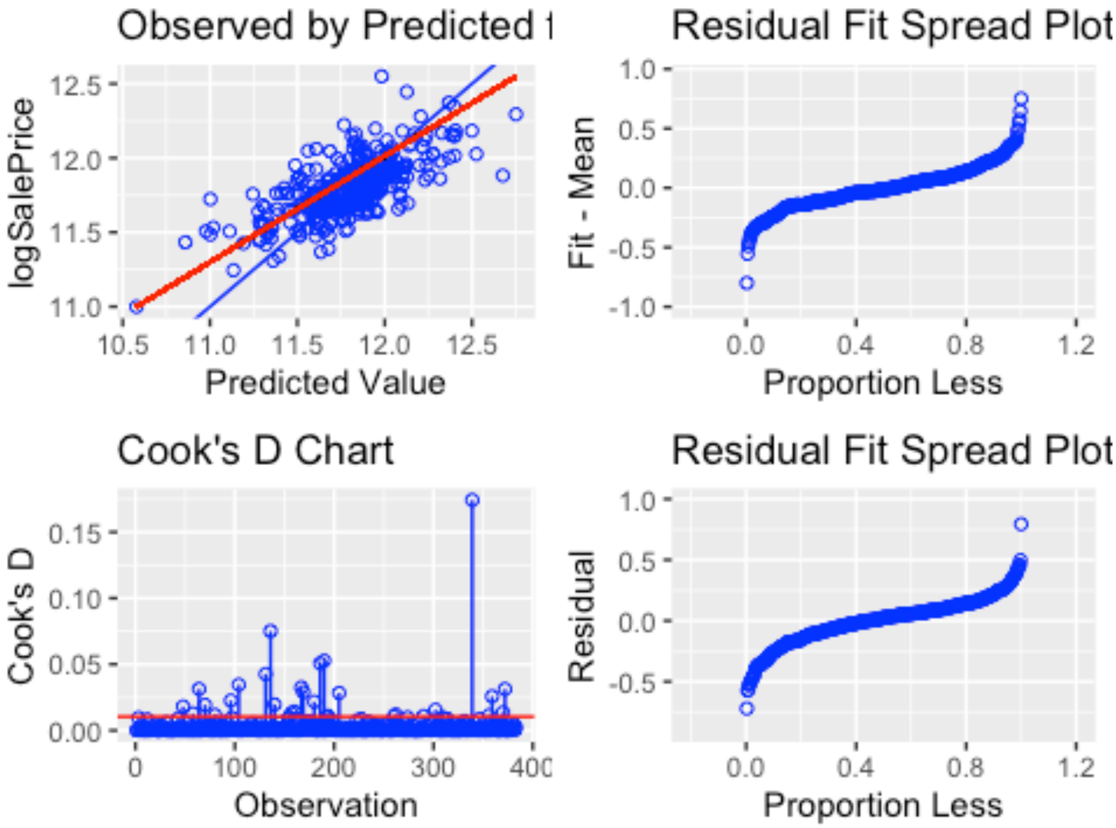
There is no reason to suggest that the variables are not independent.

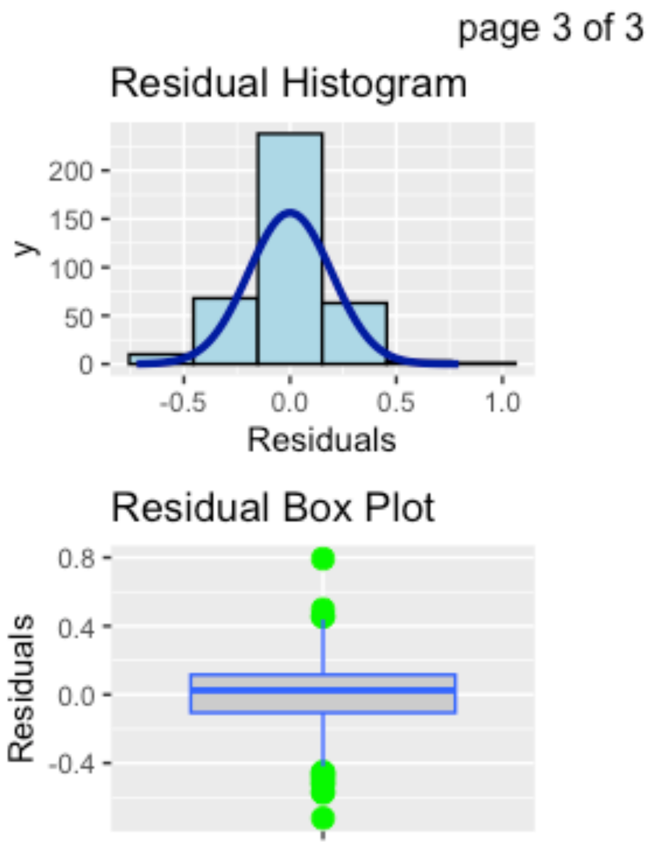
Influential Points

```r
#fit model 1 of log data
fit = lm(logSalePrice~logGrLivArea+factor(Neighborhood), data=century21)
summary(fit)

##
## Call:
## lm(formula = logSalePrice ~ logGrLivArea + factor(Neighborhood),
##     data = century21)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.72154 -0.10592  0.02469  0.11565  0.79364
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   7.76936    0.22919  33.900  < 2e-16 ***
## logGrLivArea                  0.55579    0.03237  17.171  < 2e-16 ***
## factor(Neighborhood)Edwards -0.02044    0.03252  -0.629     0.53
## factor(Neighborhood)NAmes    0.13279    0.02906   4.569 6.63e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1961 on 379 degrees of freedom
## Multiple R-squared:  0.4897, Adjusted R-squared:  0.4857
## F-statistic: 121.2 on 3 and 379 DF,  p-value: < 2.2e-16

#Check the residuals
# par(mfrow=c(2,2)) # Set up a 2x2 plot grid
# plot(fit, which = 1) # Residuals vs Fitted
# plot(fit, which = 2) # Normal Q-Q
# plot(fit, which = 3) # Scale-Location
# plot(fit, which = 4) # Cook's distance
ols_plot_diagnostics(fit)
```



Residual vs Predicted V
Residual
Predicted Value
Outlier and Leverage Diag
RStudent
Leverage
Observation
normal
leverage
outlier
outlier & leverage
Deleted Studentized Res
Deleted Studentized Residual
Predicted Value
Observation
normal
outlier
Normal Q-Q Plot
Sample Quantiles
Theoretical Quantiles

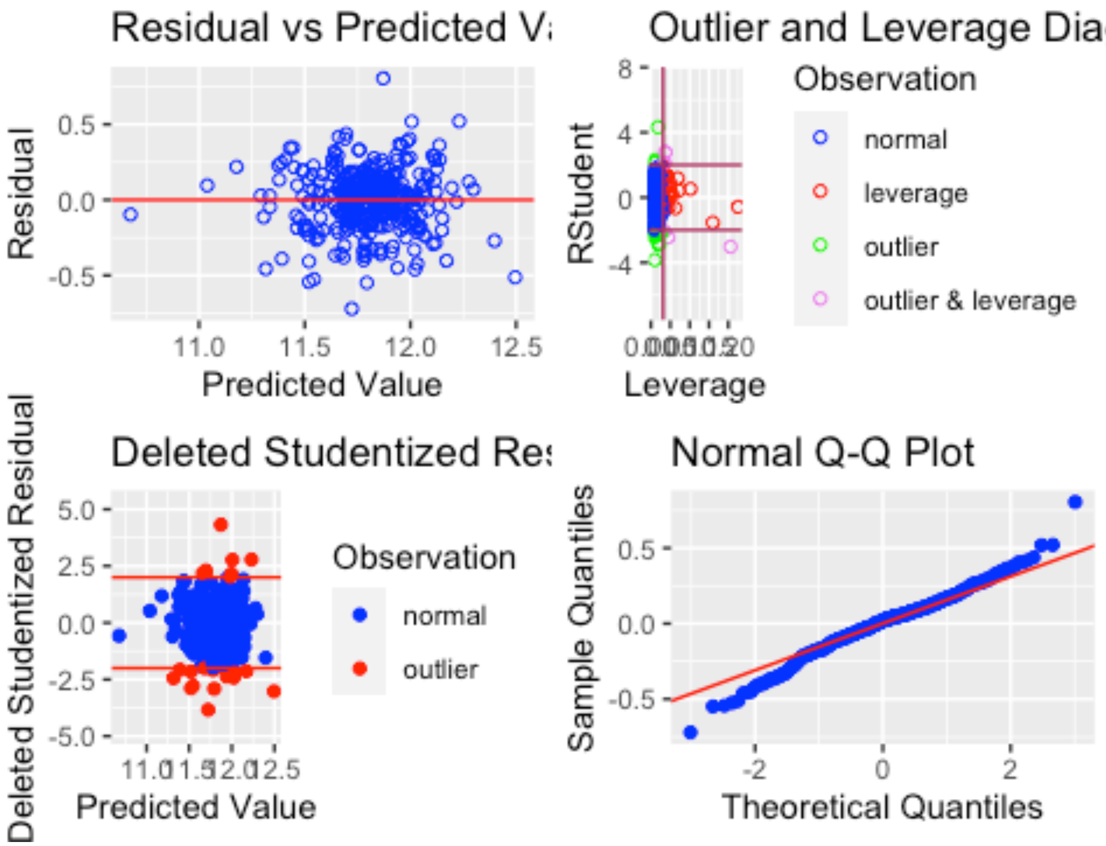

Observed by Predicted f
logSalePrice
12.5
12.0
11.5
11.0
10.5
11.0
11.5
12.0
12.5
Predicted Value
Residual Fit Spread Plot
Fit - Mean
1.0
0.5
0.0
-0.5
-1.0
0.0
0.4
0.8
1.2
Proportion Less
Cook's D Chart
Cook's D
0.15
0.10
0.05
0.00
0
100
200
300
400
Observation
Residual Fit Spread Plot
Residual
1.0
0.5
0.0
-0.5
0.0
0.4
0.8
1.2
Proportion Less

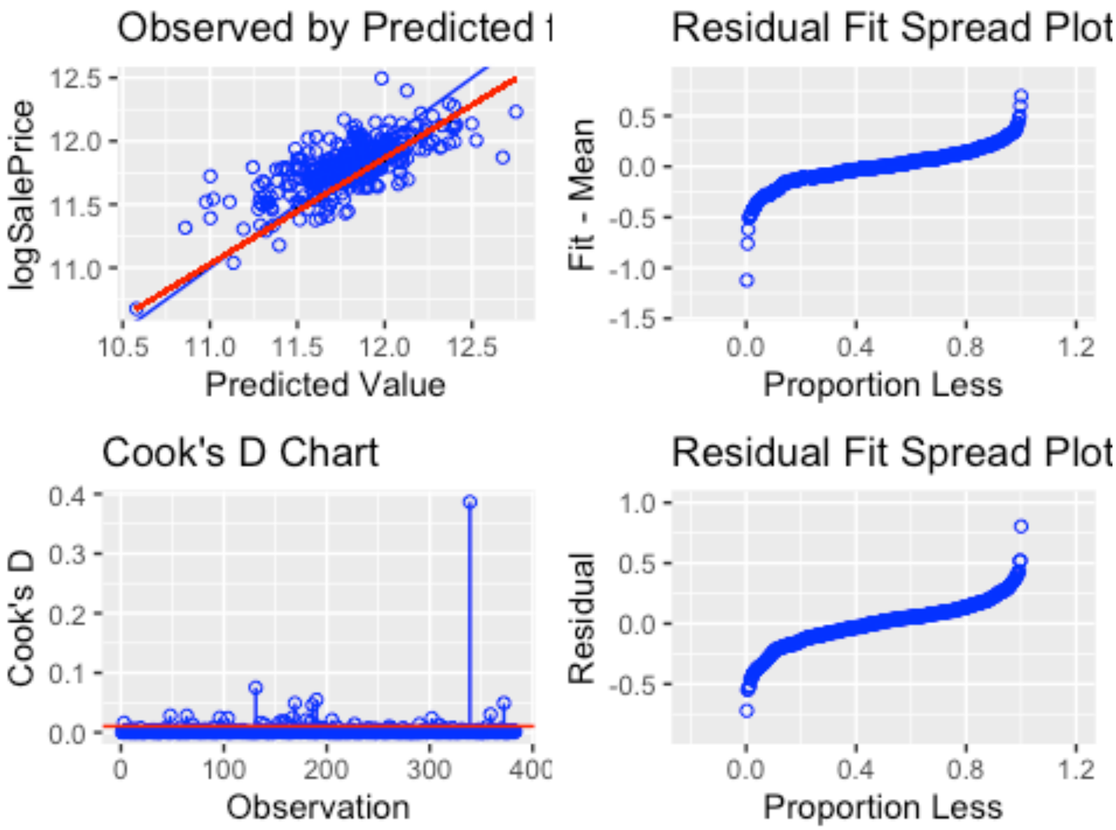


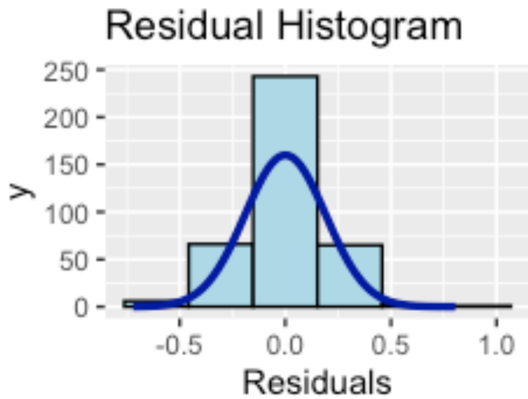

```
#fit model 2 of log data / model to account for different intercepts and diff
erent slopes
fit2 = lm(logSalePrice~logGrLivArea*factor(Neighborhood), data=century21)
summary(fit2)

##
## Call:
## lm(formula = logSalePrice ~ logGrLivArea * factor(Neighborhood),
##     data = century21)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.72080 -0.10353  0.02184  0.10586  0.80470
##
## Coefficients:
##                                          Estimate Std. Error t value Pr(>|
t|)
## (Intercept)                               5.91292    0.50459  11.718  < 2e
-16
## logGrLivArea                              0.81965    0.07163  11.443  < 2e
-16
## factor(Neighborhood)Edwards               2.09359    0.64589   3.241   0.0
013
```

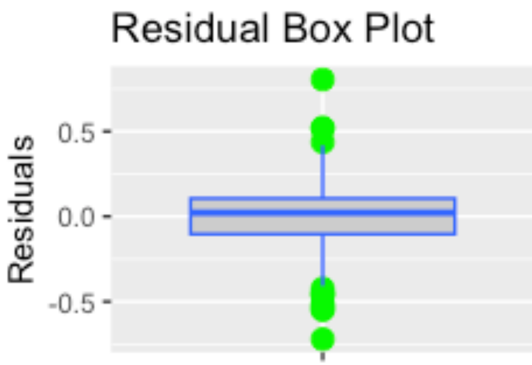
```
## factor(Neighborhood)NAmes                  2.57981    0.59988   4.301 2.17e
-05
## logGrLivArea:factor(Neighborhood)Edwards -0.29998    0.09122  -3.289   0.0
011
## logGrLivArea:factor(Neighborhood)NAmes   -0.34662    0.08482  -4.087 5.35e
-05
##
## (Intercept)                              ***
## logGrLivArea                             ***
## factor(Neighborhood)Edwards              **
## factor(Neighborhood)NAmes                ***
## logGrLivArea:factor(Neighborhood)Edwards **
## logGrLivArea:factor(Neighborhood)NAmes   ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1923 on 377 degrees of freedom
## Multiple R-squared:  0.5121, Adjusted R-squared:  0.5056
## F-statistic: 79.14 on 5 and 377 DF,  p-value: < 2.2e-16
```

```r
#Check the residuals
# par(mfrow=c(2,2)) # Set up a 2x2 plot grid
# plot(fit2, which = 1) # Residuals vs Fitted
# plot(fit2, which = 2) # Normal Q-Q
# plot(fit2, which = 3) # Scale-Location
# plot(fit2, which = 4) # Cook's distance


ols_plot_diagnostics(fit2)
```

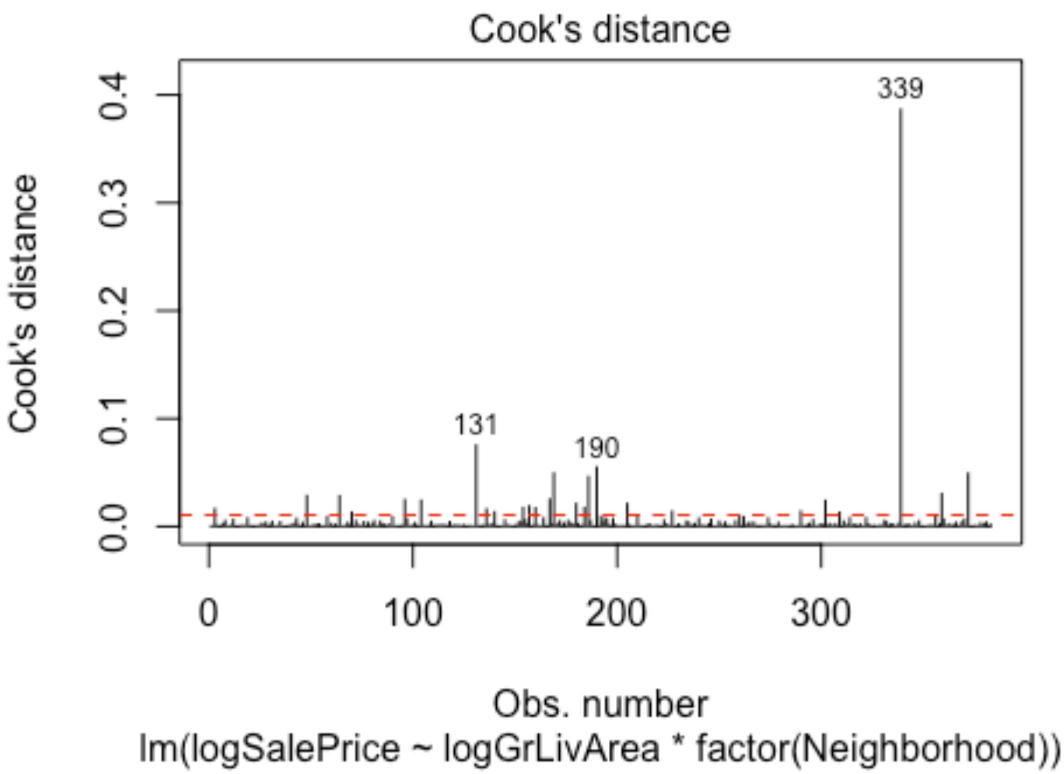

Residual vs Predicted V
Residual
0.5
0.0
-0.5
11.0
11.5
12.0
12.5
Predicted Value
Outlier and Leverage Diag
RStudent
8
4
0
-4
Leverage
Observation
normal
leverage
outlier
outlier & leverage
Deleted Studentized Res
Deleted Studentized Residual
5.0
2.5
0.0
-2.5
-5.0
Predicted Value
Observation
normal
outlier
Normal Q-Q Plot
Sample Quantiles
0.5
0.0
-0.5
-2
0
2
Theoretical Quantiles


Observed by Predicted f
logSalePrice
Predicted Value
Residual Fit Spread Plot
Fit - Mean
Proportion Less
Cook's D Chart
Cook's D
Observation
Residual Fit Spread Plot
Residual
Proportion Less

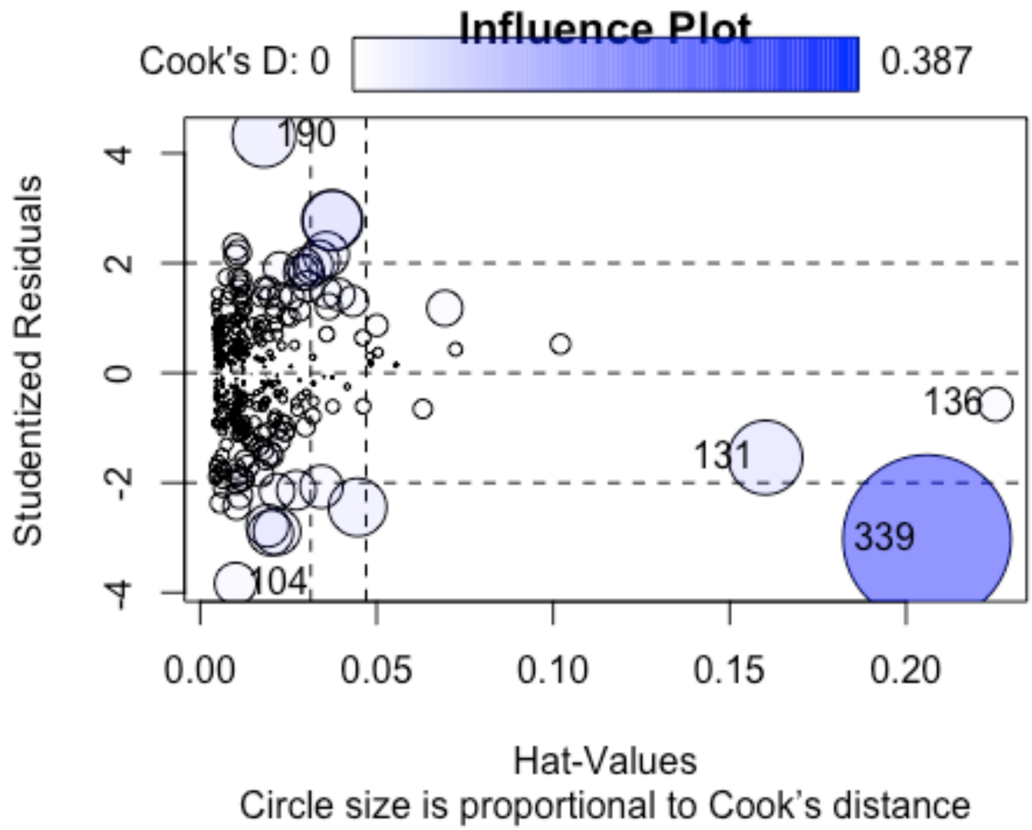





```
#Cooks D Plot
cutoff <- 4/(nrow(century21)-length(fit2$coefficients)-2)
plot(fit2, which=4, cook.levels=cutoff)
abline(h=cutoff, lty=2, col="red")
```

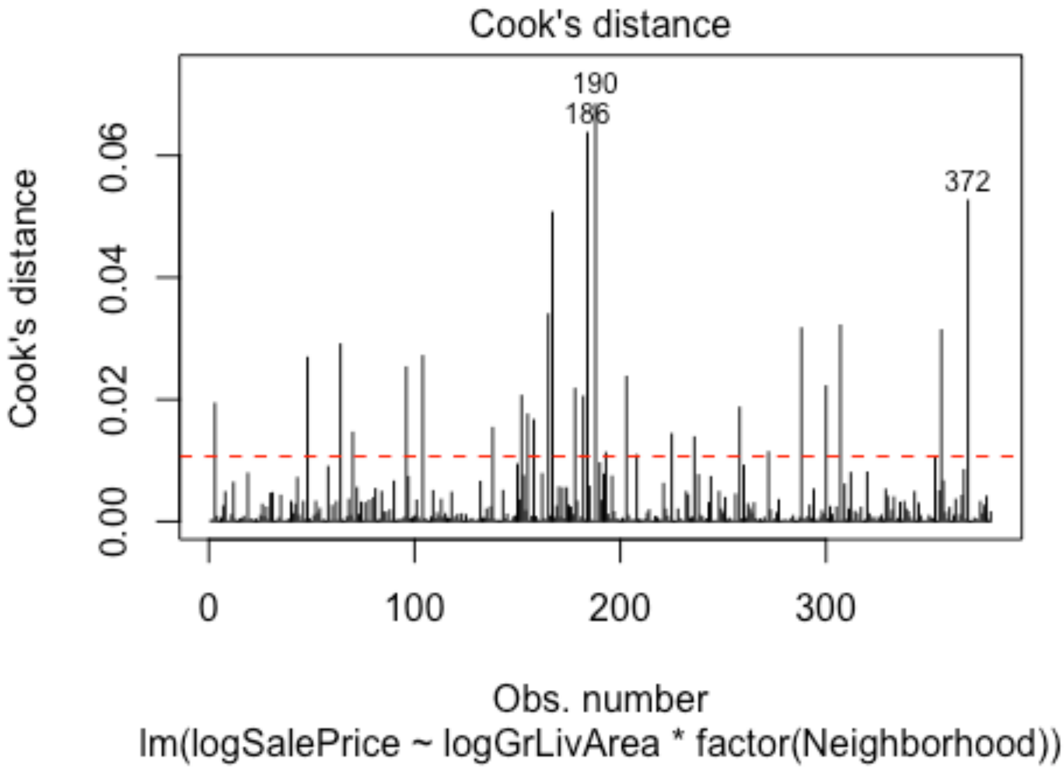



```
#Influence Plot
influencePlot(fit2, id.method="identify", main="Influence Plot", sub="Circle
size is proportional to Cook’s distance")
```

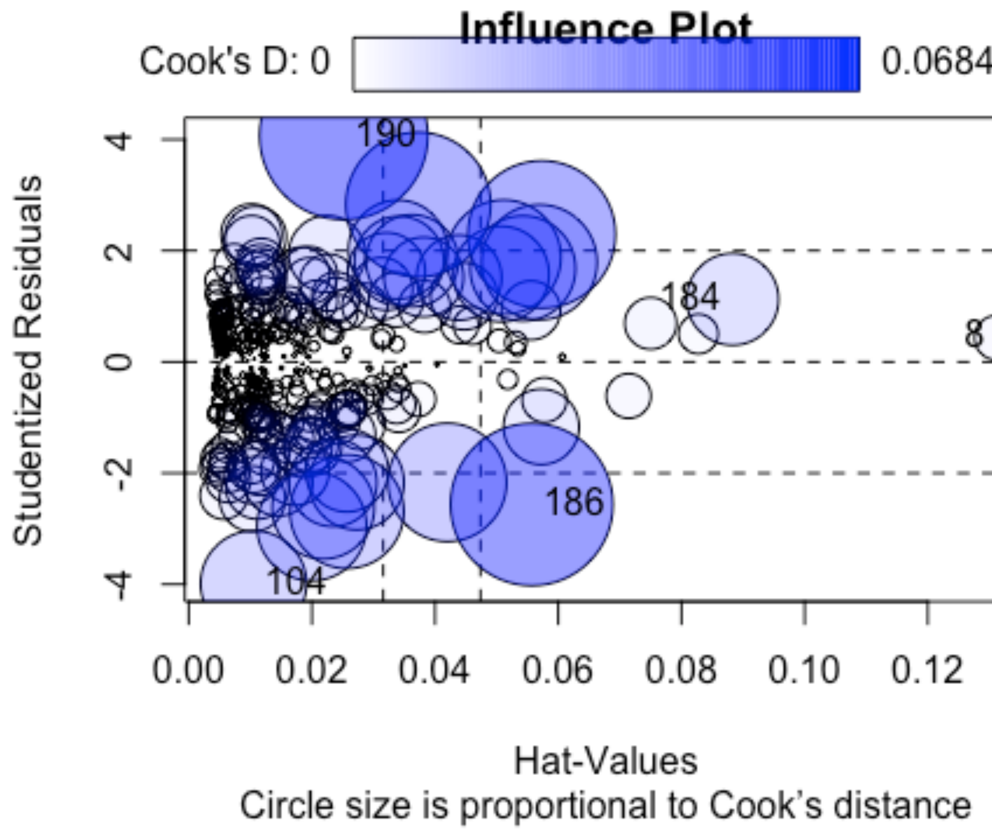



```
##        StudRes        Hat      CookD
## 104 -3.8356341 0.01003581 0.02398505
## 131 -1.5409476 0.16011659 0.07517290
## 136 -0.5728569 0.22534912 0.01593916
## 190  4.3213329 0.01808341 0.05475111
## 339 -3.0229829 0.20590279 0.38657449

#Outliers Addressed/Observation 339 influential point/outlier removed.
outliers_to_remove <- c(131,136,339)
century21_rmOutliers <- century21[-outliers_to_remove,]
str(century21)

## 'data.frame':    383 obs. of  5 variables:
##  $ Neighborhood: chr  "BrkSide" "NAmes" "BrkSide" "NAmes" ...
##  $ GrLivArea   : int  1077 1253 854 1004 1339 900 1600 520 1700 1297 ...
##  $ SalePrice   : int  118000 157000 132000 149000 139000 134800 207500 685
00 165500 153000 ...
##  $ logSalePrice: num  11.7 12 11.8 11.9 11.8 ...
##  $ logGrLivArea: num  6.98 7.13 6.75 6.91 7.2 ...

fit3 = lm(logSalePrice~logGrLivArea*factor(Neighborhood), data=century21_rmOu
tliers)
summary(fit3)
```

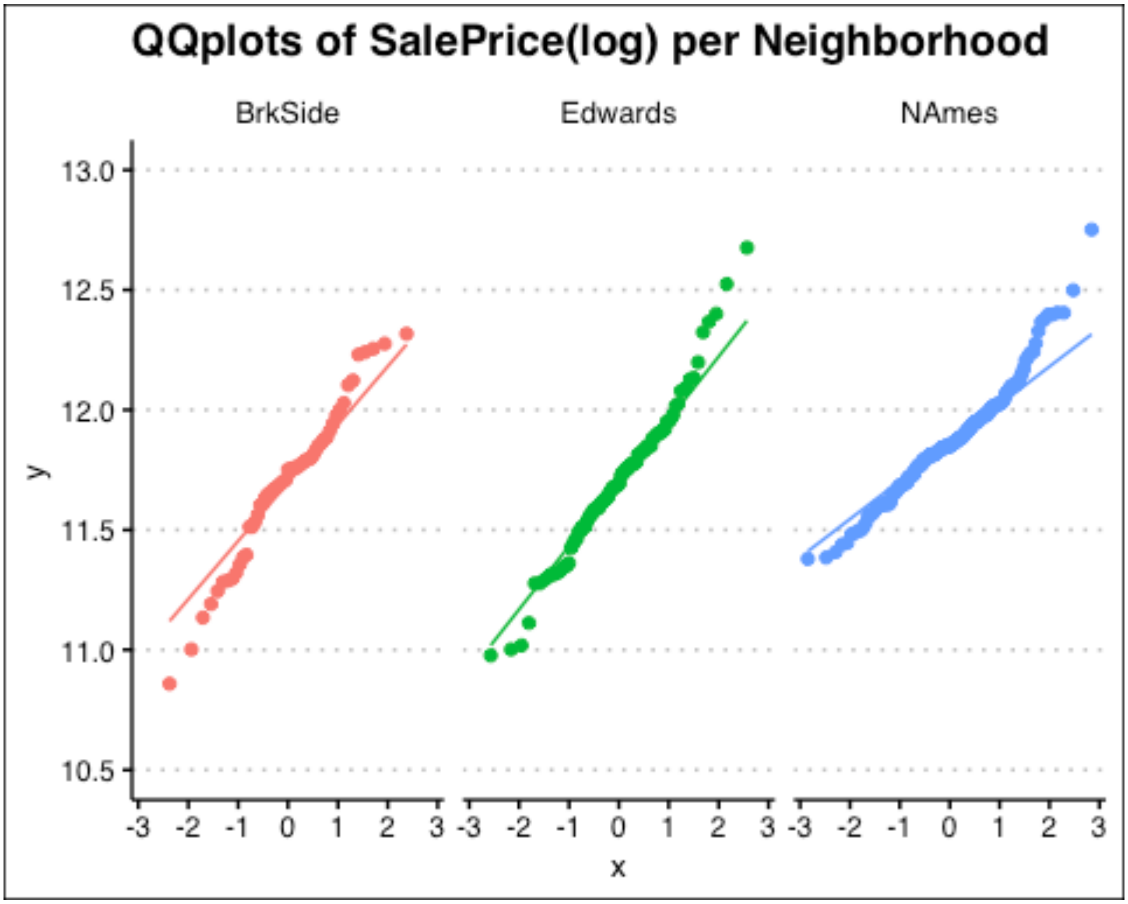
```
##
## Call:
## lm(formula = logSalePrice ~ logGrLivArea * factor(Neighborhood),
##     data = century21_rmOutliers)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.73636 -0.10922  0.02052  0.10528  0.74523
##
## Coefficients:
##                                          Estimate Std. Error t value Pr(>|
t|)
## (Intercept)                               6.06485    0.56120  10.807  < 2e
-16
## logGrLivArea                              0.79836    0.07944  10.050  < 2e
-16
## factor(Neighborhood)Edwards               0.85824    0.74594   1.151 0.250
652
## factor(Neighborhood)NAmes                 2.42788    0.64574   3.760 0.000
197
## logGrLivArea:factor(Neighborhood)Edwards -0.12502    0.10531  -1.187 0.235
924
## logGrLivArea:factor(Neighborhood)NAmes   -0.32534    0.09117  -3.568 0.000
406
##
## (Intercept)                              ***
## logGrLivArea                             ***
## factor(Neighborhood)Edwards
## factor(Neighborhood)NAmes                ***
## logGrLivArea:factor(Neighborhood)Edwards
## logGrLivArea:factor(Neighborhood)NAmes   ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1893 on 374 degrees of freedom
## Multiple R-squared:  0.5023, Adjusted R-squared:  0.4956
## F-statistic: 75.49 on 5 and 374 DF,  p-value: < 2.2e-16

#Check the residuals
# par(mfrow=c(2,2)) # Set up a 2x2 plot grid
# plot(fit3, which = 1) # Residuals vs Fitted
# plot(fit3, which = 2) # Normal Q-Q
# plot(fit3, which = 3) # Scale-Location
# plot(fit3, which = 4) # Cook's distance
# ols_plot_diagnostics(fit3)

#Cooks D Plot
cutoff <- 4/(nrow(century21)-length(fit3$coefficients)-2)
plot(fit3, which=4, cook.levels=cutoff)
abline(h=cutoff, lty=2, col="red")
```



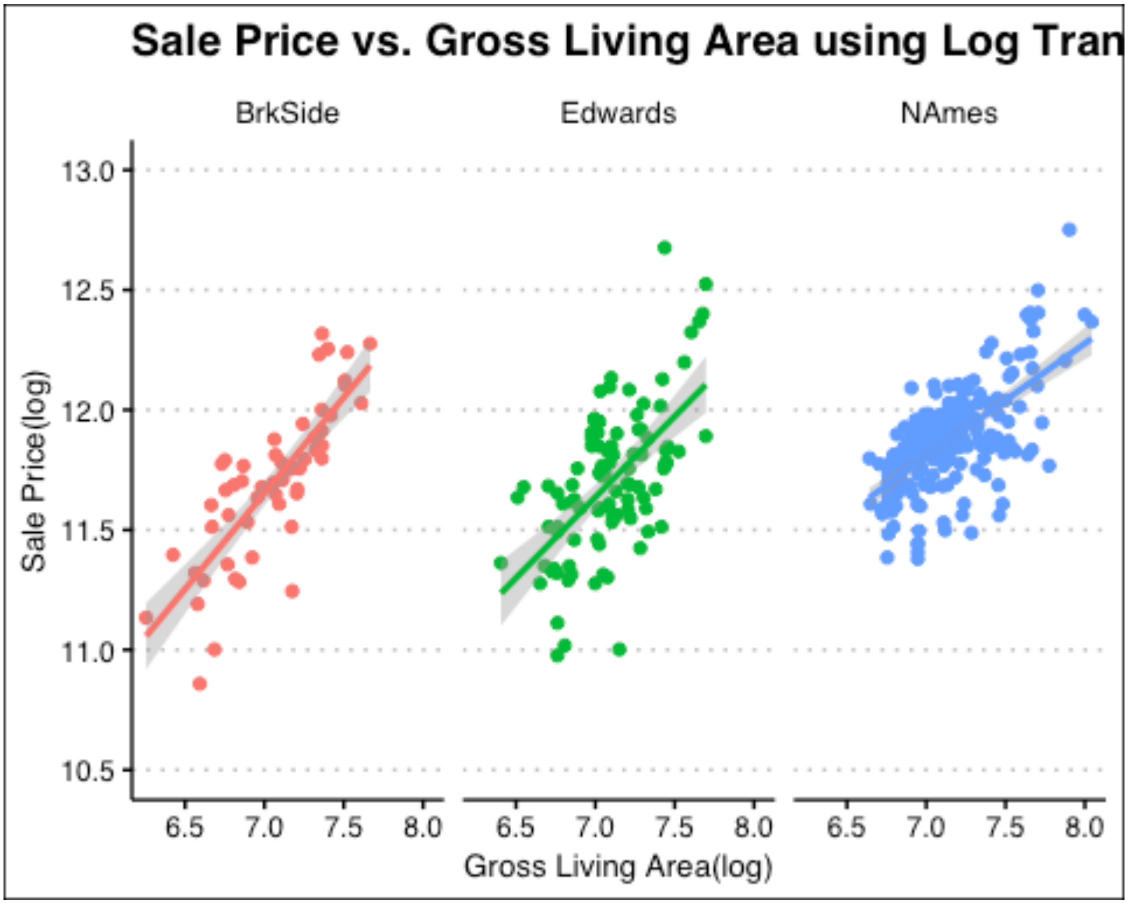

```
#Influence Plot
influencePlot(fit3, id.method="identify", main="Influence Plot", sub="Circle size is proportional to Cook’s distance")
```




```
##         StudRes        Hat       CookD
## 8    0.4354319 0.13117263 0.004781245
## 104 -3.9870454 0.01052703 0.027107566
## 184  1.1259461 0.08830066 0.020449631
## 186 -2.5647588 0.05569780 0.063714664
## 190  4.0687189 0.02516380 0.068377320

#QQ plot with removed outliers
ggplot(century21_rmOutliers, aes(sample = logSalePrice, color = Neighborhood)
) +
  geom_qq() +
  geom_qq_line() +
  ggtitle("QQplots of SalePrice(log) per Neighborhood") +
  theme_clean() + theme(legend.position = "none") +
  facet_wrap(~Neighborhood) +
  scale_y_continuous(limits = c(10.5, 13), breaks = seq(10.5, 13, 0.5))
```




```
# Visualize Correlation with the linear model superposed with removed outlier
s
ggplot(century21_rmOutliers, aes(x = logGrLivArea, y = logSalePrice, color =
Neighborhood))+
  geom_point()+
  geom_smooth(method = "lm") +
  facet_wrap(vars(Neighborhood)) +
  labs(
    title = "Sale Price vs. Gross Living Area using Log Transformation",
    x = "Gross Living Area(log)",
    y = "Sale Price(log)") + theme_clean() + theme(legend.position = "none")
+
    scale_y_continuous(limits = c(10.5, 13), breaks = seq(10.5, 13, 0.5))
```



####Linear Regression Model

```
fit <- lm(logSalePrice ~ logGrLivArea * Neighborhood, data=century21_rmOutlie
rs)
summary(fit)
```

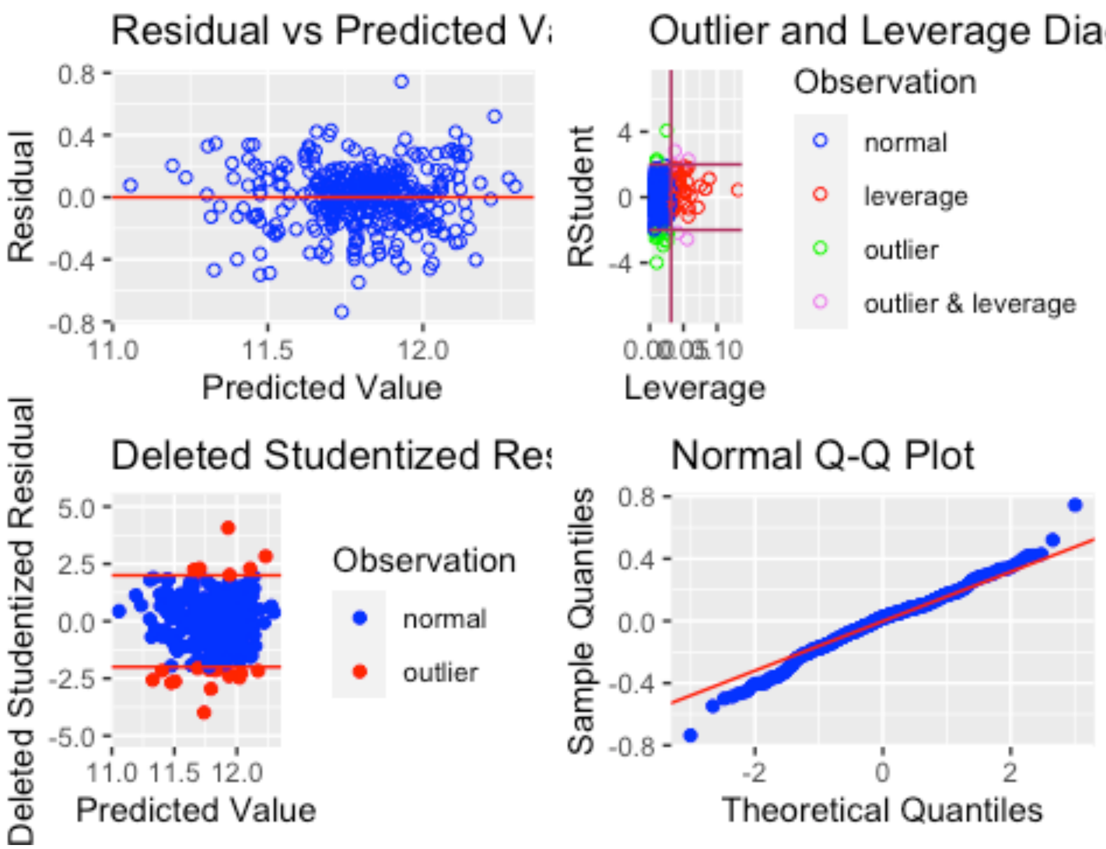
```
##
## Call:
## lm(formula = logSalePrice ~ logGrLivArea * Neighborhood, data = century21_
rmOutliers)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.73636 -0.10922  0.02052  0.10528  0.74523
##
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        6.06485    0.56120  10.807  < 2e-16 ***
## logGrLivArea                       0.79836    0.07944  10.050  < 2e-16 ***
## NeighborhoodEdwards                0.85824    0.74594   1.151 0.250652
## NeighborhoodNAmes                  2.42788    0.64574   3.760 0.000197 ***
## logGrLivArea:NeighborhoodEdwards -0.12502    0.10531  -1.187 0.235924
## logGrLivArea:NeighborhoodNAmes   -0.32534    0.09117  -3.568 0.000406 ***
## ---
```

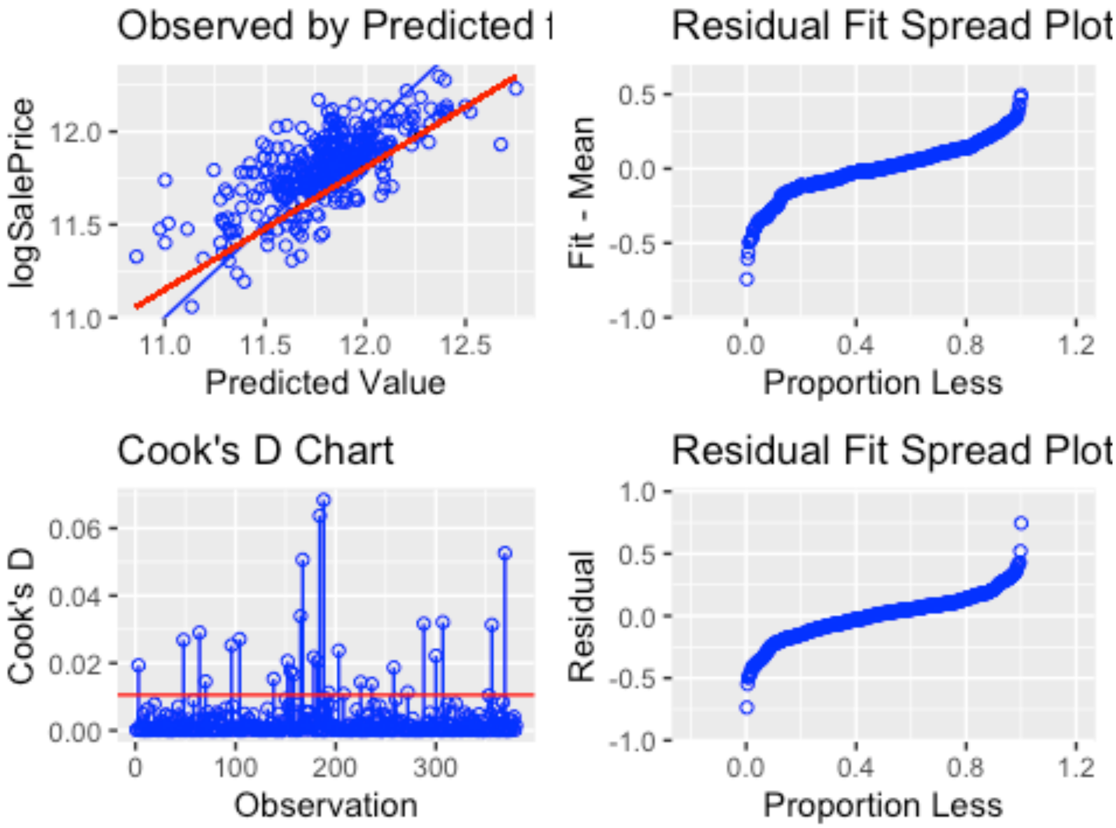
```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1893 on 374 degrees of freedom
## Multiple R-squared:  0.5023, Adjusted R-squared:  0.4956
## F-statistic: 75.49 on 5 and 374 DF,  p-value: < 2.2e-16
```

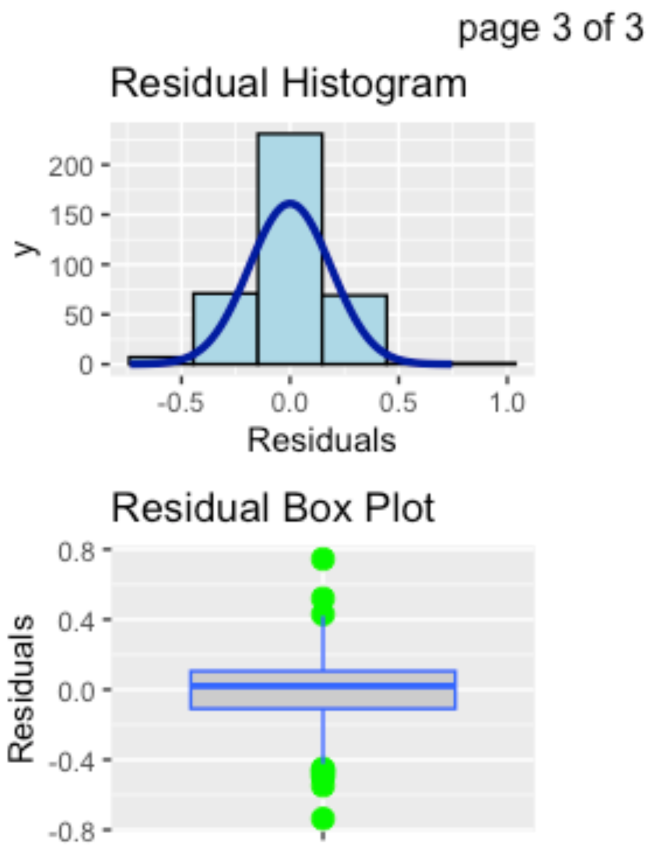
```
model1<- train(logSalePrice ~ logGrLivArea * Neighborhood,
               data=century21_rmOutliers,
               trControl= trainControl(method = "LOOCV"),
               method="lm")
model1
```

```
## Linear Regression
##
## 380 samples
##   2 predictor
##
## No pre-processing
## Resampling: Leave-One-Out Cross-Validation
## Summary of sample sizes: 379, 379, 379, 379, 379, 379, ...
## Resampling results:
##
##   RMSE       Rsquared  MAE
##   0.1915976  0.482313  0.1465014
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
# RMSE       Rsquared   MAE
# 0.1953044  0.4890223  0.1471398

ols_plot_diagnostics(fit) #plots of model
```


Residual vs Predicted V
Residual
Predicted Value
Outlier and Leverage Diag
RStudent
Leverage
Observation
normal
leverage
outlier
outlier & leverage
Deleted Studentized Res
Deleted Studentized Residual
Predicted Value
Observation
normal
outlier
Normal Q-Q Plot
Sample Quantiles
Theoretical Quantiles


Observed by Predicted f
logSalePrice
Predicted Value
Residual Fit Spread Plot
Fit - Mean
Proportion Less
Cook's D Chart
Cook's D
Observation
Residual Fit Spread Plot
Residual
Proportion Less




```
# vcov(fit)
```

####FitAll

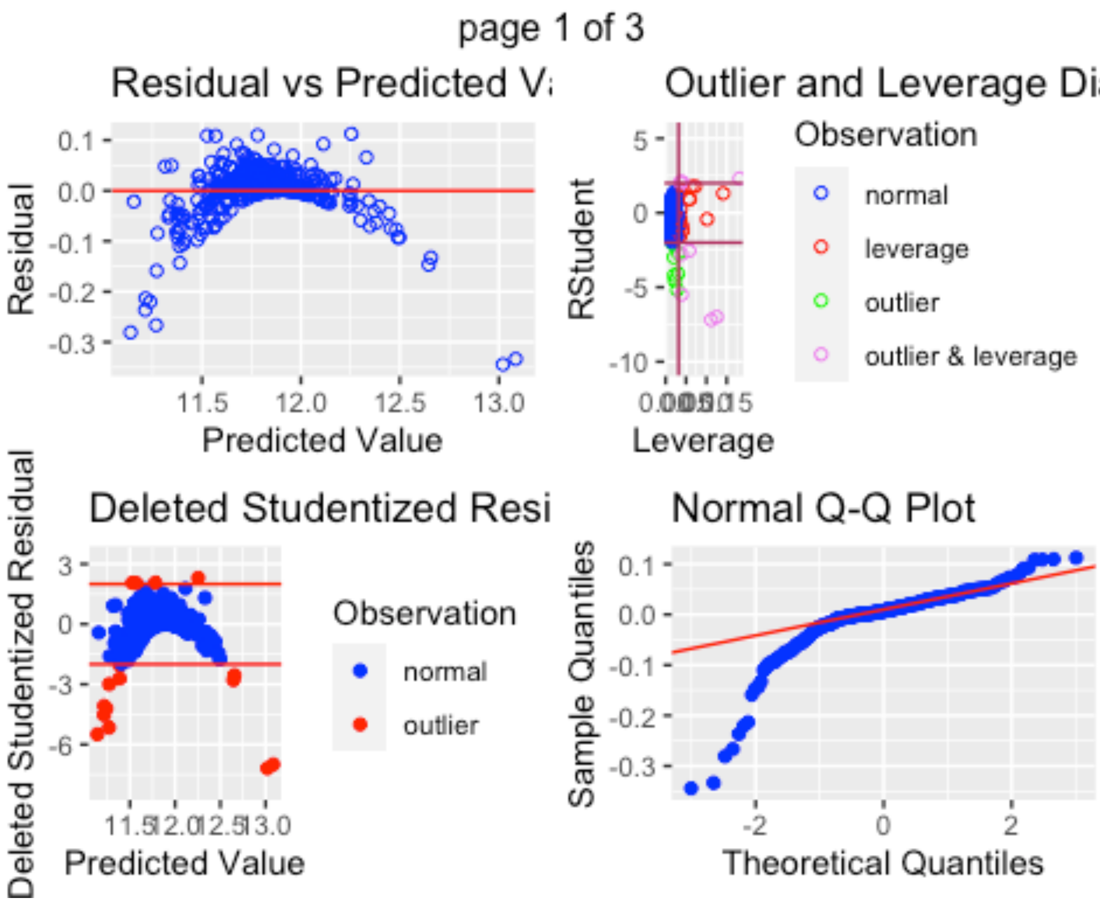
```
fit_all <- lm(logSalePrice ~ ., data = century21_rmOutliers)
summary(fit_all)

##
## Call:
## lm(formula = logSalePrice ~ ., data = century21_rmOutliers)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.34435 -0.00732  0.00718  0.02749  0.11245
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            7.638e+00  3.028e-01  25.226  < 2e-16 ***
## NeighborhoodEdwards -7.067e-03  9.052e-03  -0.781 0.435443
## NeighborhoodNAmes    3.101e-02  8.201e-03   3.782 0.000181 ***
## GrLivArea           -3.831e-04  3.629e-05 -10.557  < 2e-16 ***
## SalePrice            6.788e-06  1.025e-07  66.192  < 2e-16 ***
## logGrLivArea         5.201e-01  4.891e-02  10.634  < 2e-16 ***
## ---
```

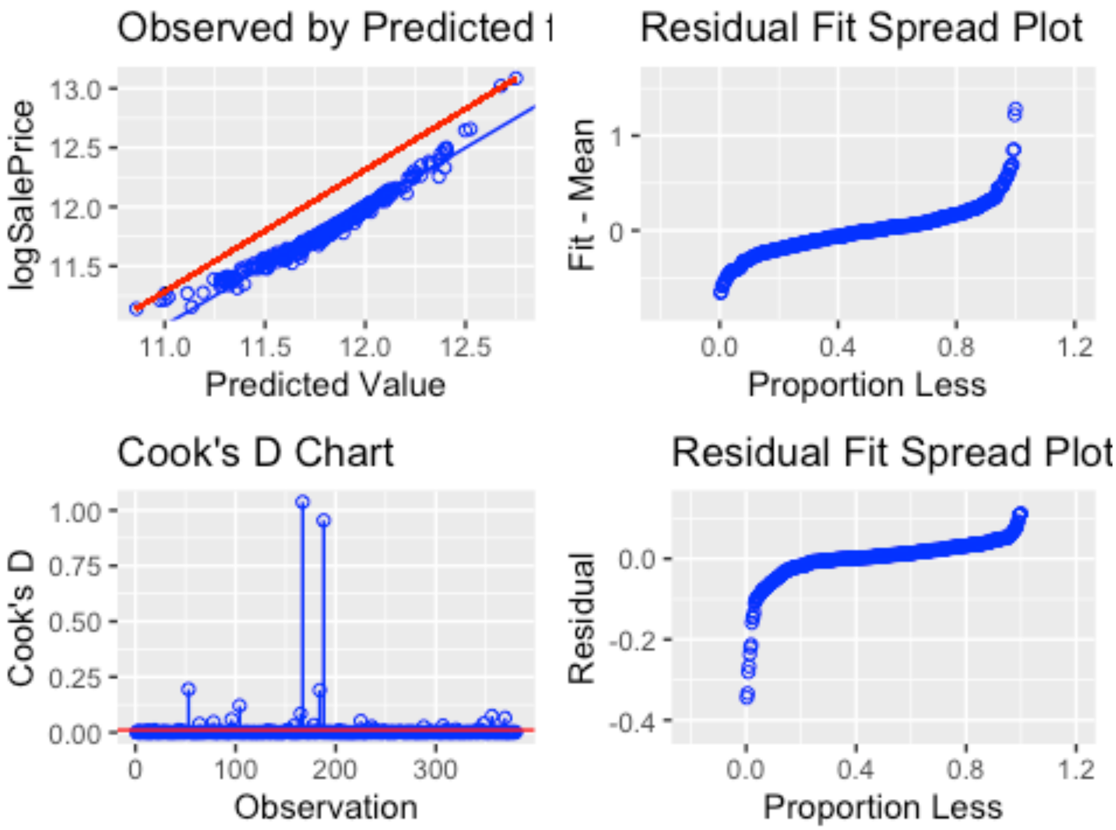
```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05415 on 374 degrees of freedom
## Multiple R-squared:  0.9593, Adjusted R-squared:  0.9587
## F-statistic:  1763 on 5 and 374 DF,  p-value: < 2.2e-16
```

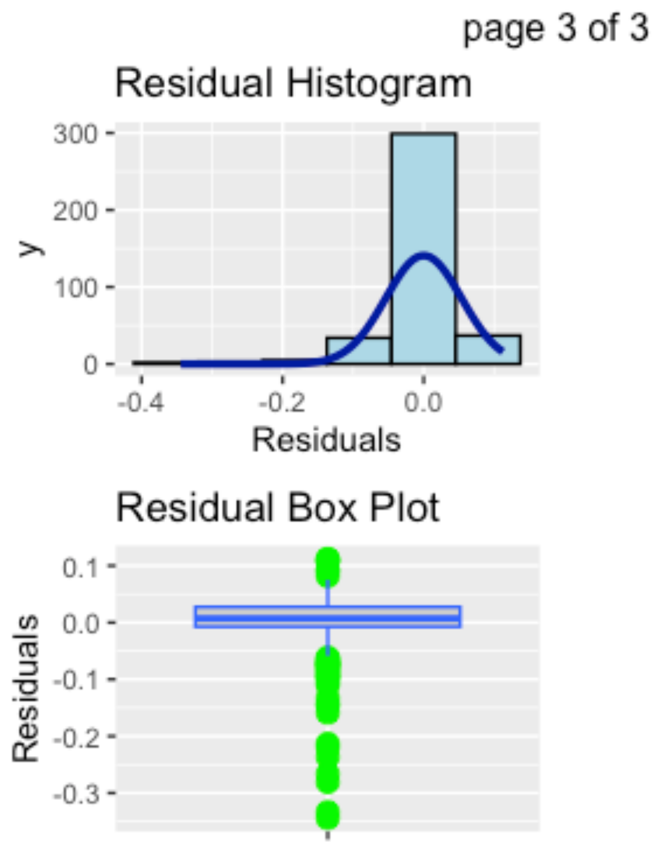
```
model_all<- train(logSalePrice ~ .,
                  data=century21,
                  trControl= trainControl(method = "LOOCV"),
                  method="lm")
model_all
```

```
## Linear Regression
##
## 383 samples
##   4 predictor
##
## No pre-processing
## Resampling: Leave-One-Out Cross-Validation
## Summary of sample sizes: 382, 382, 382, 382, 382, 382, ...
## Resampling results:
##
##   RMSE        Rsquared   MAE
##   0.07058551  0.9333152  0.03999122
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
# RMSE        Rsquared   MAE
# 0.07058551  0.9333152  0.03999122

ols_plot_diagnostics(fit_all) #plots of model
```


page 1 of 3
Residual vs Predicted Va
Residual
0.1
0.0
-0.1
-0.2
-0.3
11.5
12.0
12.5
13.0
Predicted Value
Outlier and Leverage Dia
RStudent
5
0
-5
-10
Leverage
Observation
normal
leverage
outlier
outlier & leverage
Deleted Studentized Resi
Deleted Studentized Residual
3
0
-3
-6
Predicted Value
Observation
normal
outlier
Normal Q-Q Plot
Sample Quantiles
0.1
0.0
-0.1
-0.2
-0.3
-2
0
2
Theoretical Quantiles


Observed by Predicted f
logSalePrice
13.0
12.5
12.0
11.5
11.0
11.5
12.0
12.5
Predicted Value
Residual Fit Spread Plot
Fit - Mean
1
0
0.0
0.4
0.8
1.2
Proportion Less
Cook's D Chart
Cook's D
1.00
0.75
0.50
0.25
0.00
0
100
200
300
Observation
Residual Fit Spread Plot
Residual
0.0
-0.2
-0.4
0.0
0.4
0.8
1.2
Proportion Less



####KNN model

```
# Train the KNN model with removed outliers
set.seed(123)
model_knn1 <- train(logSalePrice ~ logGrLivArea * Neighborhood, data = centur
y21_rmOutliers,
                   trControl = trainControl(method = "cv", number = 5),
                   method = "knn",
                   tuneGrid = data.frame(k = 3:50),
                   metric = "RMSE")
print(model_knn1)  # k= 10 was best model
```

```
## k-Nearest Neighbors
##
## 380 samples
##   2 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 305, 304, 304, 304, 303
## Resampling results across tuning parameters:
##
##   k   RMSE       Rsquared   MAE
##    3  0.2035761  0.4421091  0.1545601
```

```
##    4  0.1965671  0.4708876  0.1476002
##    5  0.1904300  0.4928833  0.1444115
##    6  0.1905016  0.4893824  0.1449592
##    7  0.1875523  0.5038054  0.1419662
##    8  0.1882008  0.4999375  0.1427657
##    9  0.1879826  0.5003859  0.1420521
##   10  0.1877742  0.5026462  0.1420739
##   11  0.1885834  0.4984000  0.1425174
##   12  0.1895908  0.4920082  0.1429158
##   13  0.1905983  0.4884234  0.1431292
##   14  0.1922573  0.4799351  0.1442879
##   15  0.1945448  0.4676967  0.1457871
##   16  0.1954137  0.4613352  0.1456162
##   17  0.1966233  0.4561868  0.1466138
##   18  0.1974475  0.4512342  0.1472237
##   19  0.1981174  0.4488901  0.1479110
##   20  0.1988310  0.4469837  0.1485428
##   21  0.2002359  0.4418613  0.1493220
##   22  0.2017827  0.4346177  0.1502933
##   23  0.2023012  0.4318923  0.1505045
##   24  0.2020520  0.4377465  0.1505713
##   25  0.2023459  0.4373182  0.1503665
##   26  0.2038553  0.4281555  0.1515109
##   27  0.2042314  0.4261720  0.1515466
##   28  0.2047093  0.4224635  0.1515828
##   29  0.2060835  0.4146934  0.1524340
##   30  0.2066364  0.4119263  0.1527276
##   31  0.2075907  0.4062685  0.1538671
##   32  0.2091681  0.3957879  0.1543943
##   33  0.2100520  0.3905107  0.1546691
##   34  0.2112161  0.3837026  0.1554874
##   35  0.2119426  0.3798020  0.1561180
##   36  0.2130822  0.3733286  0.1569027
##   37  0.2147290  0.3629356  0.1576367
##   38  0.2160606  0.3536580  0.1586868
##   39  0.2175838  0.3445984  0.1594796
##   40  0.2192817  0.3349703  0.1607623
##   41  0.2212346  0.3213890  0.1623347
##   42  0.2228454  0.3100127  0.1632310
##   43  0.2244508  0.2982933  0.1639580
##   44  0.2259533  0.2884003  0.1649673
##   45  0.2266033  0.2840750  0.1653394
##   46  0.2270076  0.2806375  0.1654080
##   47  0.2273156  0.2785578  0.1655607
##   48  0.2276863  0.2744603  0.1656568
##   49  0.2281508  0.2720533  0.1659864
##   50  0.2282236  0.2718407  0.1659890
## 
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was k = 7.
```

```
#   RMSE        Rsquared    MAE
#   0.1854957   0.5186253   0.1398700
```

Conclusion

RMSE was used to select the optimal model using the smallest value. The final value used for the model was k = 7. RMSE Rsquared MAE
0.1854957 0.5186253 0.1398700

There is a relationship between the sale price and the square footage of a house. An increase in square footage results in an increase in house prices. It also appears that the sale price of a house does depend on the neighborhood. BrkSide and Edwards appear to have houses similar in price while NAmes neighborhood in more expensive.

Analysis 2: Sale Price

Data Cleaning

In order to use a linear regression model, we need to convert all of the categorical variables into dummy variables.

```
library(olsrr)
#create new df for this analysis
ames2 <- ames_subset
ames2$logLotArea <- log(ames2$LotArea)
ames2$logGrLivArea <- log(ames2$GrLivArea)
ames2$logSalePrice <- log(ames2$SalePrice)

# Use the dummyVars() function to convert categorical variables into dummy va
riables
dummy_model <- dummyVars(~ ., data = ames2)
ames_dummy <- as.data.frame(predict(dummy_model, newdata = ames2))
str(ames_dummy)

## 'data.frame':    2919 obs. of  300 variables:
##  $ Id                  : num  1 2 3 4 5 6 7 8 9 10 ...
##  $ MSSubClass          : num  60 20 60 70 60 50 20 60 50 190 ...
##  $ MSZoningC (all)     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ MSZoningFV          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ MSZoningNone        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ MSZoningRH          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ MSZoningRL          : num  1 1 1 1 1 1 1 1 0 1 ...
##  $ MSZoningRM          : num  0 0 0 0 0 0 0 0 1 0 ...
##  $ LotFrontage         : num  65 80 68 60 84 85 75 0 51 50 ...
##  $ LotArea             : num  8450 9600 11250 9550 14260 ...
##  $ StreetGrvl          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ StreetPave          : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ LotShapeIR1         : num  0 0 1 1 1 1 0 1 0 0 ...
##  $ LotShapeIR2         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ LotShapeIR3         : num  0 0 0 0 0 0 0 0 0 0 ...
```

```
##  $ LotShapeReg         : num  1 1 0 0 0 0 1 0 1 1 ...
##  $ LandContourBnk      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ LandContourHLS      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ LandContourLow      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ LandContourLvl      : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ UtilitiesAllPub     : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ UtilitiesNone       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ UtilitiesNoSeWa     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ LotConfigCorner     : num  0 0 0 1 0 0 0 1 0 1 ...
##  $ LotConfigCulDSac    : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ LotConfigFR2        : num  0 1 0 0 1 0 0 0 0 0 ...
##  $ LotConfigFR3        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ LotConfigInside     : num  1 0 1 0 0 1 1 0 1 0 ...
##  $ LandSlopeGtl        : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ LandSlopeMod        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ LandSlopeSev        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ NeighborhoodBlmngtn : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ NeighborhoodBlueste : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ NeighborhoodBrDale  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ NeighborhoodBrkSide : num  0 0 0 0 0 0 0 0 0 1 ...
##  $ NeighborhoodClearCr : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ NeighborhoodCollgCr : num  1 0 1 0 0 0 0 0 0 0 ...
##  $ NeighborhoodCrawfor : num  0 0 0 1 0 0 0 0 0 0 ...
##  $ NeighborhoodEdwards : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ NeighborhoodGilbert : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ NeighborhoodIDOTRR  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ NeighborhoodMeadowV : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ NeighborhoodMitchel : num  0 0 0 0 0 1 0 0 0 0 ...
##  $ NeighborhoodNAmes   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ NeighborhoodNoRidge : num  0 0 0 0 1 0 0 0 0 0 ...
##  $ NeighborhoodNPkVill : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ NeighborhoodNridgHt : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ NeighborhoodNWAmes  : num  0 0 0 0 0 0 0 1 0 0 ...
##  $ NeighborhoodOldTown : num  0 0 0 0 0 0 0 0 1 0 ...
##  $ NeighborhoodSawyer  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ NeighborhoodSawyerW : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ NeighborhoodSomerst : num  0 0 0 0 0 0 1 0 0 0 ...
##  $ NeighborhoodStoneBr : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ NeighborhoodSWISU   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ NeighborhoodTimber  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ NeighborhoodVeenker : num  0 1 0 0 0 0 0 0 0 0 ...
##  $ Condition1Artery    : num  0 0 0 0 0 0 0 0 1 1 ...
##  $ Condition1Feedr     : num  0 1 0 0 0 0 0 0 0 0 ...
##  $ Condition1Norm      : num  1 0 1 1 1 1 1 0 0 0 ...
##  $ Condition1PosA      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Condition1PosN      : num  0 0 0 0 0 0 0 1 0 0 ...
##  $ Condition1RRAe      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Condition1RRAn      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Condition1RRNe      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Condition1RRNn      : num  0 0 0 0 0 0 0 0 0 0 ...
```

```
## $ Condition2Artery    : num  0 0 0 0 0 0 0 0 0 1 ...
## $ Condition2Feedr     : num  0 0 0 0 0 0 0 0 0 0 ...
## $ Condition2Norm      : num  1 1 1 1 1 1 1 1 1 0 ...
## $ Condition2PosA      : num  0 0 0 0 0 0 0 0 0 0 ...
## $ Condition2PosN      : num  0 0 0 0 0 0 0 0 0 0 ...
## $ Condition2RRAe      : num  0 0 0 0 0 0 0 0 0 0 ...
## $ Condition2RRAn      : num  0 0 0 0 0 0 0 0 0 0 ...
## $ Condition2RRNn      : num  0 0 0 0 0 0 0 0 0 0 ...
## $ BldgType1Fam        : num  1 1 1 1 1 1 1 1 1 0 ...
## $ BldgType2fmCon      : num  0 0 0 0 0 0 0 0 0 1 ...
## $ BldgTypeDuplex      : num  0 0 0 0 0 0 0 0 0 0 ...
## $ BldgTypeTwnhs       : num  0 0 0 0 0 0 0 0 0 0 ...
## $ BldgTypeTwnhsE      : num  0 0 0 0 0 0 0 0 0 0 ...
## $ HouseStyle1.5Fin    : num  0 0 0 0 0 1 0 0 1 0 ...
## $ HouseStyle1.5Unf    : num  0 0 0 0 0 0 0 0 0 1 ...
## $ HouseStyle1Story    : num  0 1 0 0 0 0 1 0 0 0 ...
## $ HouseStyle2.5Fin    : num  0 0 0 0 0 0 0 0 0 0 ...
## $ HouseStyle2.5Unf    : num  0 0 0 0 0 0 0 0 0 0 ...
## $ HouseStyle2Story    : num  1 0 1 1 1 0 0 1 0 0 ...
## $ HouseStyleSFoyer    : num  0 0 0 0 0 0 0 0 0 0 ...
## $ HouseStyleSLvl      : num  0 0 0 0 0 0 0 0 0 0 ...
## $ OverallQual         : num  7 6 7 7 8 5 8 7 7 5 ...
## $ OverallCond         : num  5 8 5 5 5 5 5 6 5 6 ...
## $ YearBuilt           : num  2003 1976 2001 1915 2000 ...
## $ YearRemodAdd        : num  2003 1976 2002 1970 2000 ...
## $ RoofStyleFlat       : num  0 0 0 0 0 0 0 0 0 0 ...
## $ RoofStyleGable      : num  1 1 1 1 1 1 1 1 1 1 ...
## $ RoofStyleGambrel    : num  0 0 0 0 0 0 0 0 0 0 ...
## $ RoofStyleHip        : num  0 0 0 0 0 0 0 0 0 0 ...
## $ RoofStyleMansard    : num  0 0 0 0 0 0 0 0 0 0 ...
## $ RoofStyleShed       : num  0 0 0 0 0 0 0 0 0 0 ...
## $ RoofMatlClyTile     : num  0 0 0 0 0 0 0 0 0 0 ...
## $ RoofMatlCompShg     : num  1 1 1 1 1 1 1 1 1 1 ...
## $ RoofMatlMembran     : num  0 0 0 0 0 0 0 0 0 0 ...
##   [list output truncated]

# Split the data into training and testing sets
train_model <- ames_dummy[ames_dummy$traintrain == 1, ]
test_model <- ames_dummy[ames_dummy$traintrain == 0, ]
train_model$traintest <- NULL
train_model$traintrain <- NULL
test_model$traintest <- NULL
test_model$traintrain <- NULL
```

Removing columns with over 80% likeness

```
# Check the difference in column numbers
cat("Original number of columns:", ncol(train_model), "\n")

## Original number of columns: 298
```

```r
# Apply the function
train_model <- remove_low_variance_columns(train_model)
cat("Number of columns after removal:", ncol(train_model), "\n")

## Number of columns after removal: 64
```

Remove unique columns from the test_df to match train_df

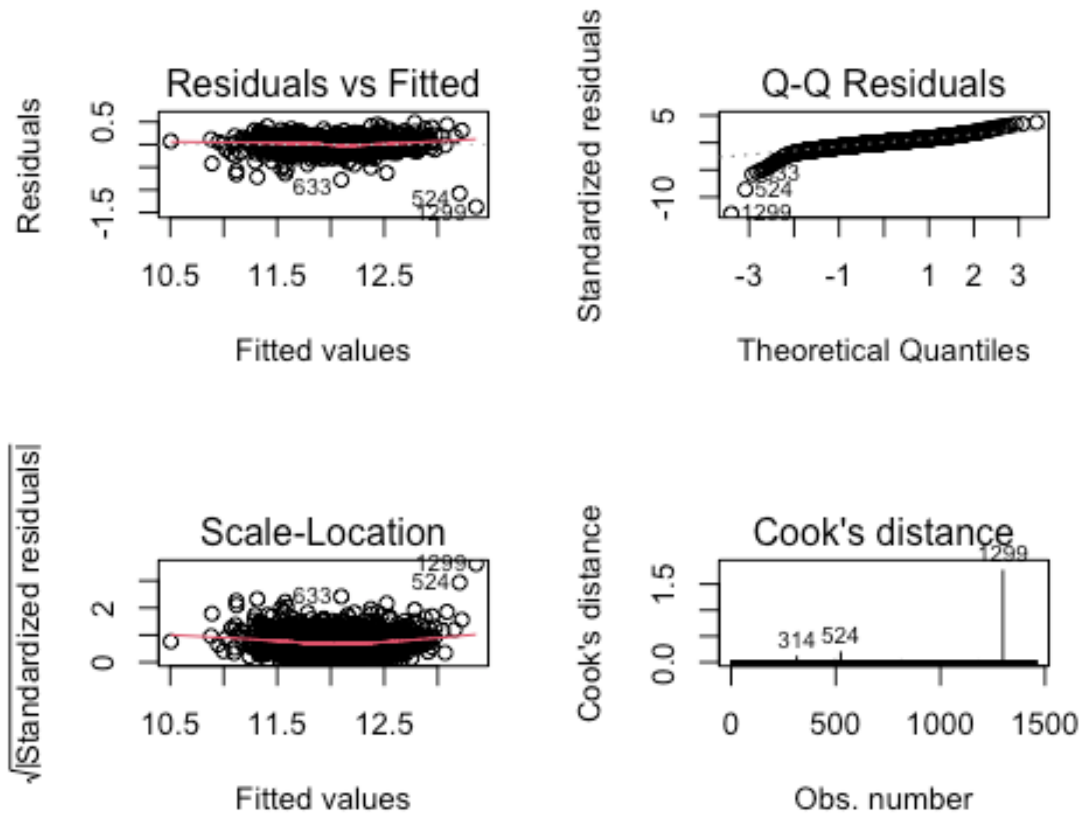
```r
# Find the common column names between train_model and test_model
common_columns <- intersect(names(train_model), names(test_model))

# Keep only the common columns in test_model
test_model <- test_model[, common_columns]
```

Assumptions and Influential Points

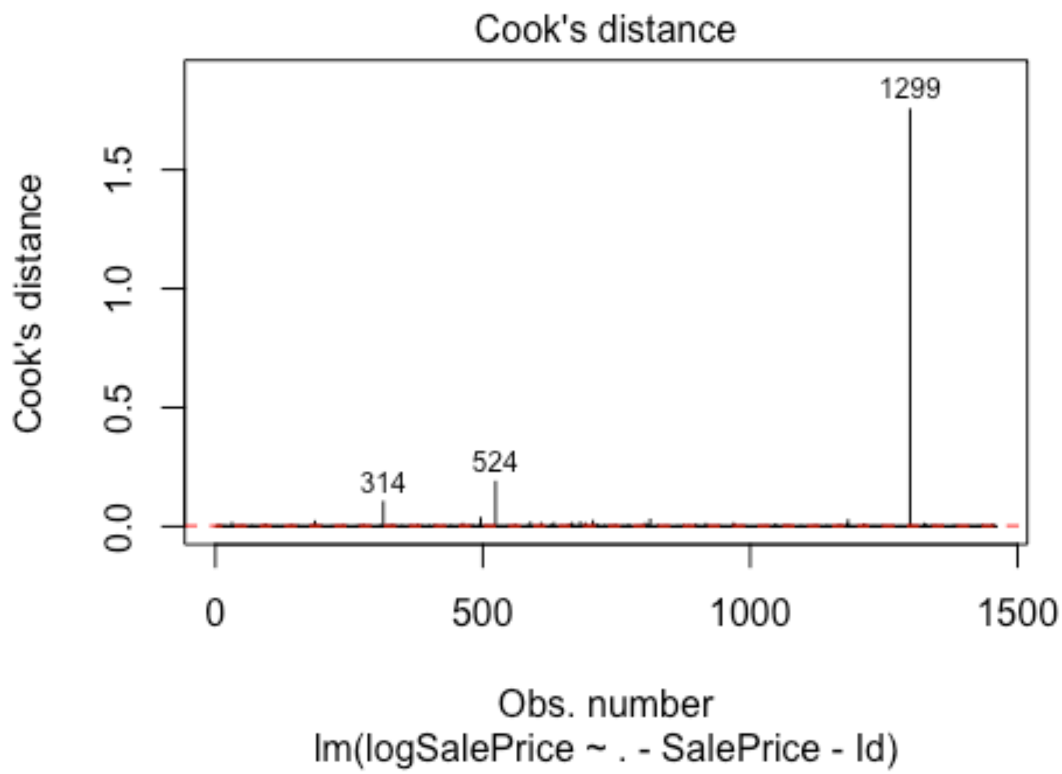
```r
#fit model 1 of log data
fit <- lm(logSalePrice ~ . - SalePrice - Id, data = train_model) # Exclude Sa
lePrice, ID
summary(fit)

##
## Call:
## lm(formula = logSalePrice ~ . - SalePrice - Id, data = train_model)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.37585 -0.06221  0.00616  0.07040  0.49068
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         1.469e+01  5.618e+00   2.615 0.009012 **
## MSSubClass         -2.119e-04  1.252e-04  -1.693 0.090686 .
## MSZoningRL          3.527e-02  1.163e-02   3.034 0.002459 **
## LotFrontage        -2.556e-04  1.188e-04  -2.151 0.031627 *
## LotArea            -1.633e-07  5.640e-07  -0.290 0.772190
## LotShapeIR1         5.250e-03  2.152e-02   0.244 0.807298
## LotShapeReg         5.983e-03  2.194e-02   0.273 0.785172
## LotConfigInside    -3.674e-03  8.380e-03  -0.438 0.661196
## HouseStyle1Story    3.959e-02  1.443e-02   2.744 0.006141 **
## HouseStyle2Story   -6.014e-02  1.544e-02  -3.896 0.000102 ***
## OverallQual         7.241e-02  4.983e-03  14.532  < 2e-16 ***
## OverallCond         5.549e-02  4.206e-03  13.195  < 2e-16 ***
## YearBuilt           2.293e-03  3.170e-04   7.235 7.67e-13 ***
## YearRemodAdd        8.371e-04  2.891e-04   2.895 0.003848 **
## RoofStyleGable     -4.801e-03  9.696e-03  -0.495 0.620571
## Exterior1stVinylSd -1.487e-02  3.625e-02  -0.410 0.681656
## Exterior2ndVinylSd  8.254e-04  3.635e-02   0.023 0.981889
## MasVnrTypeBrkFace  -9.703e-03  1.431e-02  -0.678 0.497715
## MasVnrTypeNone      1.500e-02  1.561e-02   0.961 0.336661
## MasVnrArea          8.534e-05  3.091e-05   2.761 0.005840 **
## ExterQualGd         3.780e-03  2.233e-02   0.169 0.865603
```

```
## ExterQualTA        -8.242e-03  2.335e-02  -0.353 0.724150    
## FoundationCBlock    2.827e-02  1.494e-02   1.891 0.058765 .  
## FoundationPConc     6.037e-02  1.726e-02   3.498 0.000483 ***
## BsmtQualGd         -2.505e-02  1.427e-02  -1.756 0.079313 .  
## BsmtQualTA         -2.234e-02  1.553e-02  -1.438 0.150597    
## BsmtExposureNo     -2.310e-02  8.923e-03  -2.589 0.009719 ** 
## BsmtFinType1GLQ     1.318e-02  1.214e-02   1.085 0.278070    
## BsmtFinType1Unf    -5.058e-02  1.284e-02  -3.939 8.59e-05 ***
## BsmtFinSF1         -8.668e-06  2.441e-05  -0.355 0.722546    
## BsmtUnfSF          -7.496e-06  2.607e-05  -0.288 0.773752    
## TotalBsmtSF         1.105e-04  2.878e-05   3.838 0.000130 ***
## HeatingQCEx         2.327e-02  1.085e-02   2.145 0.032106 *  
## HeatingQCTA        -1.006e-02  1.084e-02  -0.928 0.353529    
## X1stFlrSF          -7.638e-05  8.112e-05  -0.942 0.346602    
## X2ndFlrSF           7.337e-05  8.119e-05   0.904 0.366311    
## GrLivArea          -1.489e-04  8.253e-05  -1.804 0.071450 .  
## BsmtFullBath        4.225e-02  1.007e-02   4.197 2.88e-05 ***
## FullBath            2.389e-02  1.140e-02   2.096 0.036276 *  
## HalfBath            1.849e-02  1.104e-02   1.675 0.094226 .  
## BedroomAbvGr       -9.982e-03  7.045e-03  -1.417 0.156768    
## KitchenQualGd      -5.529e-02  1.587e-02  -3.484 0.000510 ***
## KitchenQualTA      -5.514e-02  1.618e-02  -3.407 0.000674 ***
## TotRmsAbvGrd        2.314e-03  4.835e-03   0.479 0.632307    
## Fireplaces          3.044e-02  1.393e-02   2.185 0.029059 *  
## FireplaceQuGd       4.067e-03  1.759e-02   0.231 0.817140    
## FireplaceQuNone    -1.779e-02  2.283e-02  -0.779 0.435928    
## FireplaceQuTA      -1.996e-02  1.797e-02  -1.111 0.266668    
## GarageTypeAttchd    1.683e-02  1.456e-02   1.156 0.248005    
## GarageTypeDetchd    2.209e-02  1.727e-02   1.279 0.201082    
## GarageYrBlt        -9.818e-05  3.073e-04  -0.319 0.749422    
## GarageFinishFin     2.022e-01  5.995e-01   0.337 0.735931    
## GarageFinishRFn     1.995e-01  5.988e-01   0.333 0.739002    
## GarageFinishUnf     1.851e-01  5.984e-01   0.309 0.757057    
## GarageCars          5.418e-02  1.194e-02   4.538 6.17e-06 ***
## GarageArea          2.932e-05  4.119e-05   0.712 0.476633    
## WoodDeckSF          7.362e-05  3.134e-05   2.349 0.018939 *  
## OpenPorchSF        -2.849e-05  6.028e-05  -0.473 0.636569    
## MoSold              2.110e-04  1.344e-03   0.157 0.875291    
## YrSold             -7.339e-03  2.772e-03  -2.648 0.008198 ** 
## logLotArea          7.673e-02  1.477e-02   5.193 2.37e-07 ***
## logGrLivArea        6.150e-01  5.150e-02  11.943  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.1349 on 1398 degrees of freedom
## Multiple R-squared:  0.8908, Adjusted R-squared:  0.886 
## F-statistic: 186.9 on 61 and 1398 DF,  p-value: < 2.2e-16

#Check the residuals
par(mfrow=c(2,2)) # Set up a 2x2 plot grid
```

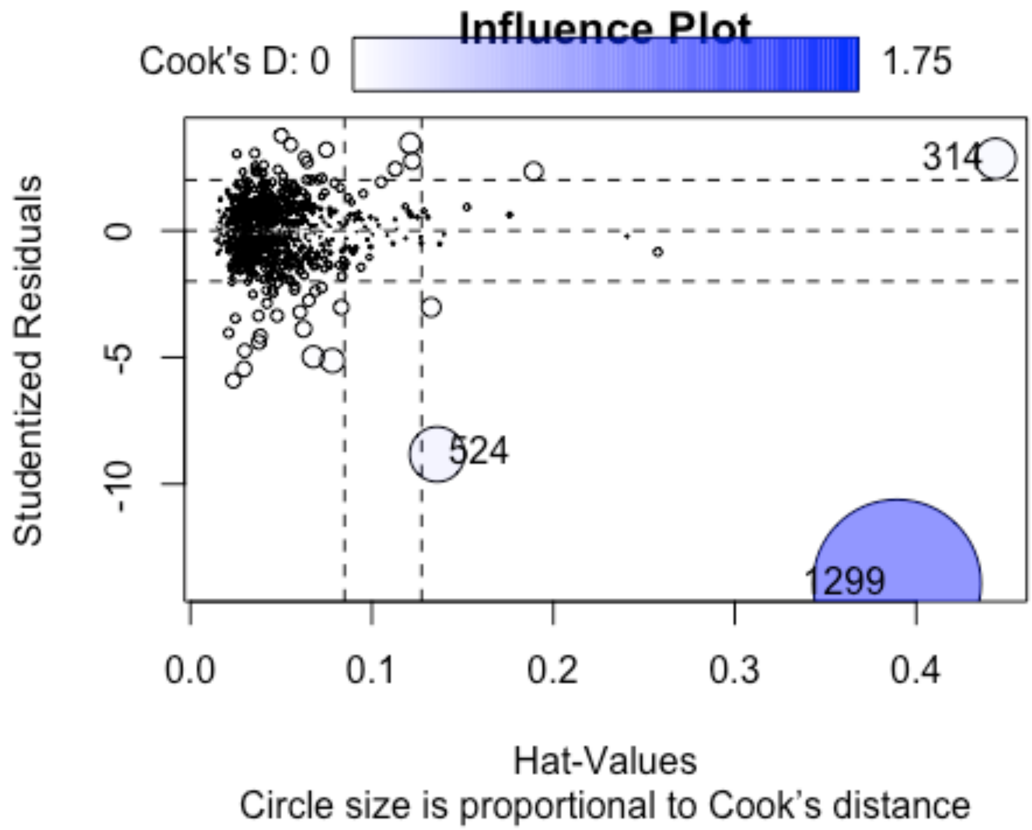
```r
plot(fit, which = 1) # Residuals vs Fitted
plot(fit, which = 2) # Normal Q-Q
plot(fit, which = 3) # Scale-Location
plot(fit, which = 4) # Cook's distance
```




```r
par(mfrow = c(1, 1)) # Set up a 1x1 plot grid


# ols_plot_diagnostics(fit)

#Cooks D Plot
cutoff <- 4/(nrow(train_model)-length(fit$coefficients)-2)
plot(fit, which=4, cook.levels=cutoff)
abline(h=cutoff, lty=2, col="red")
```




```
#Influence Plot
influencePlot(fit, id.method="identify", main="Influence Plot", sub="Circle s
ize is proportional to Cook’s distance")
```




```
##          StudRes       Hat     CookD
## 314     2.837453 0.4437823 0.1030874
## 524    -8.825894 0.1359864 0.1874331
## 1299 -13.927822 0.3894210 1.7534510

#Influencial points 314, 524, 1299
```

Removing Influential Outliers

```
#Outliers Addressed/Observation 339 influential point/outlier removed.
outliers_to_remove <- c(314,524,1299)
train_model_rmOutliers <- train_model[-outliers_to_remove,]
str(train_model)
```

```
## 'data.frame':    1460 obs. of  64 variables:
##  $ Id                : num  1 2 3 4 5 6 7 8 9 10 ...
##  $ MSSubClass        : num  60 20 60 70 60 50 20 60 50 190 ...
##  $ MSZoningRL        : num  1 1 1 1 1 1 1 1 0 1 ...
##  $ LotFrontage       : num  65 80 68 60 84 85 75 0 51 50 ...
##  $ LotArea           : num  8450 9600 11250 9550 14260 ...
##  $ LotShapeIR1       : num  0 0 1 1 1 1 0 1 0 0 ...
##  $ LotShapeReg       : num  1 1 0 0 0 0 1 0 1 1 ...
##  $ LotConfigInside   : num  1 0 1 0 0 1 1 0 1 0 ...
##  $ HouseStyle1Story  : num  0 1 0 0 0 0 1 0 0 0 ...
##  $ HouseStyle2Story  : num  1 0 1 1 1 0 0 1 0 0 ...
```

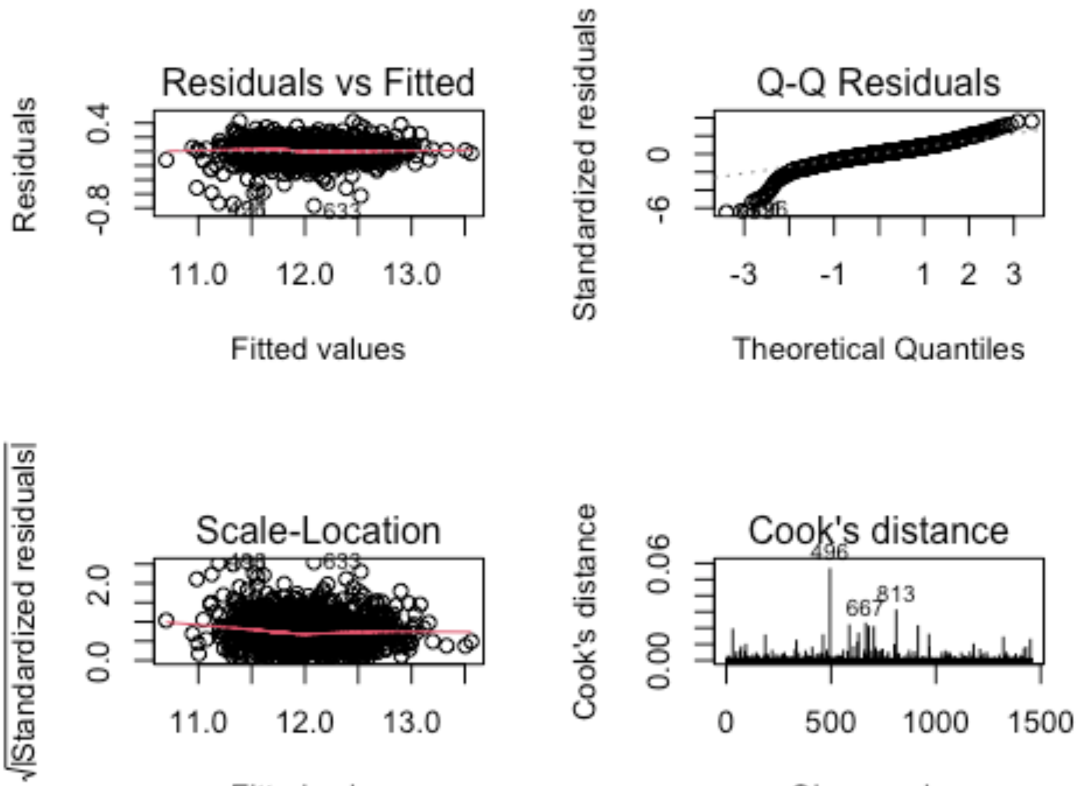
```
##  $ OverallQual       : num  7 6 7 7 8 5 8 7 7 5 ...
##  $ OverallCond       : num  5 8 5 5 5 5 5 6 5 6 ...
##  $ YearBuilt         : num  2003 1976 2001 1915 2000 ...
##  $ YearRemodAdd      : num  2003 1976 2002 1970 2000 ...
##  $ RoofStyleGable    : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ Exterior1stVinylSd: num  1 0 1 0 1 1 1 0 0 0 ...
##  $ Exterior2ndVinylSd: num  1 0 1 0 1 1 1 0 0 0 ...
##  $ MasVnrTypeBrkFace : num  1 0 1 0 1 0 0 0 0 0 ...
##  $ MasVnrTypeNone    : num  0 1 0 1 0 1 0 0 1 1 ...
##  $ MasVnrArea        : num  196 0 162 0 350 0 186 240 0 0 ...
##  $ ExterQualGd       : num  1 0 1 0 1 0 1 0 0 0 ...
##  $ ExterQualTA       : num  0 1 0 1 0 1 0 1 1 1 ...
##  $ FoundationCBlock  : num  0 1 0 0 0 0 0 1 0 0 ...
##  $ FoundationPConc   : num  1 0 1 0 1 0 1 0 0 0 ...
##  $ BsmtQualGd        : num  1 1 1 0 1 1 0 1 0 0 ...
##  $ BsmtQualTA        : num  0 0 0 1 0 0 0 0 1 1 ...
##  $ BsmtExposureNo    : num  1 0 0 1 0 1 0 0 1 1 ...
##  $ BsmtFinType1GLQ   : num  1 0 1 0 1 1 1 0 0 1 ...
##  $ BsmtFinType1Unf   : num  0 0 0 0 0 0 0 0 1 0 ...
##  $ BsmtFinSF1        : num  706 978 486 216 655 ...
##  $ BsmtUnfSF         : num  150 284 434 540 490 64 317 216 952 140 ...
##  $ TotalBsmtSF       : num  856 1262 920 756 1145 ...
##  $ HeatingQCEx       : num  1 1 1 0 1 1 1 1 0 1 ...
##  $ HeatingQCTA       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ X1stFlrSF         : num  856 1262 920 961 1145 ...
##  $ X2ndFlrSF         : num  854 0 866 756 1053 ...
##  $ GrLivArea         : num  1710 1262 1786 1717 2198 ...
##  $ BsmtFullBath      : num  1 0 1 1 1 1 1 1 0 1 ...
##  $ FullBath          : num  2 2 2 1 2 1 2 2 2 1 ...
##  $ HalfBath          : num  1 0 1 0 1 1 0 1 0 0 ...
##  $ BedroomAbvGr      : num  3 3 3 3 4 1 3 3 2 2 ...
##  $ KitchenQualGd     : num  1 0 1 1 1 0 1 0 0 0 ...
##  $ KitchenQualTA     : num  0 1 0 0 0 1 0 1 1 1 ...
##  $ TotRmsAbvGrd      : num  8 6 6 7 9 5 7 7 8 5 ...
##  $ Fireplaces        : num  0 1 1 1 1 0 1 2 2 2 ...
##  $ FireplaceQuGd     : num  0 0 0 1 0 0 1 0 0 0 ...
##  $ FireplaceQuNone   : num  1 0 0 0 0 1 0 0 0 0 ...
##  $ FireplaceQuTA     : num  0 1 1 0 1 0 0 1 1 1 ...
##  $ GarageTypeAttchd  : num  1 1 1 0 1 1 1 1 0 1 ...
##  $ GarageTypeDetchd  : num  0 0 0 1 0 0 0 0 1 0 ...
##  $ GarageYrBlt       : num  2003 1976 2001 1998 2000 ...
##  $ GarageFinishFin   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ GarageFinishRFn   : num  1 1 1 0 1 0 1 1 0 1 ...
##  $ GarageFinishUnf   : num  0 0 0 1 0 1 0 0 1 0 ...
##  $ GarageCars        : num  2 2 2 3 3 2 2 2 2 1 ...
##  $ GarageArea        : num  548 460 608 642 836 480 636 484 468 205 ...
##  $ WoodDeckSF        : num  0 298 0 0 192 40 255 235 90 0 ...
##  $ OpenPorchSF       : num  61 0 42 35 84 30 57 204 0 4 ...
##  $ MoSold            : num  2 5 9 2 12 10 8 11 4 1 ...
##  $ YrSold            : num  2008 2007 2008 2006 2008 ...
```

```
##  $ SalePrice        : num  208500 181500 223500 140000 250000 ...
##  $ logLotArea       : num  9.04 9.17 9.33 9.16 9.57 ...
##  $ logGrLivArea     : num  7.44 7.14 7.49 7.45 7.7 ...
##  $ logSalePrice     : num  12.2 12.1 12.3 11.8 12.4 ...
```

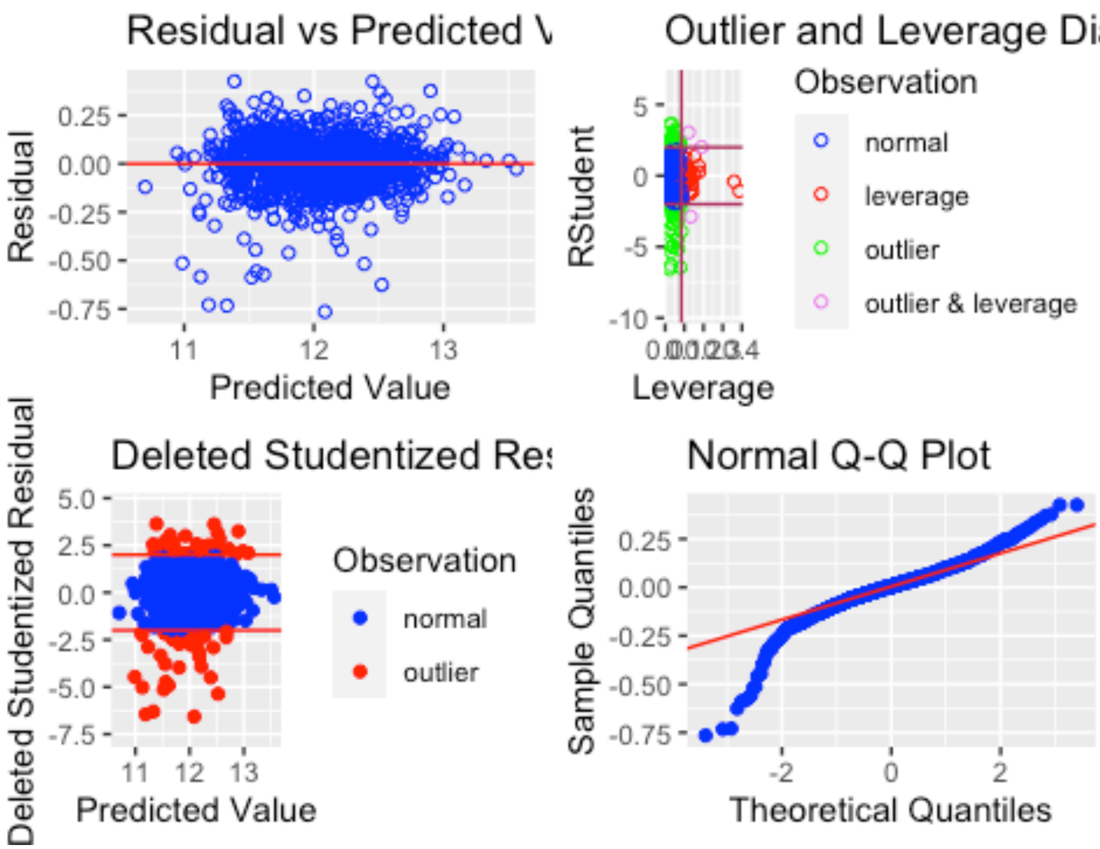
```
#Verify Assumption again
refit <- lm(logSalePrice ~ . - SalePrice - Id, data = train_model_rmOutliers)
# Exclude SalePrice, ID
summary(refit)
```

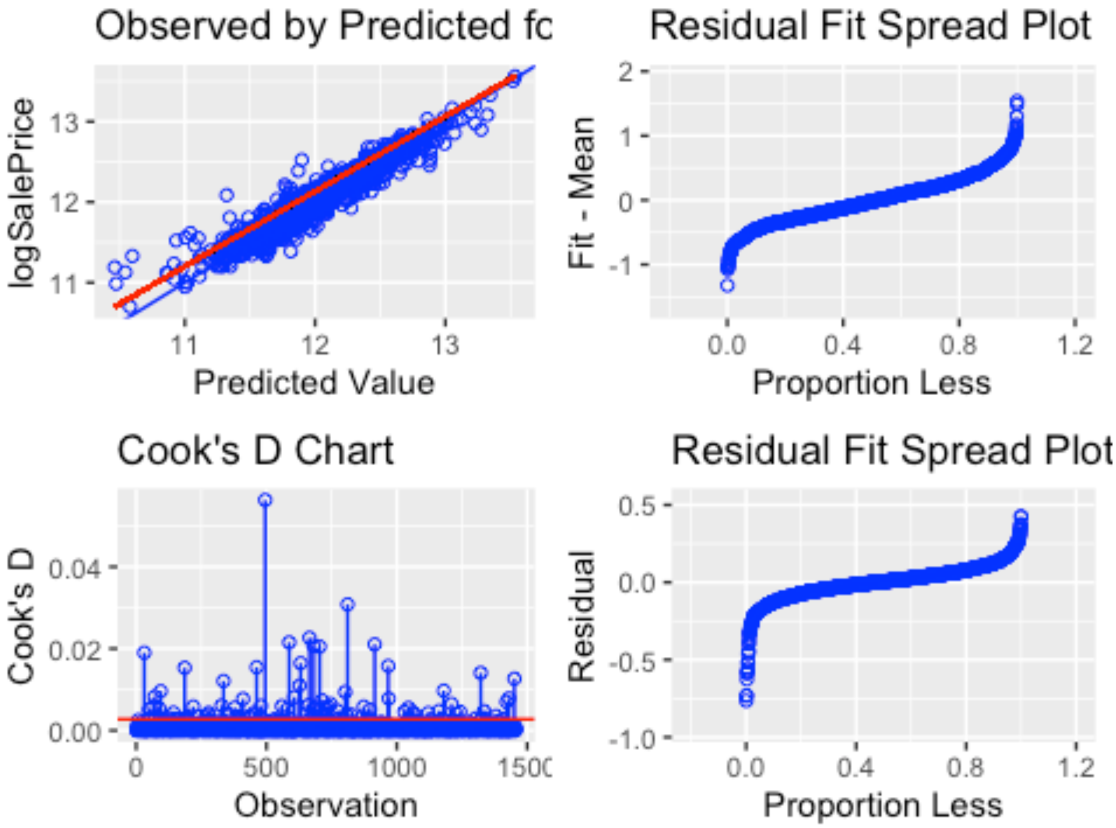
```
##
## Call:
## lm(formula = logSalePrice ~ . - SalePrice - Id, data = train_model_rmOutli
ers)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.76499 -0.05326  0.00599  0.06245  0.42550
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          1.414e+01  4.985e+00   2.836 0.004637 **
## MSSubClass          -1.100e-04  1.121e-04  -0.981 0.326753
## MSZoningRL           4.005e-02  1.032e-02   3.881 0.000109 ***
## LotFrontage         -5.016e-06  1.075e-04  -0.047 0.962776
## LotArea             -2.360e-07  6.422e-07  -0.367 0.713340
## LotShapeIR1         -1.072e-02  1.920e-02  -0.558 0.576618
## LotShapeReg         -1.147e-02  1.957e-02  -0.586 0.557921
## LotConfigInside     -2.703e-03  7.446e-03  -0.363 0.716670
## HouseStyle1Story     2.159e-04  1.297e-02   0.017 0.986718
## HouseStyle2Story    -2.422e-02  1.382e-02  -1.752 0.079988 .
## OverallQual          6.500e-02  4.442e-03  14.633  < 2e-16 ***
## OverallCond          5.552e-02  3.730e-03  14.883  < 2e-16 ***
## YearBuilt            2.611e-03  2.817e-04   9.267  < 2e-16 ***
## YearRemodAdd         8.933e-04  2.566e-04   3.481 0.000516 ***
## RoofStyleGable       9.971e-04  8.616e-03   0.116 0.907885
## Exterior1stVinylSd -1.178e-02  3.216e-02  -0.366 0.714058
## Exterior2ndVinylSd -4.740e-03  3.225e-02  -0.147 0.883167
## MasVnrTypeBrkFace   -2.409e-02  1.275e-02  -1.889 0.059063 .
## MasVnrTypeNone      -4.991e-03  1.390e-02  -0.359 0.719575
## MasVnrArea           5.769e-05  2.753e-05   2.096 0.036306 *
## ExterQualGd         -8.895e-03  1.983e-02  -0.449 0.653728
## ExterQualTA         -1.099e-02  2.072e-02  -0.531 0.595836
## FoundationCBlock     6.346e-03  1.332e-02   0.477 0.633767
## FoundationPConc      4.924e-02  1.532e-02   3.214 0.001339 **
## BsmtQualGd          -3.199e-02  1.266e-02  -2.527 0.011617 *
## BsmtQualTA          -2.771e-02  1.379e-02  -2.009 0.044699 *
## BsmtExposureNo      -1.566e-02  7.925e-03  -1.976 0.048335 *
## BsmtFinType1GLQ      3.841e-03  1.078e-02   0.356 0.721797
## BsmtFinType1Unf     -2.330e-02  1.151e-02  -2.025 0.043086 *
```

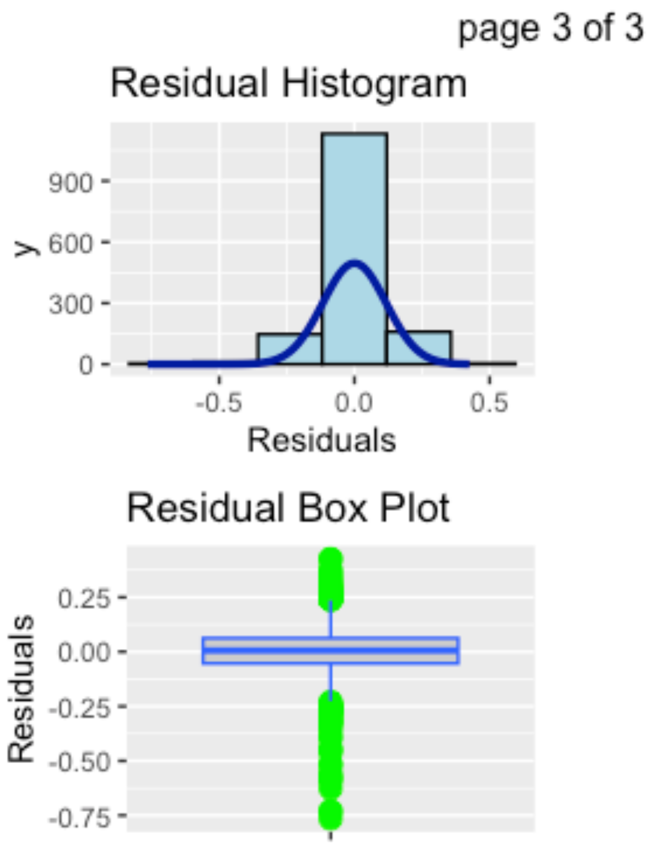
```
## BsmtFinSF1          4.130e-05  2.187e-05   1.888 0.059227 .
## BsmtUnfSF          -1.600e-05  2.326e-05  -0.688 0.491585
## TotalBsmtSF         1.521e-04  2.577e-05   5.903 4.48e-09 ***
## HeatingQCEx         1.942e-02  9.624e-03   2.018 0.043817 *
## HeatingQCTA        -1.340e-02  9.620e-03  -1.393 0.163757
## X1stFlrSF           3.697e-05  7.220e-05   0.512 0.608742
## X2ndFlrSF           9.980e-05  7.204e-05   1.385 0.166160
## GrLivArea           5.264e-05  7.393e-05   0.712 0.476594
## BsmtFullBath        2.243e-02  9.003e-03   2.492 0.012835 *
## FullBath            1.300e-02  1.013e-02   1.283 0.199686
## HalfBath            2.010e-02  9.796e-03   2.051 0.040427 *
## BedroomAbvGr       -1.013e-02  6.253e-03  -1.620 0.105410
## KitchenQualGd      -3.480e-02  1.412e-02  -2.465 0.013840 *
## KitchenQualTA      -3.594e-02  1.439e-02  -2.498 0.012615 *
## TotRmsAbvGrd       -3.252e-03  4.306e-03  -0.755 0.450265
## Fireplaces          2.762e-02  1.245e-02   2.218 0.026681 *
## FireplaceQuGd       2.703e-02  1.565e-02   1.727 0.084318 .
## FireplaceQuNone     3.500e-04  2.034e-02   0.017 0.986272
## FireplaceQuTA      -4.545e-03  1.596e-02  -0.285 0.775896
## GarageTypeAttchd    1.494e-02  1.294e-02   1.154 0.248594
## GarageTypeDetchd    2.354e-02  1.533e-02   1.535 0.125025
## GarageYrBlt         8.855e-05  2.730e-04   0.324 0.745745
## GarageFinishFin    -1.255e-01  5.326e-01  -0.236 0.813674
## GarageFinishRFn    -1.274e-01  5.319e-01  -0.239 0.810763
## GarageFinishUnf    -1.392e-01  5.316e-01  -0.262 0.793442
## GarageCars          2.422e-02  1.077e-02   2.249 0.024696 *
## GarageArea          5.795e-05  3.681e-05   1.574 0.115679
## WoodDeckSF          4.518e-05  2.801e-05   1.613 0.106929
## OpenPorchSF         5.904e-05  5.378e-05   1.098 0.272457
## MoSold              2.356e-04  1.193e-03   0.197 0.843532
## YrSold             -6.226e-03  2.461e-03  -2.530 0.011525 *
## logLotArea          7.652e-02  1.405e-02   5.445 6.10e-08 ***
## logGrLivArea        2.280e-01  4.982e-02   4.577 5.13e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1196 on 1395 degrees of freedom
## Multiple R-squared:  0.914,  Adjusted R-squared:  0.9102
## F-statistic:   243 on 61 and 1395 DF,  p-value: < 2.2e-16

#Check the residuals
par(mfrow=c(2,2)) # Set up a 2x2 plot grid
plot(refit, which = 1) # Residuals vs Fitted
plot(refit, which = 2) # Normal Q-Q
plot(refit, which = 3) # Scale-Location
plot(refit, which = 4) # Cook's distance
```



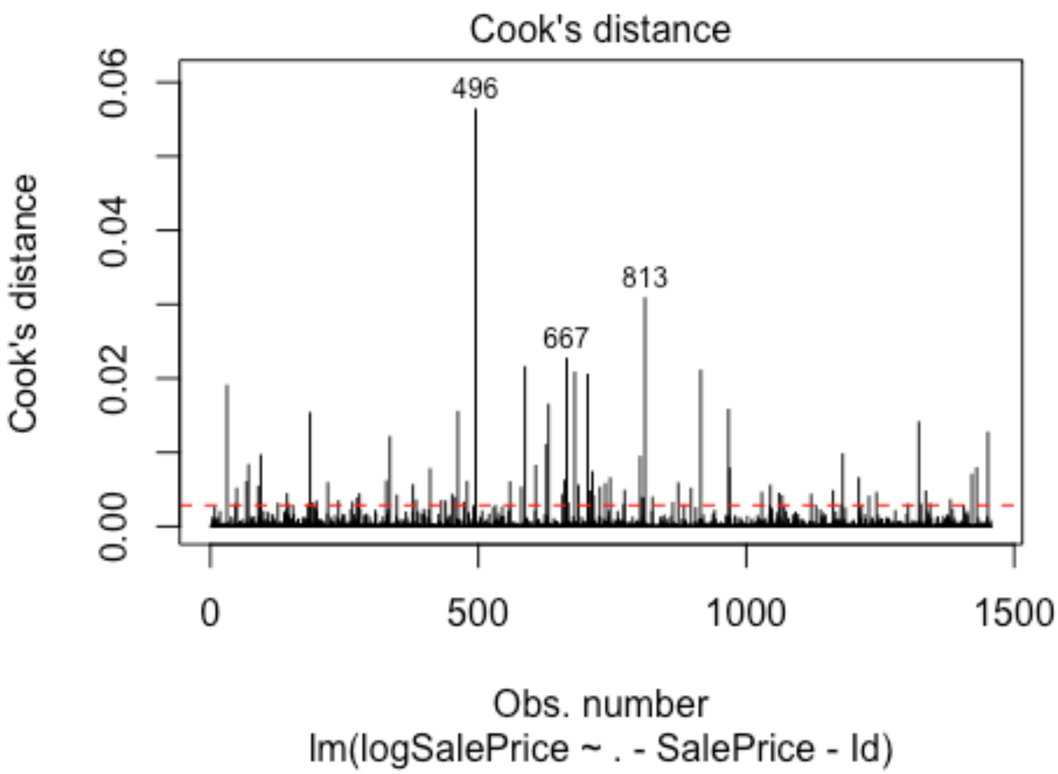

```r
par(mfrow = c(1, 1)) # Set up a 1x1 plot grid

ols_plot_diagnostics(refit)
```


Residual vs Predicted V
Residual
Predicted Value
Outlier and Leverage Dia
RStudent
Leverage
Observation
normal
leverage
outlier
outlier & leverage
Deleted Studentized Res
Deleted Studentized Residual
Predicted Value
Observation
normal
outlier
Normal Q-Q Plot
Sample Quantiles
Theoretical Quantiles

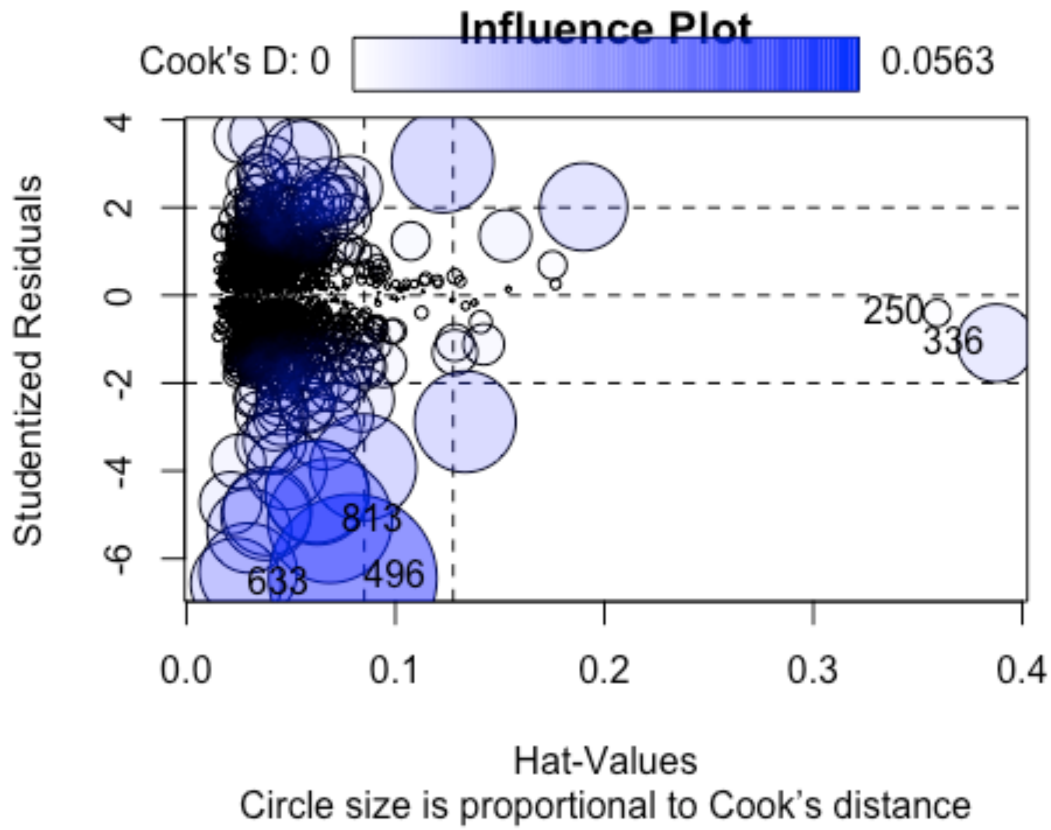

Observed by Predicted fc
logSalePrice
Predicted Value
Residual Fit Spread Plot
Fit - Mean
Proportion Less
Cook's D Chart
Cook's D
Observation
Residual Fit Spread Plot
Residual
Proportion Less




```r
#Cooks D Plot
cutoff <- 4/(nrow(train_model_rmOutliers)-length(refit$coefficients)-2)
plot(refit, which=4, cook.levels=cutoff)
abline(h=cutoff, lty=2, col="red")
```

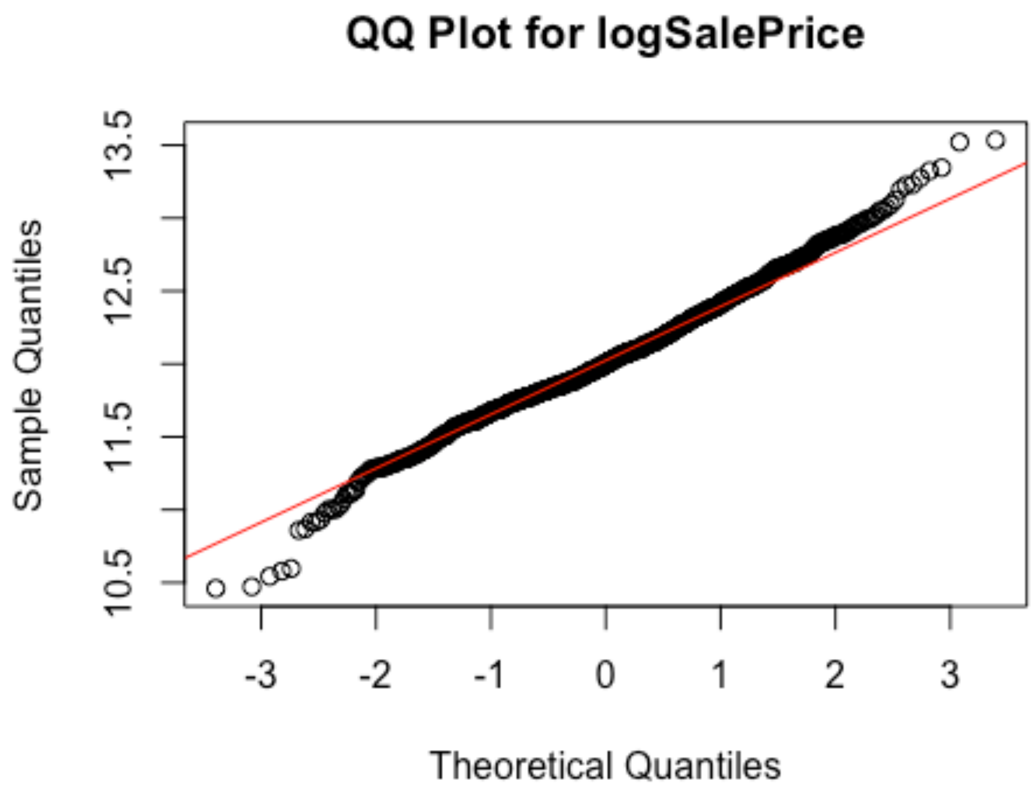



```
#Influence Plot
influencePlot(refit, id.method="identify", main="Influence Plot", sub="Circle
size is proportional to Cook’s distance")
```




```
##         StudRes        Hat       CookD
## 250 -0.4023924 0.35921570 0.001464915
## 336 -1.0887307 0.38747326 0.012092281
## 496 -6.4471286 0.07958575 0.056330568
## 633 -6.5690063 0.02376937 0.016449186
## 813 -5.1370949 0.06863692 0.030806968
```

```
qqnorm(train_model_rmOutliers$logSalePrice, main = "QQ Plot for logSalePrice"
)
qqline(train_model_rmOutliers$logSalePrice, col = "red")
```




```
# Going forward with the removed outliers
train_model <- train_model_rmOutliers
```

Model Selections

Forwards by AIC

```
gc() # free unused memory

##             used  (Mb) gc trigger  (Mb) limit (Mb) max used  (Mb)
## Ncells 2857710 152.7    4926097 263.1         NA  4926097 263.1
## Vcells 7098019  54.2   14786712 112.9      16384 12255556  93.6

set.seed(123)
# Train the model using forward feature selection
#forward on AIC
fit <- lm(logSalePrice ~ . - SalePrice - Id, data = train_model) # Exclude Sa
lePrice, ID

forward_result = ols_step_forward_aic(fit, details = FALSE)
selected_predictors <- forward_result$predictors
print(selected_predictors)

##  [1] "OverallQual"        "logGrLivArea"       "BsmtFinSF1"
##  [4] "YearBuilt"          "logLotArea"         "OverallCond"
```

```
##  [7] "TotalBsmtSF"        "GarageCars"         "FireplaceQuGd"
## [10] "GrLivArea"          "FoundationPConc"    "HeatingQCEx"
## [13] "Fireplaces"         "MSZoningRL"         "BedroomAbvGr"
## [16] "YearRemodAdd"       "BsmtQualGd"         "BsmtFullBath"
## [19] "GarageYrBlt"        "BsmtQualTA"         "HalfBath"
## [22] "BsmtFinType1Unf"    "YrSold"             "X2ndFlrSF"
## [25] "BsmtExposureNo"     "GarageArea"         "Exterior2ndVinylSd"
## [28] "WoodDeckSF"         "HeatingQCTA"        "MasVnrTypeBrkFace"
## [31] "MasVnrArea"         "HouseStyle2Story"
```

```
# Create the formula
response_variable <- "logSalePrice"
formula_str <- paste(response_variable, "~", paste(selected_predictors, colla
pse = " + "))
print(formula_str)
```

```
## [1] "logSalePrice ~ OverallQual + logGrLivArea + BsmtFinSF1 + YearBuilt +
logLotArea + OverallCond + TotalBsmtSF + GarageCars + FireplaceQuGd + GrLivAr
ea + FoundationPConc + HeatingQCEx + Fireplaces + MSZoningRL + BedroomAbvGr +
YearRemodAdd + BsmtQualGd + BsmtFullBath + GarageYrBlt + BsmtQualTA + HalfBat
h + BsmtFinType1Unf + YrSold + X2ndFlrSF + BsmtExposureNo + GarageArea + Exte
rior2ndVinylSd + WoodDeckSF + HeatingQCTA + MasVnrTypeBrkFace + MasVnrArea +
HouseStyle2Story"
```

```
forward_model <- train(as.formula(formula_str),
                       data = train_model,
                       trControl = trainControl(method="LOOCV"),
                       method = "lm")

# Make predictions on the test set using the trained model
predictions_log <- predict(forward_model, newdata = test_model)
predictions <- exp(predictions_log)

# Write the data frame to a CSV file
predictions_df <- data.frame(
  Id = as.numeric(names(predictions)),
  SalePrice = round(as.numeric(predictions))
)
write.csv(predictions_df, file = "Predictions/forward_model.csv", row.names =
FALSE)

#Model Stats
# Get the summary of the linear regression model
model_summary <- summary(forward_model$finalModel)
#adjusted R-squared value
fwd_r2 <- model_summary$adj.r.squared
# AIC from the model
fwd_aic <- (AIC(forward_model$finalModel))

forward_model$results
```

```
##   intercept      RMSE  Rsquared        MAE
## 1      TRUE 0.1208168 0.9083867 0.08544228
```

```
residuals <- model_summary$residuals
residuals_numeric <- unname(residuals)
fwd_press <- PRESS(residuals_numeric)
fwd_name <- "Forward Model:"
cat("Adjusted R2:", formatC(fwd_r2, width = 6, format = "f", flag = " "),
    " CV Press:", formatC(fwd_press, width = 6, format = "f", flag = " "),
    " AIC:", formatC(fwd_aic, width = 6, format = "f", flag = " "))
```

```
## Adjusted R2:  0.9109  CV Press:  20.2602  AIC: -2028.5811
```

Backwards on AIC

```
gc() # free unused memory
```

```
##           used  (Mb) gc trigger  (Mb) limit (Mb) max used  (Mb)
## Ncells 2881121 153.9    4926097 263.1         NA  4926097 263.1
## Vcells 8939726  68.3   17829725 136.1      16384 17829724 136.1
```

```
set.seed(123)
# Train the model using forward feature selection
#Backward on AIC
fit <- lm(logSalePrice ~ . - SalePrice - Id, data = train_model) # Exclude Sa
lePrice, ID

backward_result = ols_step_backward_aic(fit,details = FALSE)
selected_predictors <- backward_result$predictors
print(selected_predictors)
```

```
##  [1] "HouseStyle1Story"   "FireplaceQuNone"    "LotFrontage"
##  [4] "RoofStyleGable"     "Exterior2ndVinylSd" "MoSold"
##  [7] "GarageFinishFin"    "GarageFinishRFn"    "LotArea"
## [10] "MasVnrTypeNone"     "LotConfigInside"    "BsmtFinType1GLQ"
## [13] "FireplaceQuTA"      "FoundationCBlock"   "X1stFlrSF"
## [16] "ExterQualGd"        "ExterQualTA"        "LotShapeIR1"
## [19] "LotShapeReg"        "BsmtUnfSF"          "TotRmsAbvGrd"
## [22] "GarageTypeAttchd"   "GarageTypeDetchd"   "GarageFinishUnf"
## [25] "MSSubClass"         "OpenPorchSF"        "FullBath"
```

```
response_variable <- "logSalePrice"

# Create the formula
formula_str <- paste(response_variable, "~", paste(selected_predictors, colla
pse = " + "))
print(formula_str)
```

```
## [1] "logSalePrice ~ HouseStyle1Story + FireplaceQuNone + LotFrontage + Roo
fStyleGable + Exterior2ndVinylSd + MoSold + GarageFinishFin + GarageFinishRFn
+ LotArea + MasVnrTypeNone + LotConfigInside + BsmtFinType1GLQ + FireplaceQuT
A + FoundationCBlock + X1stFlrSF + ExterQualGd + ExterQualTA + LotShapeIR1 +
```

```
LotShapeReg + BsmtUnfSF + TotRmsAbvGrd + GarageTypeAttchd + GarageTypeDetchd
+ GarageFinishUnf + MSSubClass + OpenPorchSF + FullBath"

backward_model <- train(as.formula(formula_str),
                        data = train_model,
                        trControl = trainControl(method="LOOCV"),
                        method = "lm")


# Make predictions on the test set using the trained model
predictions_log <- predict(backward_model, newdata = test_model)
predictions <- exp(predictions_log)

# Write the data frame to a CSV file
backwards_predictions <- data.frame(
  Id = as.numeric(names(predictions)),
  SalePrice = round(as.numeric(predictions))
)
write.csv(backwards_predictions, file = "Predictions/backward_model.csv", row
.names = FALSE)

#Model Stats
# Get the summary of the linear regression model
model_summary <- summary(backward_model$finalModel)
#adjusted R-squared value
bkwd_r2 <- model_summary$adj.r.squared
# AIC from the model
bkwd_aic <- (AIC(backward_model$finalModel))

backward_model$results

##   intercept      RMSE  Rsquared       MAE
## 1      TRUE 0.1883986 0.7772606 0.1404231

residuals <- model_summary$residuals
residuals_numeric <- unname(residuals)
bkwd_press <- PRESS(residuals_numeric)
bkwd_name <- "Backward Model:"
cat("Adjusted R2:", formatC(bkwd_r2, width = 6, format = "f", flag = " "),
    " CV Press:", formatC(bkwd_press, width = 6, format = "f", flag = " "),
    " AIC:", formatC(bkwd_aic, width = 6, format = "f", flag = " "))

## Adjusted R2:  0.7841  CV Press:  49.2627  AIC: -744.0257
```

Stepwise on AIC

```
gc() # free unused memory

##            used  (Mb) gc trigger  (Mb) limit (Mb) max used  (Mb)
## Ncells  2890866 154.4    4926097 263.1         NA  4926097 263.1
## Vcells 10911174  83.3   21475670 163.9      16384 21475670 163.9
```

```r
set.seed(123)
# Train the model using Stepwise feature selection
fit <- lm(logSalePrice ~ . - SalePrice - Id, data = train_model) # Exclude Sa
lePrice, ID

stepwise_result = ols_step_both_aic(fit,details = FALSE)
selected_predictors <- stepwise_result$predictors
print(selected_predictors)
```

```
##  [1] "OverallQual"        "logGrLivArea"       "BsmtFinSF1"
##  [4] "YearBuilt"          "logLotArea"         "OverallCond"
##  [7] "TotalBsmtSF"        "GarageCars"         "FireplaceQuGd"
## [10] "GrLivArea"          "FoundationPConc"    "HeatingQCEx"
## [13] "Fireplaces"         "MSZoningRL"         "BedroomAbvGr"
## [16] "YearRemodAdd"       "BsmtQualGd"         "BsmtFullBath"
## [19] "GarageYrBlt"        "BsmtQualTA"         "HalfBath"
## [22] "BsmtFinType1Unf"    "YrSold"             "X2ndFlrSF"
## [25] "BsmtExposureNo"     "GarageArea"         "Exterior2ndVinylSd"
## [28] "WoodDeckSF"         "HeatingQCTA"        "MasVnrTypeBrkFace"
## [31] "MasVnrArea"         "HouseStyle2Story"
```

```r
response_variable <- "logSalePrice"

# Create the formula
formula_str <- paste(response_variable, "~", paste(selected_predictors, colla
pse = " + "))
print(formula_str)
```

```
## [1] "logSalePrice ~ OverallQual + logGrLivArea + BsmtFinSF1 + YearBuilt +
logLotArea + OverallCond + TotalBsmtSF + GarageCars + FireplaceQuGd + GrLivAr
ea + FoundationPConc + HeatingQCEx + Fireplaces + MSZoningRL + BedroomAbvGr +
YearRemodAdd + BsmtQualGd + BsmtFullBath + GarageYrBlt + BsmtQualTA + HalfBat
h + BsmtFinType1Unf + YrSold + X2ndFlrSF + BsmtExposureNo + GarageArea + Exte
rior2ndVinylSd + WoodDeckSF + HeatingQCTA + MasVnrTypeBrkFace + MasVnrArea +
HouseStyle2Story"
```

```r
stepwise_model <- train(as.formula(formula_str),
                        data = train_model,
                        trControl = trainControl(method="LOOCV"),
                        method = "lm",
                        direction = "both")

# Make predictions on the test set using the trained model
predictions_log <- predict(stepwise_model$final, newdata = test_model)
predictions <- exp(predictions_log)

# Write the data frame to a CSV file
stepwise_predict <- data.frame(
  Id = as.numeric(names(predictions)),
  SalePrice = round(as.numeric(predictions))
)
```

```r
write.csv(stepwise_predict, file = "Predictions/stepwise_model.csv", row.name
s = FALSE)


#Model Stats
# Get the summary of the linear regression model
model_summary <- summary(stepwise_model$finalModel)
#adjusted R-squared value
step_r2 <- model_summary$adj.r.squared
# AIC from the model
step_aic <- (AIC(stepwise_model$finalModel))

stepwise_model$results

##   intercept      RMSE  Rsquared        MAE
## 1      TRUE 0.1208168 0.9083867 0.08544228

residuals <- model_summary$residuals
residuals_numeric <- unname(residuals)
step_press <- PRESS(residuals_numeric)
step_name <- ("Stepwise Model:")
cat("Adjusted R2:", formatC(step_r2, width = 6, format = "f", flag = " "),
    "  CV Press:", formatC(step_press, width = 6, format = "f", flag = " "),
    "  AIC:", formatC(step_aic, width = 6, format = "f", flag = " "))

## Adjusted R2:  0.9109   CV Press:  20.2602   AIC: -2028.5811
```

####Custom model df modifications

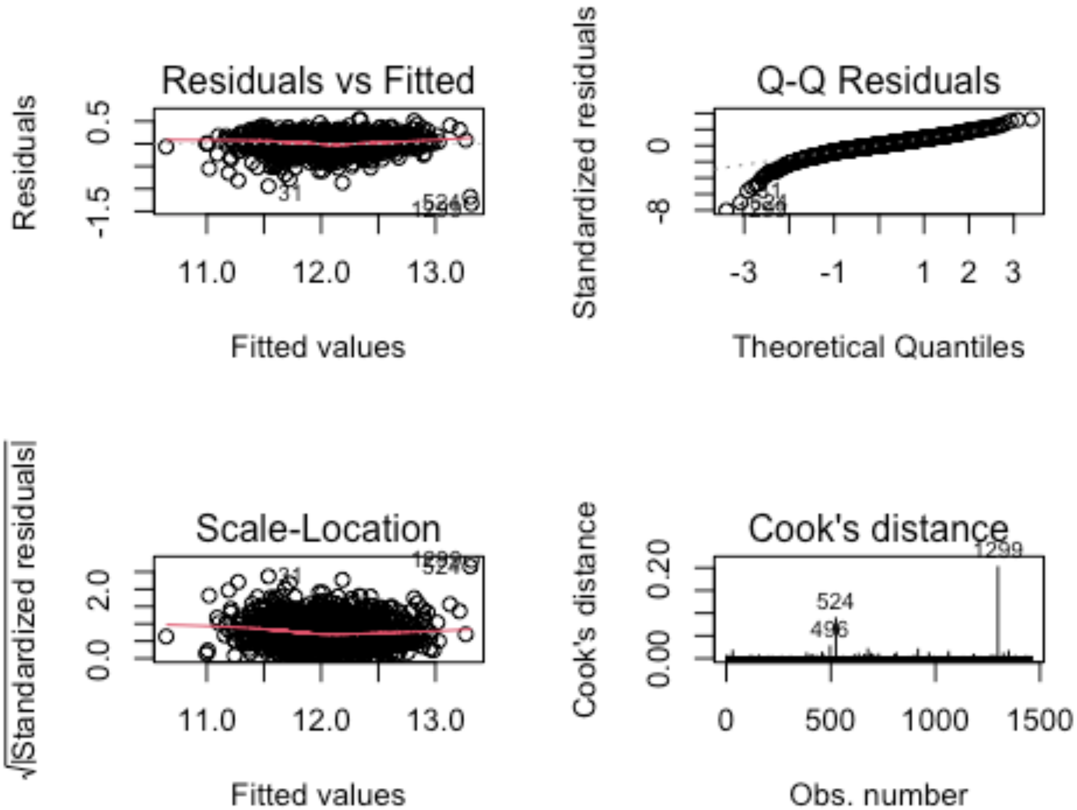
```r
custom_ames <- ames2

custom_ames$TotalBathrooms <- custom_ames$FullBath + (custom_ames$HalfBath *
0.5) + custom_ames$BsmtFullBath
custom_ames$HasGarage <- ifelse(custom_ames$GarageType != "None", 1, 0)

custom_ames$SeasonSold <- factor(
  cut(custom_ames$MoSold,
      breaks = c(0, 2, 5, 8, 11, 12),
      labels = c("Winter", "Spring", "Summer", "Autumn", "Winter")),
  levels = c("Winter", "Spring", "Summer", "Autumn")
)

custom_ames[] <- lapply(custom_ames, function(x) if (is.character(x)) as.fact
or(x) else x)

# Check the difference in column numbers
cat("Original number of columns:", ncol(custom_ames), "\n")

## Original number of columns: 84
```

```r
# Apply the function
custom_ames <- remove_low_variance_columns(custom_ames)
cat("Number of columns after removal:", ncol(custom_ames), "\n")

## Number of columns after removal: 54

train_model <- custom_ames[custom_ames$train=="train", ]
test_model <- custom_ames[custom_ames$train == "test", ]
```

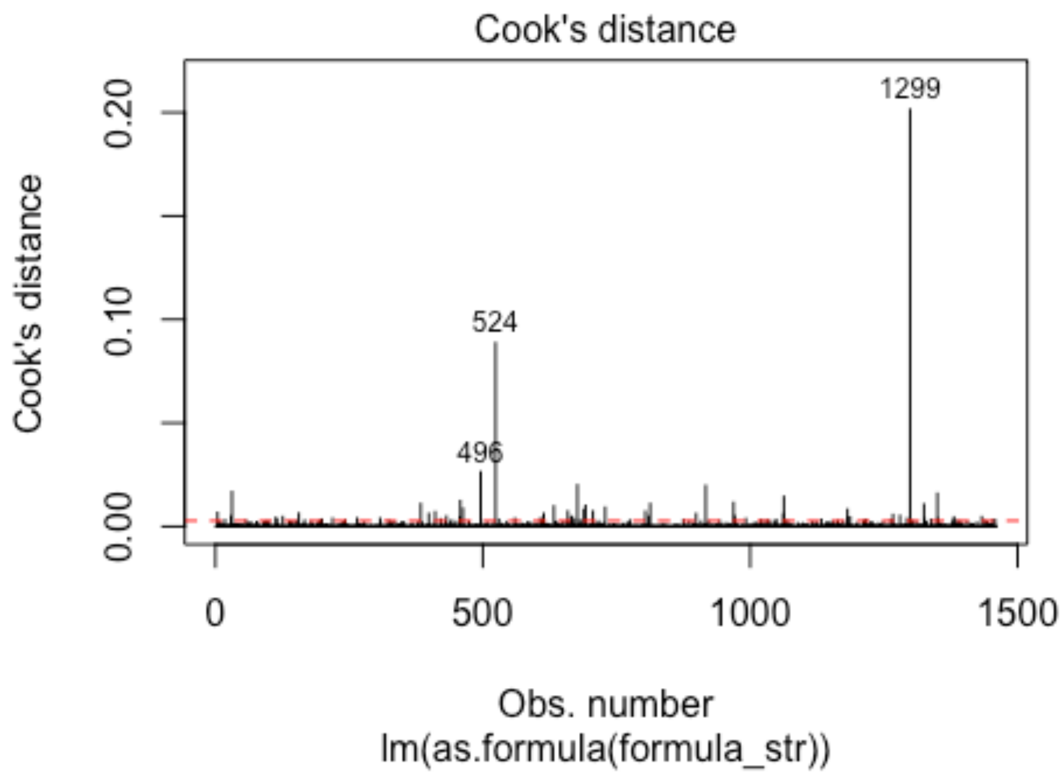
Custom - Assumptions and Influential Points

```r
#fit model 1 of log data
set.seed(123)

# Custom predictors
custom_predictors <- c("GarageCars", "OverallQual", "TotalBathrooms", "Season
Sold", "logLotArea", "logGrLivArea") #Has garage removed due to low variance

# Create the formula
response_variable <- "logSalePrice"
formula_str <- paste(response_variable, "~", paste(custom_predictors, collaps
e = " + "))

fit <- lm(as.formula(formula_str), data = train_model)
summary(fit)

##
## Call:
## lm(formula = as.formula(formula_str), data = train_model)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.33091 -0.08163  0.00807  0.10232  0.54683
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        8.115939   0.126910  63.951   <2e-16 ***
## GarageCars         0.094072   0.007718  12.189   <2e-16 ***
## OverallQual        0.137556   0.004598  29.920   <2e-16 ***
## TotalBathrooms     0.090245   0.007342  12.292   <2e-16 ***
## SeasonSoldSpring   0.017103   0.015198   1.125    0.261
## SeasonSoldSummer   0.023374   0.014651   1.595    0.111
## SeasonSoldAutumn  -0.006323   0.017052  -0.371    0.711
## logLotArea         0.121108   0.009369  12.926   <2e-16 ***
## logGrLivArea       0.218574   0.019440  11.243   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1681 on 1451 degrees of freedom
## Multiple R-squared:  0.8239, Adjusted R-squared:  0.8229
## F-statistic: 848.5 on 8 and 1451 DF,  p-value: < 2.2e-16
```

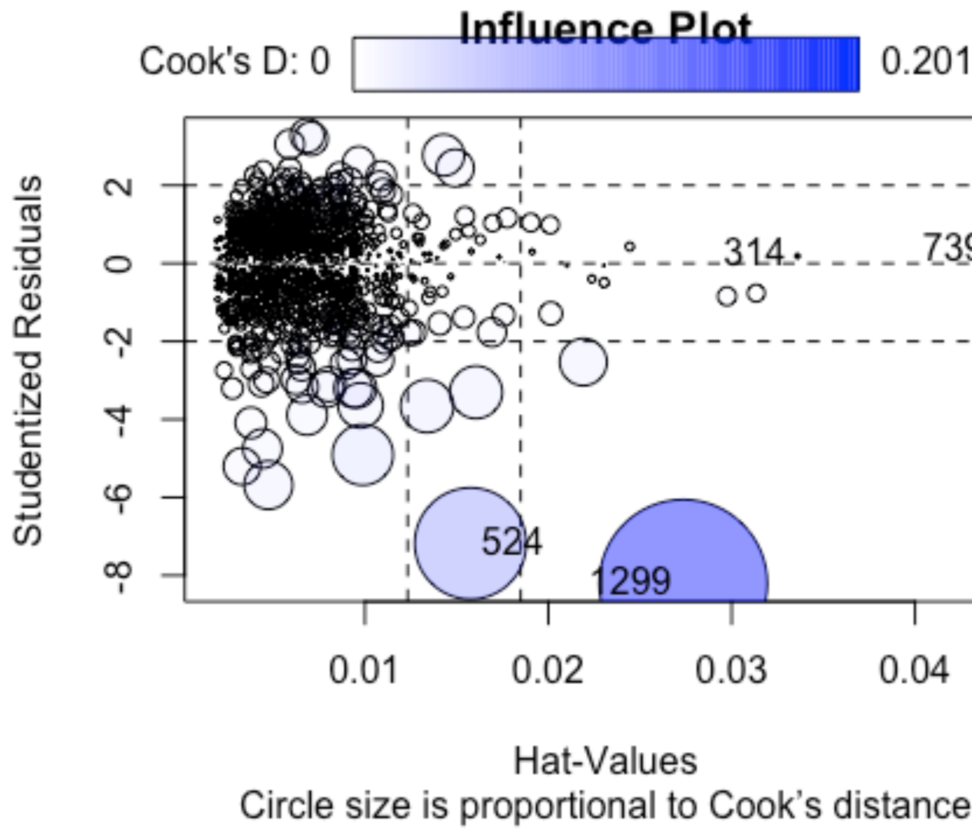
```r
#Check the residuals
par(mfrow=c(2,2)) # Set up a 2x2 plot grid
plot(fit, which = 1) # Residuals vs Fitted
plot(fit, which = 2) # Normal Q-Q
plot(fit, which = 3) # Scale-Location
plot(fit, which = 4) # Cook's distance
```




```r
par(mfrow = c(1, 1)) # Set up a 1x1 plot grid

# ols_plot_diagnostics(fit)

#Cooks D Plot
cutoff <- 4/(nrow(train_model)-length(fit$coefficients)-2)
plot(fit, which=4, cook.levels=cutoff)
abline(h=cutoff, lty=2, col="red")
```




```
#Influence Plot
influencePlot(fit, id.method="identify", main="Influence Plot", sub="Circle s
ize is proportional to Cook’s distance")
```




```
##         StudRes        Hat       CookD
## 314   0.1871802 0.03358052 0.0001353592
## 524  -7.1819693 0.01575666 0.0886594437
## 739   0.3551386 0.04442235 0.0006518540
## 1299 -8.2098350 0.02735909 0.2014384273

#Influencial points 496, 524, 1299
```

Removing Influential outliers

```
#Outliers Addressed/Observation 339 influential point/outlier removed.
outliers_to_remove <- c(496,524,1299)
train_model_rmOutliers <- train_model[-outliers_to_remove,]
str(train_model)

## 'data.frame':    1460 obs. of  54 variables:
##  $ Id           : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ MSSubClass   : int  60 20 60 70 60 50 20 60 50 190 ...
##  $ MSZoning     : Factor w/ 6 levels "C (all)","FV",..: 5 5 5 5 5 5 5 5 6
5 ...
##  $ LotFrontage  : num  65 80 68 60 84 85 75 0 51 50 ...
##  $ LotArea      : int  8450 9600 11250 9550 14260 14115 10084 10382 6120
7420 ...
##  $ LotShape     : Factor w/ 4 levels "IR1","IR2","IR3",..: 4 4 1 1 1 1 4
1 4 4 ...
```

```
##  $ LotConfig     : Factor w/ 5 levels "Corner","CulDSac",..: 5 3 5 1 3 5 5
1 5 1 ...
##  $ Neighborhood  : Factor w/ 25 levels "Blmngtn","Blueste",..: 6 25 6 7 14
12 21 17 18 4 ...
##  $ HouseStyle    : Factor w/ 8 levels "1.5Fin","1.5Unf",..: 6 3 6 6 6 1 3
6 1 2 ...
##  $ OverallQual   : int  7 6 7 7 8 5 8 7 7 5 ...
##  $ OverallCond   : int  5 8 5 5 5 5 5 6 5 6 ...
##  $ YearBuilt     : int  2003 1976 2001 1915 2000 1993 2004 1973 1931 1939
...
##  $ YearRemodAdd  : int  2003 1976 2002 1970 2000 1995 2005 1973 1950 1950
...
##  $ RoofStyle     : Factor w/ 6 levels "Flat","Gable",..: 2 2 2 2 2 2 2 2 2
2 ...
##  $ Exterior1st   : Factor w/ 16 levels "AsbShng","AsphShn",..: 14 9 14 15
14 14 14 7 4 9 ...
##  $ Exterior2nd   : Factor w/ 17 levels "AsbShng","AsphShn",..: 15 9 15 17
15 15 15 7 17 9 ...
##  $ MasVnrType    : Factor w/ 4 levels "BrkCmn","BrkFace",..: 2 3 2 3 2 3 4
4 3 3 ...
##  $ MasVnrArea    : num  196 0 162 0 350 0 186 240 0 0 ...
##  $ ExterQual     : Factor w/ 4 levels "Ex","Fa","Gd",..: 3 4 3 4 3 4 3 4 4
4 ...
##  $ Foundation    : Factor w/ 6 levels "BrkTil","CBlock",..: 3 2 3 1 3 6 3
2 1 1 ...
##  $ BsmtQual      : Factor w/ 5 levels "Ex","Fa","Gd",..: 3 3 3 5 3 3 1 3 5
5 ...
##  $ BsmtExposure  : Factor w/ 5 levels "Av","Gd","Mn",..: 4 2 3 4 1 4 1 3 4
4 ...
##  $ BsmtFinType1  : Factor w/ 7 levels "ALQ","BLQ","GLQ",..: 3 1 3 1 3 3 3
1 7 3 ...
##  $ BsmtFinSF1    : num  706 978 486 216 655 ...
##  $ BsmtUnfSF     : num  150 284 434 540 490 64 317 216 952 140 ...
##  $ TotalBsmtSF   : num  856 1262 920 756 1145 ...
##  $ HeatingQC     : Factor w/ 5 levels "Ex","Fa","Gd",..: 1 1 1 3 1 1 1 1 3
1 ...
##  $ X1stFlrSF     : int  856 1262 920 961 1145 796 1694 1107 1022 1077 ...
##  $ X2ndFlrSF     : int  854 0 866 756 1053 566 0 983 752 0 ...
##  $ GrLivArea     : int  1710 1262 1786 1717 2198 1362 1694 2090 1774 1077
...
##  $ BsmtFullBath  : int  1 0 1 1 1 1 1 1 0 1 ...
##  $ FullBath      : int  2 2 2 1 2 1 2 2 2 1 ...
##  $ HalfBath      : int  1 0 1 0 1 1 0 1 0 0 ...
##  $ BedroomAbvGr  : int  3 3 3 3 4 1 3 3 2 2 ...
##  $ KitchenQual   : Factor w/ 5 levels "Ex","Fa","Gd",..: 3 5 3 3 3 5 3 5 5
5 ...
##  $ TotRmsAbvGrd  : int  8 6 6 7 9 5 7 7 8 5 ...
##  $ Fireplaces    : int  0 1 1 1 1 0 1 2 2 2 ...
##  $ FireplaceQu   : Factor w/ 6 levels "Ex","Fa","Gd",..: 4 6 6 3 6 4 3 6 6
6 ...
```

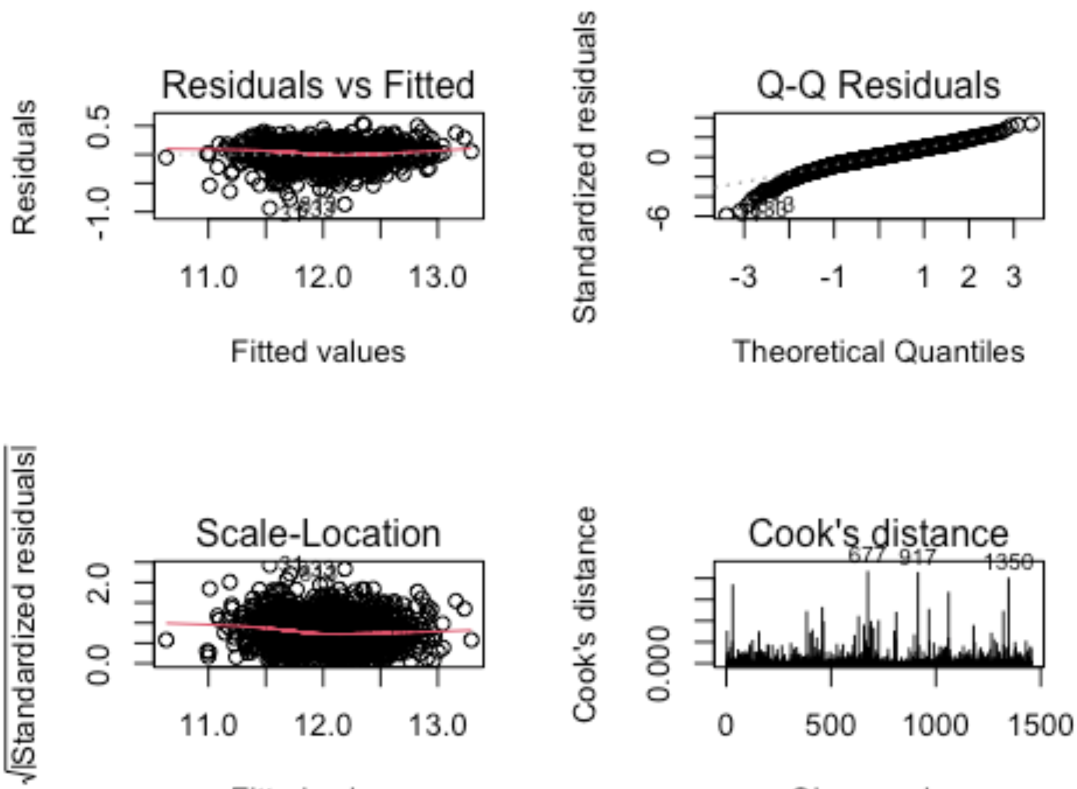
```
##  $ GarageType    : Factor w/ 7 levels "2Types","Attchd",..: 2 2 2 6 2 2 2 
2 6 2 ...
##  $ GarageYrBlt   : num  2003 1976 2001 1998 2000 ...
##  $ GarageFinish  : Factor w/ 4 levels "Fin","None","RFn",..: 3 3 3 4 3 4 3 
3 4 3 ...
##  $ GarageCars    : num  2 2 2 3 3 2 2 2 2 1 ...
##  $ GarageArea    : num  548 460 608 642 836 480 636 484 468 205 ...
##  $ WoodDeckSF    : int  0 298 0 0 192 40 255 235 90 0 ...
##  $ OpenPorchSF   : int  61 0 42 35 84 30 57 204 0 4 ...
##  $ MoSold        : int  2 5 9 2 12 10 8 11 4 1 ...
##  $ YrSold        : int  2008 2007 2008 2006 2008 2009 2007 2009 2008 2008 
...
##  $ SalePrice     : int  208500 181500 223500 140000 250000 143000 307000 2
00000 129900 118000 ...
##  $ train         : Factor w/ 2 levels "test","train": 2 2 2 2 2 2 2 2 2 2 
...
##  $ logLotArea    : num  9.04 9.17 9.33 9.16 9.57 ...
##  $ logGrLivArea  : num  7.44 7.14 7.49 7.45 7.7 ...
##  $ logSalePrice  : num  12.2 12.1 12.3 11.8 12.4 ...
##  $ TotalBathrooms: num  3.5 2 3.5 2 3.5 2.5 3 3.5 2 2 ...
##  $ SeasonSold    : Factor w/ 4 levels "Winter","Spring",..: 1 2 4 1 1 4 3 
4 2 1 ...
```

```
#Verify Assumption again
refit <- lm(as.formula(formula_str), data = train_model_rmOutliers)
summary(refit)
```

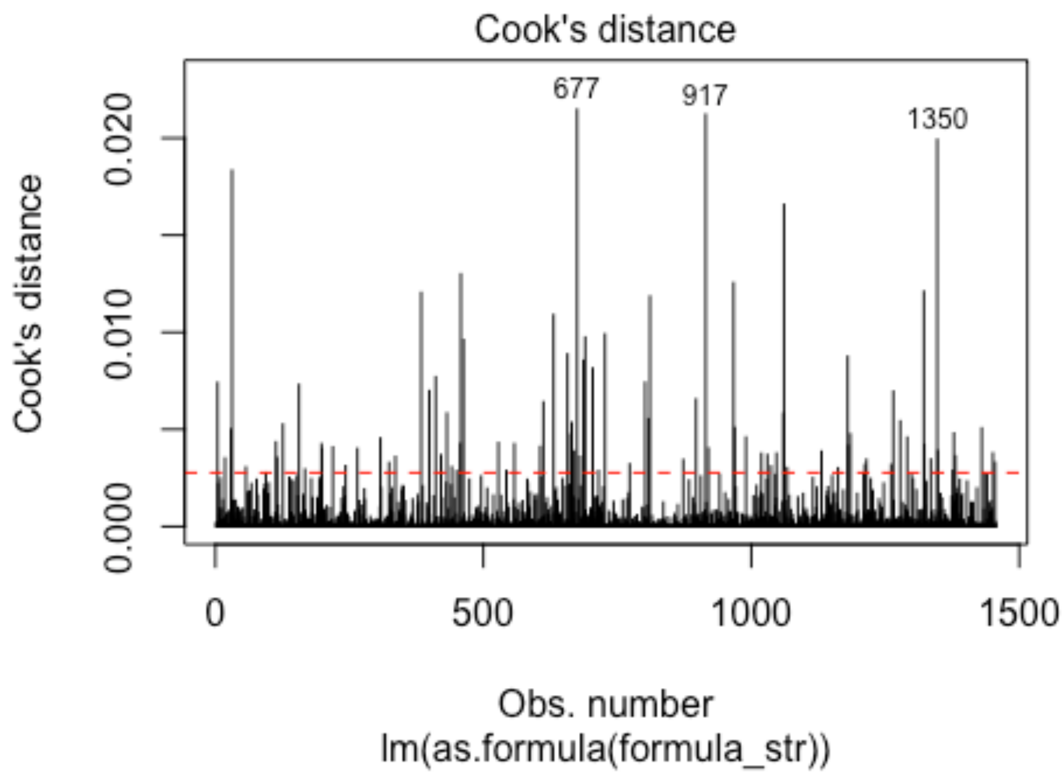
```
## 
## Call:
## lm(formula = as.formula(formula_str), data = train_model_rmOutliers)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.93897 -0.08328  0.00574  0.10200  0.53691 
## 
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)    
## (Intercept)       7.960643   0.121770  65.374   <2e-16 ***
## GarageCars        0.086270   0.007375  11.697   <2e-16 ***
## OverallQual       0.140031   0.004380  31.969   <2e-16 ***
## TotalBathrooms    0.093053   0.006991  13.310   <2e-16 ***
## SeasonSoldSpring  0.009921   0.014491   0.685    0.494    
## SeasonSoldSummer  0.015373   0.013976   1.100    0.272    
## SeasonSoldAutumn -0.006163   0.016289  -0.378    0.705    
## logLotArea        0.130709   0.008952  14.602   <2e-16 ***
## logGrLivArea      0.227967   0.018546  12.292   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.16 on 1448 degrees of freedom
```

```
## Multiple R-squared:  0.8391, Adjusted R-squared:  0.8382
## F-statistic: 944.1 on 8 and 1448 DF,  p-value: < 2.2e-16
```

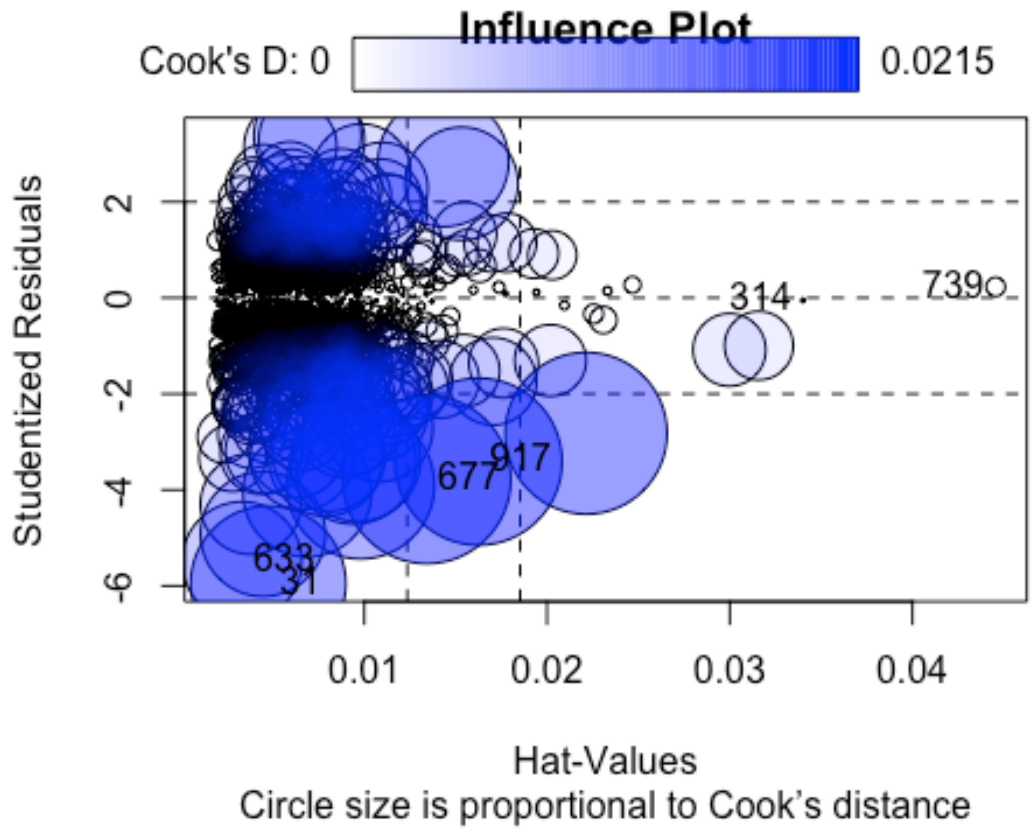
```
#Check the residuals
par(mfrow=c(2,2)) # Set up a 2x2 plot grid
plot(refit, which = 1) # Residuals vs Fitted
plot(refit, which = 2) # Normal Q-Q
plot(refit, which = 3) # Scale-Location
plot(refit, which = 4) # Cook's distance
```




```
par(mfrow = c(1, 1)) # Set up a 1x1 plot grid

# ols_plot_diagnostics(refit)

#Cooks D Plot
cutoff <- 4/(nrow(train_model_rmOutliers)-length(refit$coefficients)-2)
plot(refit, which=4, cook.levels=cutoff)
abline(h=cutoff, lty=2, col="red")
```

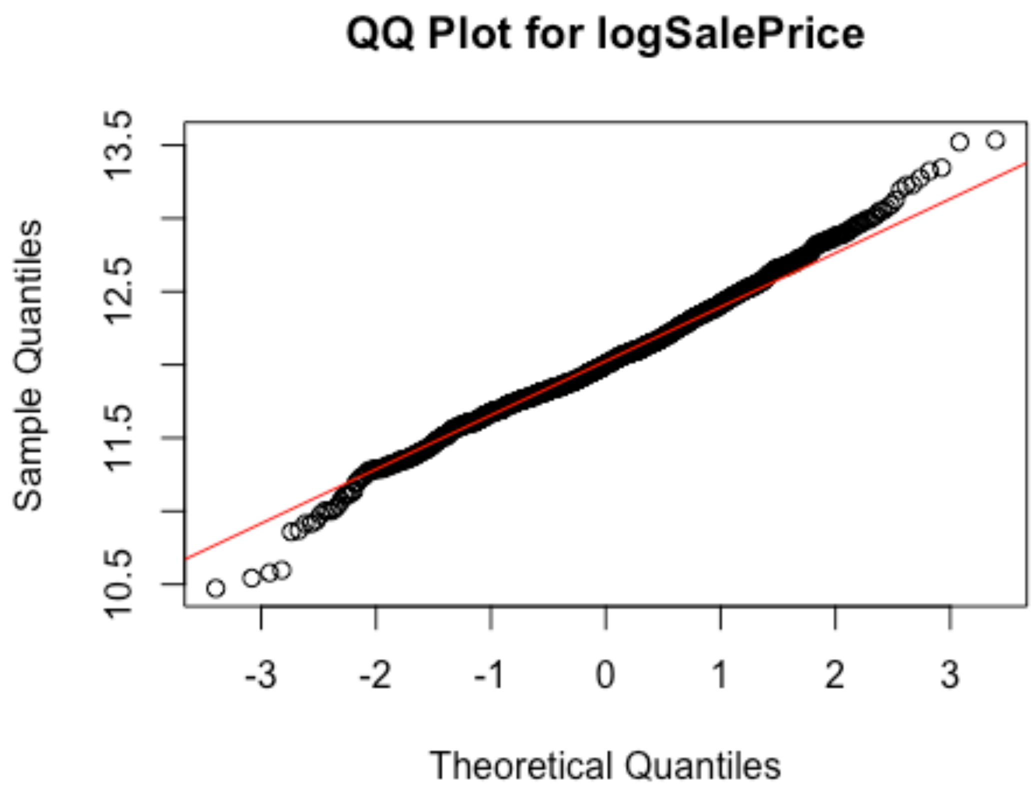



```
#Influence Plot
influencePlot(refit, id.method="identify", main="Influence Plot", sub="Circle
size is proportional to Cook’s distance")
```




```
##         StudRes         Hat        CookD
## 31  -5.95320104 0.004744067 1.833437e-02
## 314 -0.05192964 0.034012887 1.055746e-05
## 633 -5.49399543 0.003303825 1.089739e-02
## 677 -3.78684080 0.013419395 2.147480e-02
## 739  0.22932687 0.044553410 2.726630e-04
## 917 -3.41232637 0.016252371 2.121832e-02
```

```
qqnorm(train_model_rmOutliers$logSalePrice, main = "QQ Plot for logSalePrice"
)
qqline(train_model_rmOutliers$logSalePrice, col = "red")
```




```
# Going forward with the removed outliers
train_model <- train_model_rmOutliers
```

###Train Custom Model

```
gc() # free unused memory

##            used  (Mb) gc trigger  (Mb) limit (Mb) max used  (Mb)
## Ncells  2899747 154.9    4926097 263.1         NA  4926097 263.1
## Vcells 12399411  94.6   21475670 163.9      16384 21475670 163.9

set.seed(123)

# Custom predictors
custom_predictors <- c("GarageCars", "OverallQual", "TotalBathrooms", "Season
Sold", "logLotArea", "logGrLivArea")

# Create the formula
response_variable <- "logSalePrice"
formula_str <- paste(response_variable, "~", paste(custom_predictors, collaps
e = " + "))
print(formula_str)
```

```
## [1] "logSalePrice ~ GarageCars + OverallQual + TotalBathrooms + SeasonSold
+ logLotArea + logGrLivArea"

# Train the model
custom_model <- train(as.formula(formula_str),
                      data = train_model,
                      trControl = trainControl(method="LOOCV"),
                      method = "lm")

# Summary of the model
summary(custom_model)

##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.93897 -0.08328  0.00574  0.10200  0.53691
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       7.960643   0.121770  65.374   <2e-16 ***
## GarageCars        0.086270   0.007375  11.697   <2e-16 ***
## OverallQual       0.140031   0.004380  31.969   <2e-16 ***
## TotalBathrooms    0.093053   0.006991  13.310   <2e-16 ***
## SeasonSoldSpring  0.009921   0.014491   0.685    0.494
## SeasonSoldSummer  0.015373   0.013976   1.100    0.272
## SeasonSoldAutumn -0.006163   0.016289  -0.378    0.705
## logLotArea        0.130709   0.008952  14.602   <2e-16 ***
## logGrLivArea      0.227967   0.018546  12.292   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.16 on 1448 degrees of freedom
## Multiple R-squared:  0.8391, Adjusted R-squared:  0.8382
## F-statistic: 944.1 on 8 and 1448 DF,  p-value: < 2.2e-16

# Make predictions on the test set using the trained model
predictions_log <- predict(custom_model, newdata = test_model)
predictions <- exp(predictions_log)

# Write the data frame to a CSV file
predictions_df <- data.frame(
  Id = as.numeric(names(predictions)),
  SalePrice = round(as.numeric(predictions))
)
write.csv(predictions_df, file = "Predictions/custom_model.csv", row.names =
FALSE)
```

```r
#Model Stats
# Get the summary of the linear regression model
model_summary <- summary(custom_model$finalModel)
#adjusted R-squared value
cust_r2 <- model_summary$adj.r.squared
# AIC from the model
cust_aic <- (AIC(custom_model$finalModel))

custom_model$results

##   intercept      RMSE  Rsquared       MAE
## 1      TRUE 0.1605397 0.8369769 0.1190876

residuals <- model_summary$residuals
residuals_numeric <- unname(residuals)
cust_press <- PRESS(residuals_numeric)
cust_name <- ("Custom Model:")
cat("Adjusted R2:", formatC(cust_r2, width = 6, format = "f", flag = " "),
    " CV Press:", formatC(cust_press, width = 6, format = "f", flag = " "),
    " AIC:", formatC(cust_aic, width = 6, format = "f", flag = " "))

## Adjusted R2:  0.8382  CV Press:  37.1059  AIC: -1194.9275
```

Model Results

```r
Model_Results <- data.frame(
  Model = c(fwd_name, bkwd_name, step_name, cust_name),
  `Adjusted R2` = c(fwd_r2, bkwd_r2, step_r2, cust_r2),
  `CV PRESS` = c(fwd_press, bkwd_press, step_press, cust_press),
  AIC = c(fwd_aic, bkwd_aic, step_aic, cust_aic)
)

kable(Model_Results, caption = "Model Results")
```

Model Results

Model	Adjusted.R2	CV.PRESS	AIC
Forward Model:	0.9108818	20.26023	-2028.5811
Backward Model:	0.7840674	49.26270	-744.0257
Stepwise Model:	0.9108818	20.26023	-2028.5811
Custom Model:	0.8382415	37.10588	-1194.9275