House Prices Regression Analysis

Duy Nguyen, Jordan Eaddy and Leonardo Leal Filho
Southern Methodist University

1 Introduction

What determines the value of a house? Our final project for the Statistical Foundations course in the Southern Methodist University Master of Science of Data Science program was to challenge us to predict the final price of each home in Ames, Iowa, making regression analyses using what we have learned in the course.

2 Ames, Iowa Housing Data Set

The data set used in this project describes the sale of individual residential property in Ames, Iowa from 2006 to 2010[1]. It was retrieved from data set hosting website Kaggle, where it is listed as a machine learning competition called *House Prices - Advanced Regression Techniques*[2] . The data involves both a training set and a testing set, which contains 1460 observations with 81 variables and 1459 observations with 80 variables, respectively. The first analysis estimates the sales price of a house (`SalePrice`) as the response of the square footage of the living area (`GrLivArea`) of the house and the neighborhood (`Neighborhood`) where it resides. The second analysis uses several variable selection and elimination techniques to determine a combination of variables that can result in the most desired house price prediction model.

3 Analysis Question 1

3.1 Restatement of Problem

Century 21 Ames had commissioned a relationship analysis between the sales price of houses in the NAmes, Edwards and BrkSide neighborhoods and the square footage of the living area of such houses, and particularly if the sales price (and its relationship to square footage) depends on which neighborhood that the house is located in.

Provided in this section is the evidence to determine if certain model assumptions are met and methods to deal with potential outliers / influential observations, along with model estimates as well as their confidence intervals. Remarks about the living area will be kept within increments of 100 sq. ft. for purposes of interpretation.

3.2 Modeling

The explicit data of both `SalesPrice` and `GrLivArea` show evidence of a linear relationship [*see Section 5.1*]. However, improvements can be made such as log-transforming both of our variables of interest as there are numerous outliers that are affecting linearity. A more uniform distribution of points can be seen forming a diagonal line across the graph after the transformation, revealing a stronger relationship between the two variables.

Another improvement can be made by removing highly influential observations that are greater than 0.1 Cook's D from the model, one by one as well as in combinations while judging for the R^2 and adjusted R^2, allowing for more validity in our regression analysis. We came to a conclusion that removing observations 131 and 339 resulted in the best R^2 and adjusted R^2 of 0.443012 and 0.441542, respectively [*see Section 5.2*].

Model 1

$$log(SalePrice) = \hat{\beta}_0 + \hat{\beta}_1 log(GrLivArea)$$

We would like to see if adding `Neighborhood` into the regression will produce a more significant second model. We first started with an additive model with only `Neighborhood` added and judged its R^2 and adjusted R^2, and sequentially an interactive model with an adjusted slope for `Neighborhood` and `log(GrLivArea),` as well as one with `Neighborhood` and `GrLivArea`. The latter model stands victorious with an R^2 and adjusted R^2 of 0.531084 and 0.521632, respectively. We then concluded that this model would be our second model, where the Brookside neighborhood was used for reference.

Model 2

$$\begin{aligned} log(SalePrice) = \hat{\beta}_0 + \hat{\beta}_1 log(GrLivArea) + \hat{\beta}_2 Edwards + \hat{\beta}_3 NAmes \\ + \hat{\beta}_4 Edwards * GrLivArea + \hat{\beta}_5 NAmes * GrLivArea \end{aligned}$$

3.3 Assumptions

After the removal of observations 131 and 339, while there is little evidence against normality and skewness from the QQ plot and histogram, there is little evidence of SD from the studentized scatterplots. We are able to see an outlier, observation 135, from the Cook's D and leverage plots that we can remove from our second model. However, after removing observation 135 and judging the R-Squared and adjusted R-Squared of the model, we concluded that it is best to leave observation 135 in the data. We will assume that the observations are independent [*see Appendix 5.3*].

3.4 Comparing Competing Models

An extra sums of square test is conducted to compare the significance between our two models. There is strong evidence suggesting that the second model is more significant (p-value = 1.004e-12).

We also ran a 10-fold cross-validations repeated for 5 times on both models, and have concluded that the second model is more significant, which is consistent with our extra sums of square test [*see Appendix 5.4*].

3.5 Conclusion

Complete Regression Model

$$\begin{aligned} log(SalePrice) &= 5.913 + 0.82\ log(GrLivArea) + 1.01\ Edwards + 2.58\ NAmes \\ &\quad -0.146\ log(GrLivArea) * Edwards - 0.347\ log(GrLivArea) * NAmes \end{aligned}$$

The data suggests that a doubling of the square footage of living area of houses increases the estimated median of its sales price by $2^{.82} - 1 = 76.514\%$. We are 95% confident that the multiplicative increase in median sales price of houses after a doubling of its square footage of living area is $(2^{0.681} - 1, 2^{0.958} - 1) = (60.33\%, 94.26\%)$. Model 2 is a good fit for the data (p-value < 2.2E-16 for F-test with df(5, 375)). It is estimated that the square footage of living area explains about 52.79% of the variation in sales price of houses in Ames, Iowa between 2006 and 2010 [*see Section 5.5*].

The separate equations that represent the model of each neighborhood is below:

$$\{log(SalePrice) \mid BrkSide\} = 5.913 + 0.82\ log(\ GrLivArea)$$

$$\{log(SalePrice) \mid Edwards\} = 6.923 + 0.674\ log(GrLivArea)$$

$$\{log(SalePrice) \mid NAmes\} = 8.493 + 0.473\ log(GrLivArea)$$

We are 95% confident that for every doubling of the square footage of living area of a house in:

- Brookside, the estimated median sales price increases by $2^{.82} - 1 = 76.51\%$.
- Edwards, the estimated median sales price increases by $2^{.604} - 1 = 56.59\%$.
- North Ames, the estimated median sales price increases by $2^{.473} - 1 = 38.80\%$.

Finally, since this is an observational study, we cannot make any causal inferences. However, we can imply that for the houses in Ames, Iowa during 2006 and 2010, increases in square footage of living area does relate to increases in sales price.

4 Analysis Question 2

4.1 Restatement of Problem

In this analysis, we are asked to build the most predictive model for sales prices of homes in all neighborhoods of Ames, Iowa, using both the train and test datasets retrieved from the Kaggle website..

Our strategy is to produce four selection models: one from forward selection, one from backwards elimination, one from stepwise selection, and one that is built custom. We will also evaluate the adjusted R-Squared, internal CV Press, and the Kaggle score for each of the said four models.

4.2 Modeling

After much discussion amongst our team, we decided to test an initial model with all the available variables, or rather data features, in the Ames, Iowa housing data set which is contained within a text file named datadescription.txt retrieved from the Kaggle website. This dataset has a missing values percentage, or in this case Not Available (NA), of 5.88% along with some special cases which will be further discussed below. All of these are deemed crucial to be either removed or adjusted accordingly for our analysis [*see Section 5.6*].

While going down the list of all available variables of the initial model and judging for its R-Squared value, the sequential steps to clean the data consists of the following:

- NA numerical variables are replaced with zeros.
- NA non-numerical variables, or in this case character variables, are replaced with "None" to represent the non-available feature for that variable.
- Character variables are converted into factor variables, which proves to improve our prediction analysis as there is now one reference for each factor variable for RStudio to recognize and use.
- One influential observation, data point 524, was detected with a standout Cook's D and leverage. We decided to ultimately remove this observation since it may hinder our prediction analysis [*see Appendix 5.7*].
- The variables Id, MSSubClass, MSZoning, Utilities, Condition2, Exterior1st, Exterior2nd, MasVnrType, KitchenQual, Functional, SaleType and PoolQC contain levels in the testing dataset that does not exist in the training dataset. Since those are categorical variables, all of the variables need to be used in the training phase of the model. Hence, since we cannot train for levels not in the training dataset, those variables are not to be used while training the model.
- At this time, the variables OverallQual and OverallCond are converted mistakenly into numeric variables as they appear to be. These two variables actually proved to be more useful for our analysis once they are turned into factor variables to better represent the data.
- SalePrice is log-transformed to be used as the response for our models since previously its data was heavily right skewed [*see Appendix 5.8*].

4.3 Assumptions

As mentioned previously, influential observation 524 was removed to clean up some of the noise we were seeing. Our plots suggest that there is minimal evidence against normality when looking at the histograms [*see Appendix 5.9*]. And looking at the QQ plots, there are signs of data that is peaking in the middle of the graph. We will assume that our data is independent.

4.4 Model Selection

Linear Regression Model

$$log(y) = \hat{\beta}_0 + \hat{\beta}_1(x_1) + \ldots + \hat{\beta}_{n-1}x_{n-1} + \hat{\beta}_n x_n$$

The goal of the forward, backward and stepwise selections is to generate a linear regression model as seen above. A custom model is also included to predict the log-transformed `SalePrice`. The strategy for this custom model is to start with the predicting variables from both our forward and stepwise selection models, which yielded the highest and identical adjusted R-squares, and adjusting the coefficient of the numerical variables with the `Neighborhood` factor variable.

Forward & Stepwise Selection Model

Since the forward and stepwise selection yielded the same model, consequently they picked the same predictors. The forward selection model was built with a simple model and adding variables as they fit the p-value criteria. This means that variables with a p-value less than 0.01 would enter the final model. The stepwise select uses the steps of the forward and backward selection, where it adds and removes variables according to the p-value criteria. This means that it would add variables with a p-value less than 0.01 and removes those variables with a p-value above 0.01 [*see Appendix 5.10*].

Backward Selection Model

The backward selection was built with all of the predictors in it and it removes those that did not meet the p-value criteria. This means that variables with a p-value above 0.01 would be removed from the model.

As mentioned earlier, the custom model utilized the predictors derived from the backward selection and adjusted the coefficients of the numerical variables with the Neighborhood variable [*see Appendix 5.11*].

4.5 Comparing Competing Models

Predictive Models	Adjusted R^2	SSR	Kaggle Score
Forward	0.927	15.79319	0.30336
Backward	0.9267	15.89446	0.30297
Stepwise	0.927	15.79319	0.30336
Custom	0.9357	10.89817	4.50665

It is worth noting that the forward and stepwise selection models produced the identical predictors. That is why they have the same scores throughout the table. We would also like to mention that since the stepwise selection had the highest R^2 and adjusted R^2, the custom model taking the same predictors used in that model and adding an adjustment to the coefficient of the numerical variables with the Neighborhood categorical variable.

4.5 Conclusion

We believe our custom model is a good fit for the data (p-value < 2.2E-16 for F-test with df(396, 1062)). It is estimated that our model explains about 95.32% of the variation in sales price of houses in Ames, Iowa between 2006 and 2010.

Our custom model produced the best scores [*see Appendix 5.12*]. It had the highest R^2 and $adjusted\ R^2$ as well as the lowest sum of squared residuals. Interestingly, however, its kaggle score was much higher than that of any of the models created via a form of selection. The best kaggle score in fact came from the backward selection model, which yielded the lowest highest R^2 and $adjusted\ R^2$ as well as the highest sum of squared residuals.

We believe that the best approach to predicting house prices in this situation is an inspection of multiple selection methods and visualization to remove the appropriate outliers.

Perhaps a different machine learning technique, which we will learn at a later course, can provide us with clearer proofs of effectiveness within the training dataset.

5 Appendix

5.1 Checking for Linearity between `SalePrice` and `GrLivArea`

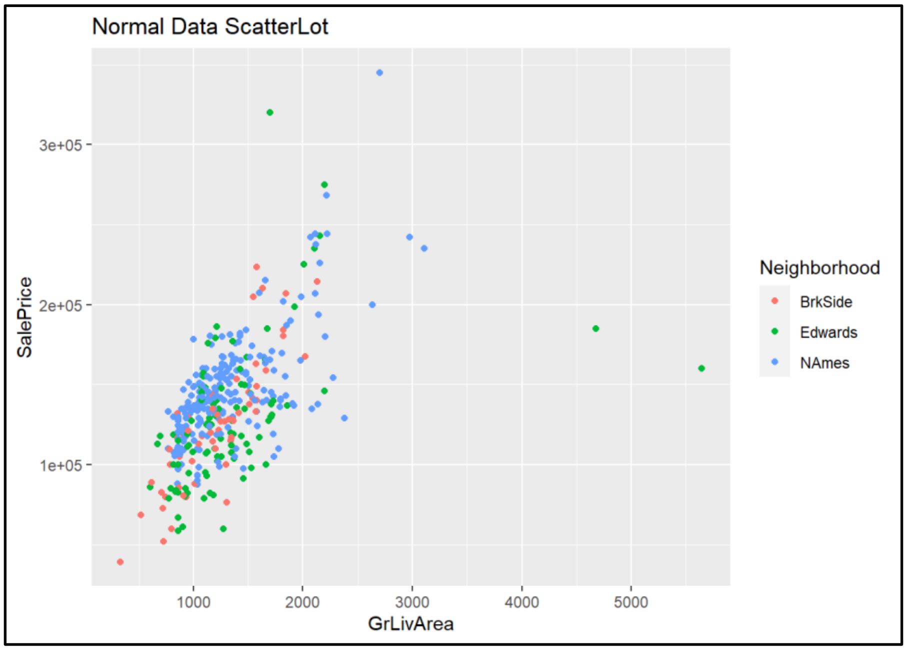


Figure 1

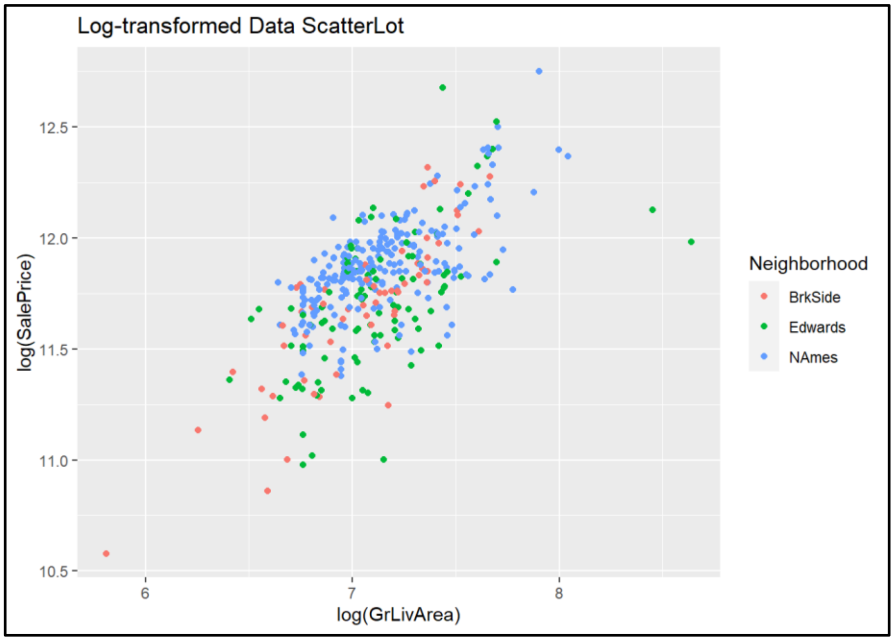


Figure 2

5.2 Fit Assessment of Model 1

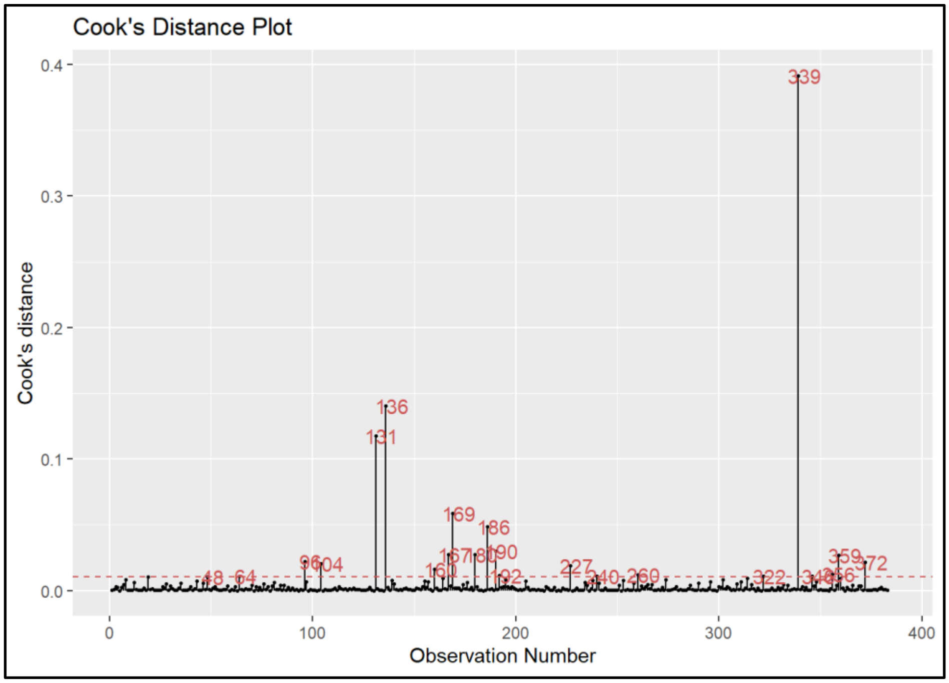


Figure 3

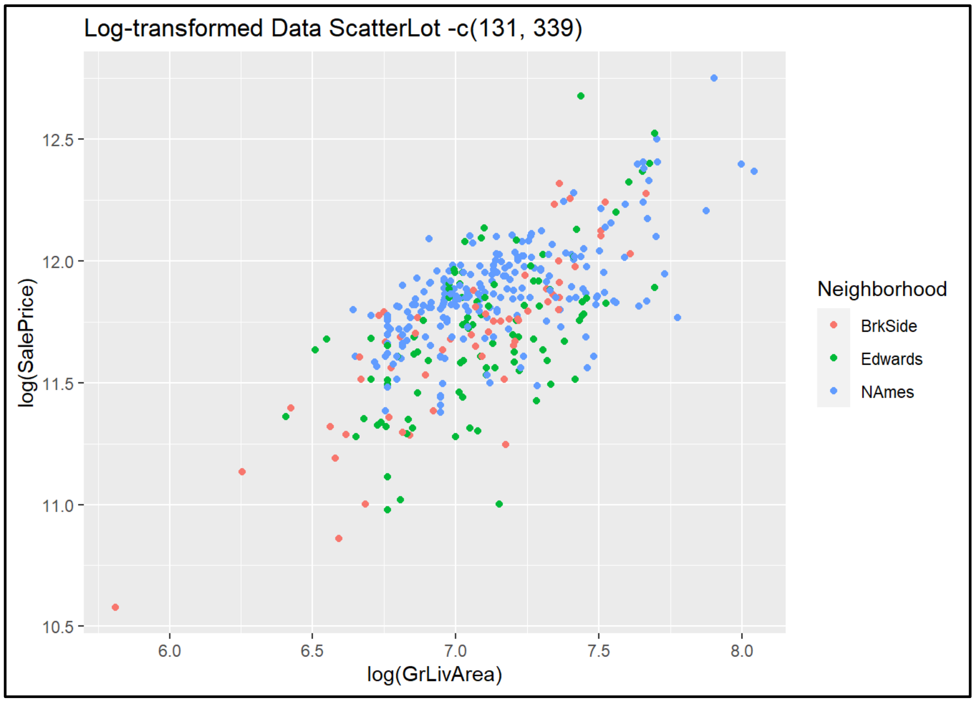


Figure 4

5.3 Fit Assessment of Model 2

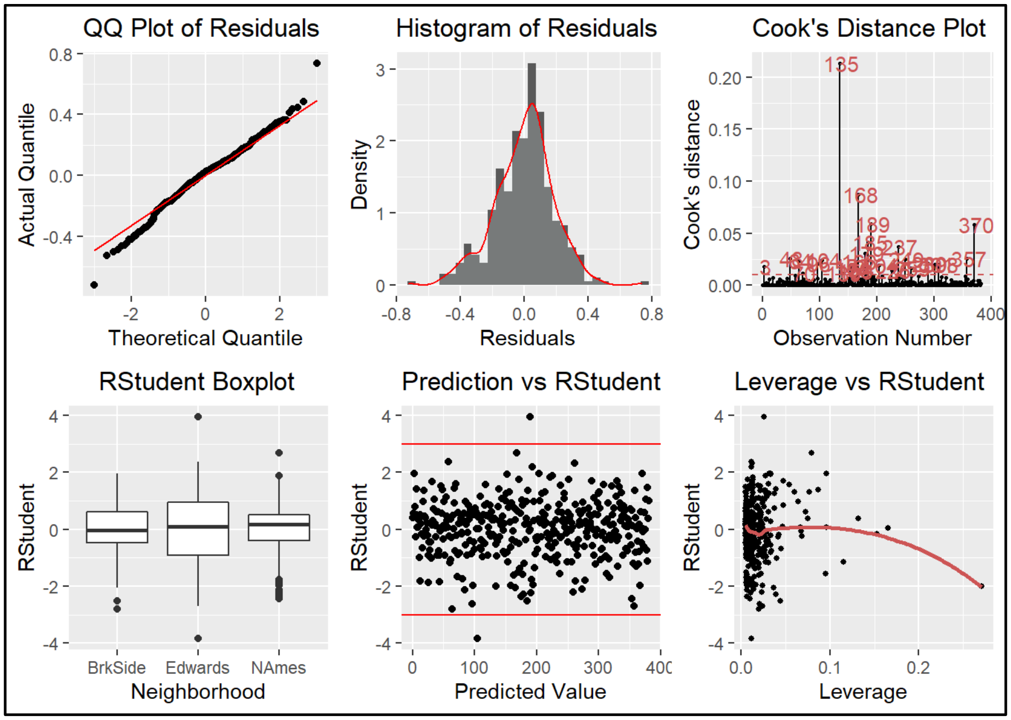


Figure 5

5.4 Comparing Competing Models

```
####Last defense between competing models####

#Extra sums of square test
anova(model1, model2)
```

```
## Analysis of Variance Table
##
## Model 1: log(SalePrice) ~ log(GrLivArea)
## Model 2: log(SalePrice) ~ log(GrLivArea) + as.factor(Neighborhood) + as.factor(Neighborhood) *
##     log(GrLivArea)
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    379 15.832
## 2    375 13.418  4    2.4137 16.863 1.004e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

	RMSE	R-Squared	CV PRESS
Model 1	0.2033879	0.4454874	16.02892
Model 2	0.189419	0.51558	13.94807

5.5 Analysis 1 Conclusion

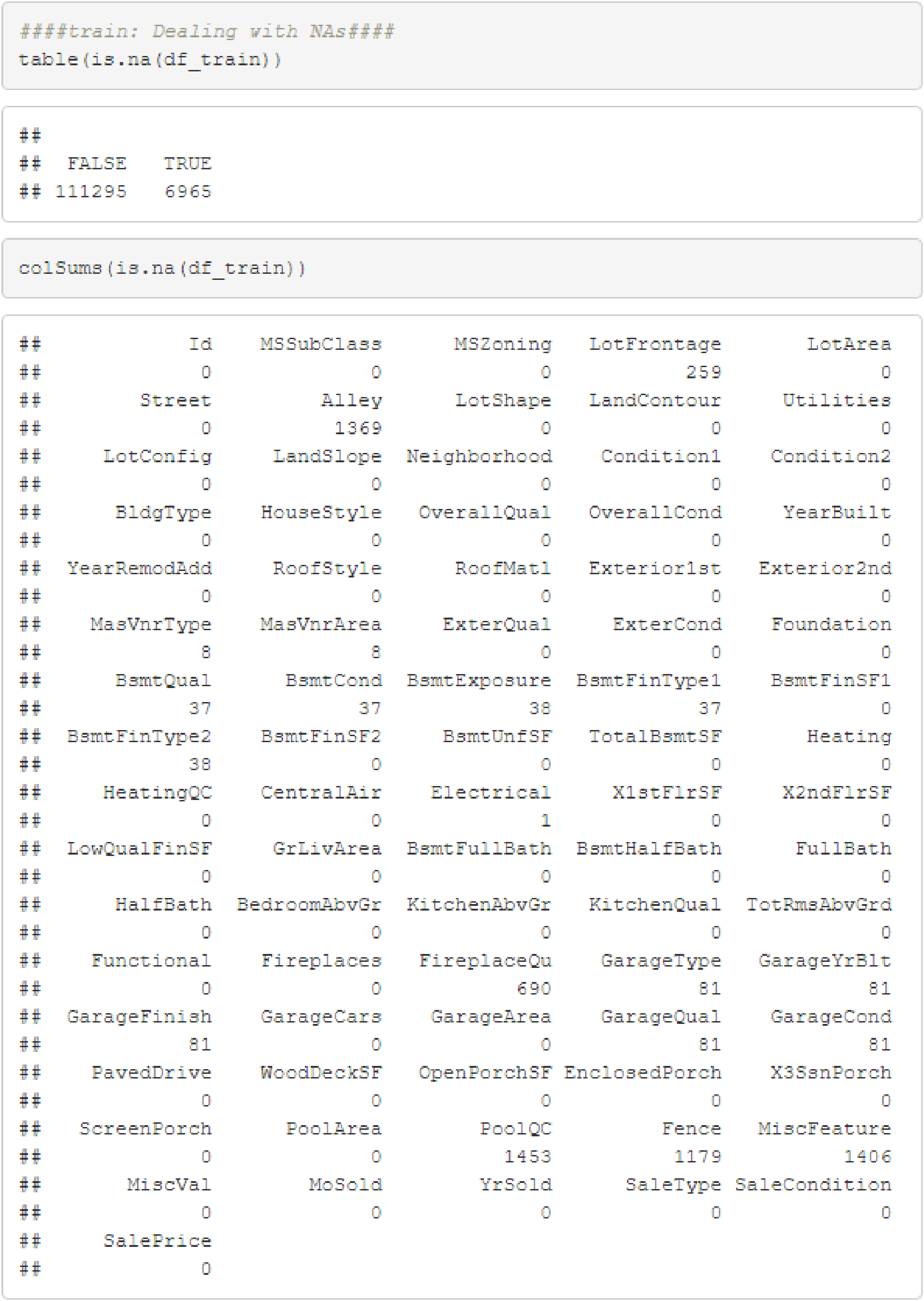
```
#Parameters
summary(model2)
```

```
##
## Call:
## lm(formula = log(SalePrice) ~ log(GrLivArea) + as.factor(Neighborhood) +
##     as.factor(Neighborhood) * GrLivArea, data = A1Data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.72239 -0.11122  0.02173  0.11025  0.73621
##
## Coefficients:
##                                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                                 8.983e+00  1.024e+00   8.775  < 2e-16
## log(GrLivArea)                              3.034e-01  1.712e-01   1.772  0.07715
## as.factor(Neighborhood)Edwards              1.823e-01  1.118e-01   1.631  0.10381
## as.factor(Neighborhood)NAmes                5.624e-01  1.047e-01   5.371 1.38e-07
## GrLivArea                                   4.670e-04  1.661e-04   2.810  0.00521
## as.factor(Neighborhood)Edwards:GrLivArea -1.620e-04  8.681e-05  -1.867  0.06273
## as.factor(Neighborhood)NAmes:GrLivArea   -3.469e-04  8.095e-05  -4.285 2.33e-05
##
## (Intercept)                              ***
## log(GrLivArea)                           .
## as.factor(Neighborhood)Edwards
## as.factor(Neighborhood)NAmes             ***
## GrLivArea                                **
## as.factor(Neighborhood)Edwards:GrLivArea .
## as.factor(Neighborhood)NAmes:GrLivArea   ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1888 on 374 degrees of freedom
## Multiple R-squared:  0.5311, Adjusted R-squared:  0.5236
## F-statistic:  70.6 on 6 and 374 DF,  p-value: < 2.2e-16
```

```
confint(model2)
```

```
##                                                 2.5 %        97.5 %
## (Intercept)                               6.9701432718  1.099594e+01
## log(GrLivArea)                           -0.0332081262  6.400836e-01
## as.factor(Neighborhood)Edwards           -0.0375382788  4.022294e-01
## as.factor(Neighborhood)NAmes              0.3564993451  7.683143e-01
## GrLivArea                                 0.0001402467  7.936767e-04
## as.factor(Neighborhood)Edwards:GrLivArea -0.0003327370  8.650323e-06
## as.factor(Neighborhood)NAmes:GrLivArea   -0.0005060273 -1.876761e-04
```

5.6 Dataset Missing Values



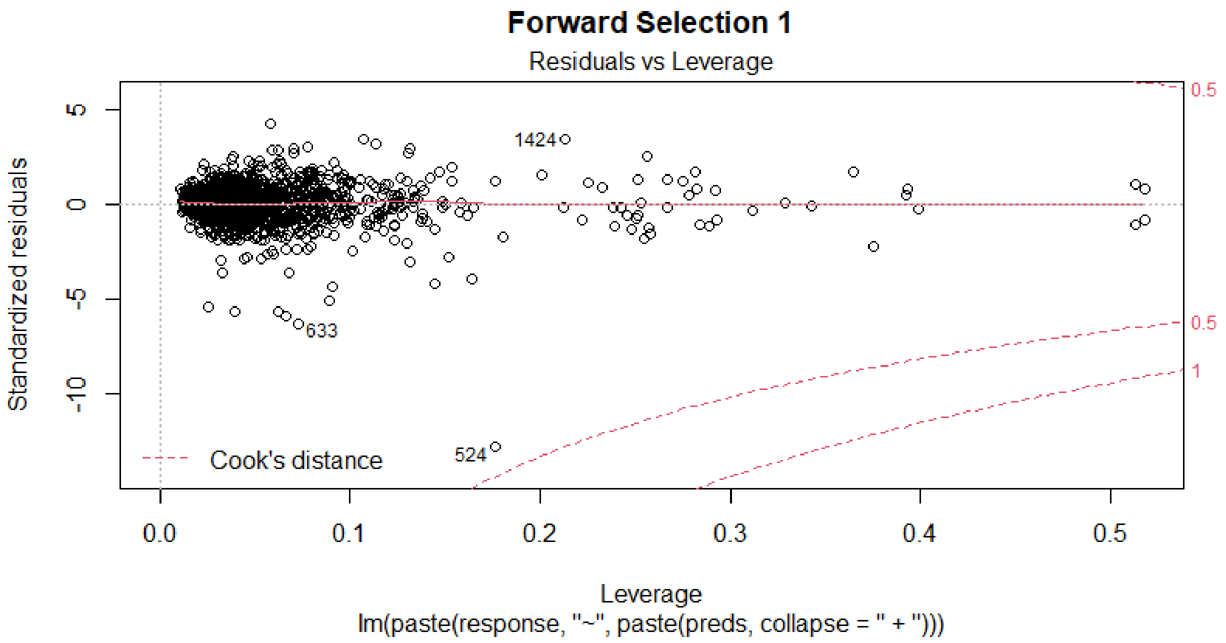
```
####train: Dealing with NAs####
table(is.na(df_train))
```

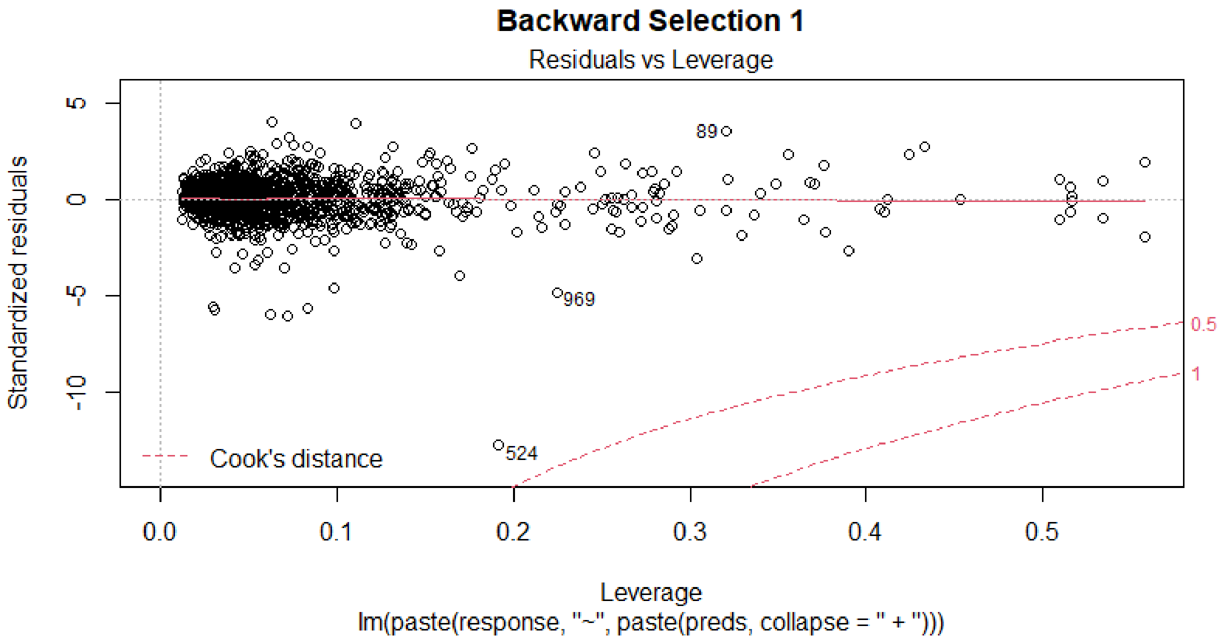
```
##
##  FALSE   TRUE
## 111295   6965
```

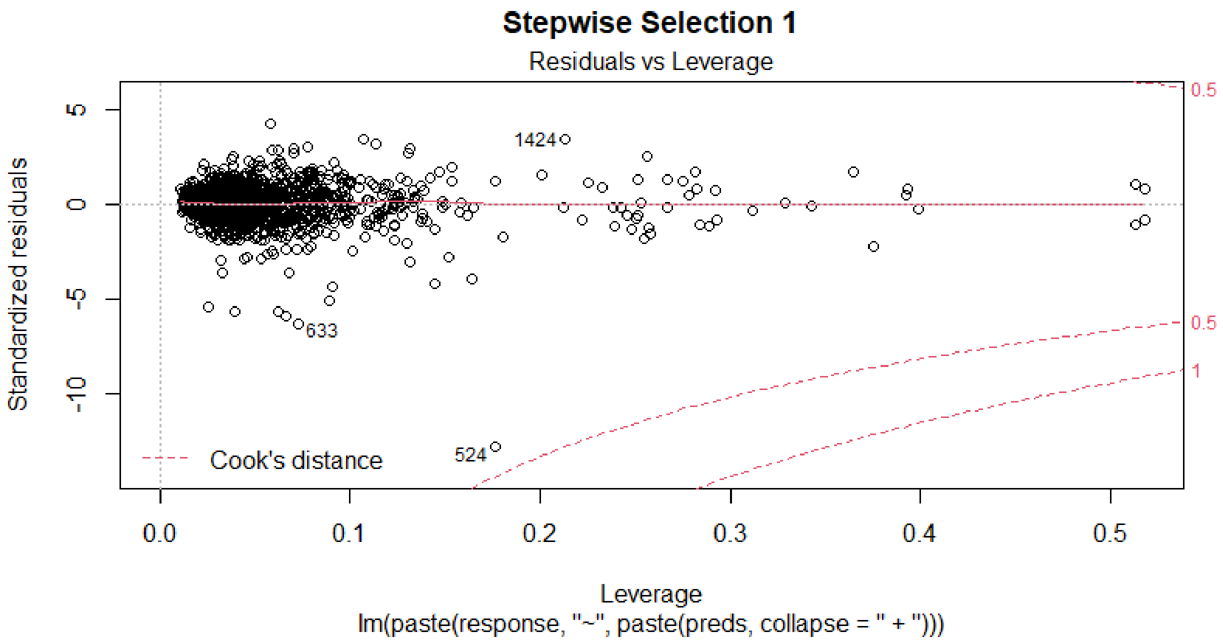
```
colSums(is.na(df_train))
```

```
##            Id    MSSubClass      MSZoning   LotFrontage       LotArea
##             0             0             0           259             0
##        Street         Alley      LotShape   LandContour     Utilities
##             0          1369             0             0             0
##     LotConfig     LandSlope  Neighborhood    Condition1    Condition2
##             0             0             0             0             0
##      BldgType    HouseStyle   OverallQual   OverallCond     YearBuilt
##             0             0             0             0             0
##  YearRemodAdd     RoofStyle      RoofMatl   Exterior1st   Exterior2nd
##             0             0             0             0             0
##    MasVnrType    MasVnrArea     ExterQual     ExterCond    Foundation
##             8             8             0             0             0
##      BsmtQual      BsmtCond  BsmtExposure  BsmtFinType1    BsmtFinSF1
##            37            37            38            37             0
##  BsmtFinType2    BsmtFinSF2     BsmtUnfSF   TotalBsmtSF       Heating
##            38             0             0             0             0
##     HeatingQC    CentralAir    Electrical     X1stFlrSF     X2ndFlrSF
##             0             0             1             0             0
##  LowQualFinSF     GrLivArea  BsmtFullBath  BsmtHalfBath      FullBath
##             0             0             0             0             0
##      HalfBath  BedroomAbvGr  KitchenAbvGr   KitchenQual  TotRmsAbvGrd
##             0             0             0             0             0
##    Functional    Fireplaces   FireplaceQu    GarageType   GarageYrBlt
##             0             0           690            81            81
##  GarageFinish    GarageCars    GarageArea    GarageQual    GarageCond
##            81             0             0            81            81
##    PavedDrive    WoodDeckSF   OpenPorchSF EnclosedPorch    X3SsnPorch
##             0             0             0             0             0
##   ScreenPorch      PoolArea        PoolQC         Fence   MiscFeature
##             0             0          1453          1179          1406
##       MiscVal        MoSold        YrSold      SaleType SaleCondition
##             0             0             0             0             0
##     SalePrice
##             0
```

5.7 Influential Observations

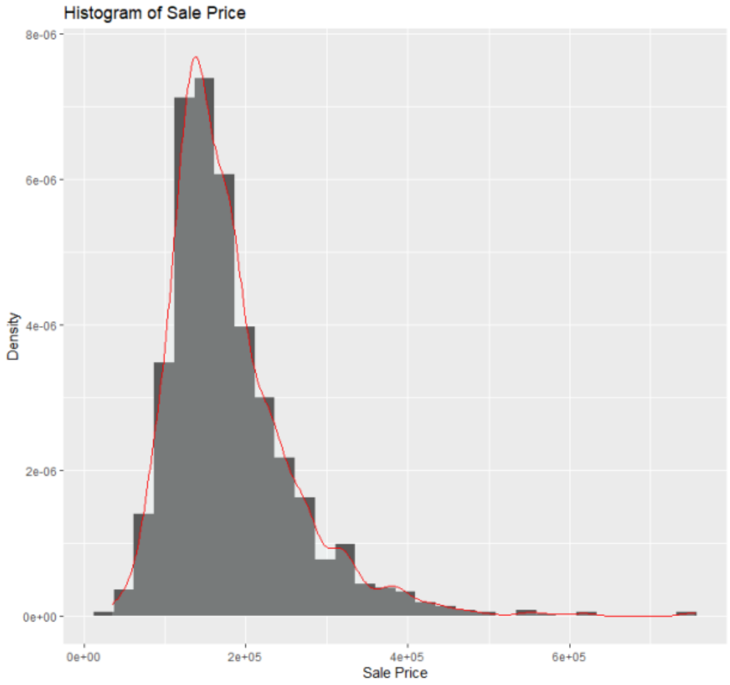

Forward Selection 1
Residuals vs Leverage
Standardized residuals
5
0
-5
-10
1424
633
524
Cook's distance
0.5
0.5
1
0.0
0.1
0.2
0.3
0.4
0.5
Leverage
lm(paste(response, "~", paste(preds, collapse = " + ")))


Backward Selection 1
Residuals vs Leverage
Standardized residuals
5
0
-5
-10
89
969
524
Cook's distance
0.5
1
0.0
0.1
0.2
0.3
0.4
0.5
Leverage
lm(paste(response, "~", paste(preds, collapse = " + ")))

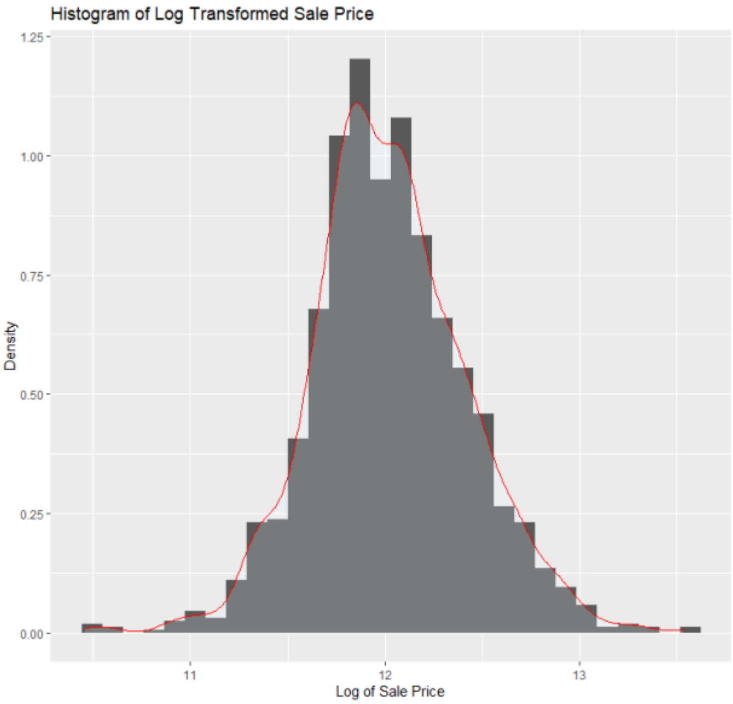


5.8 Before and After the Log Transformation

Before the log, we see the data has a right tail

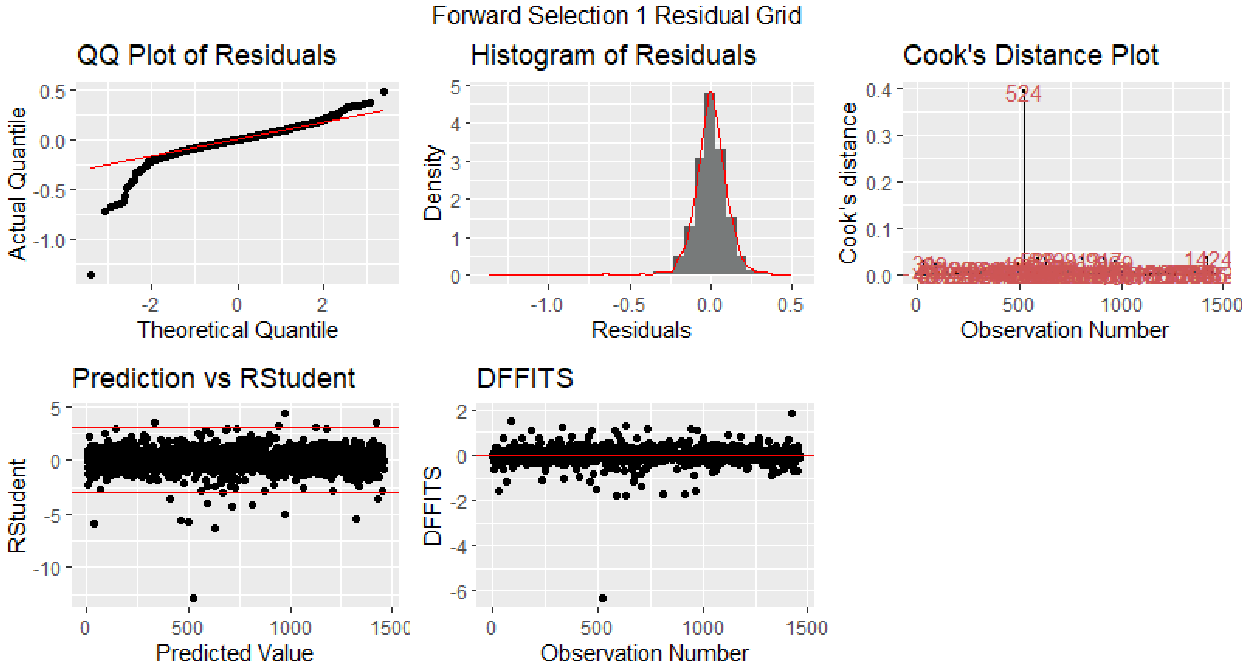


After the log, we see a much more uniform distribution

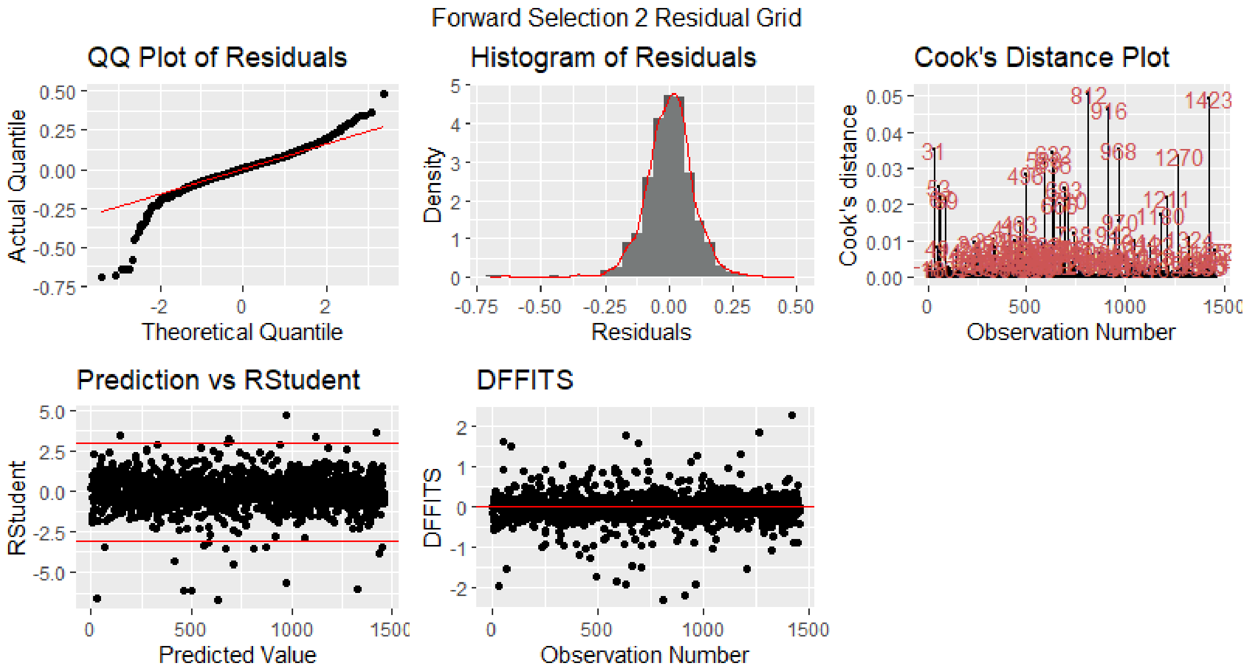


5.9 Forward, Backward and Stepwise Selection Residual Grid's

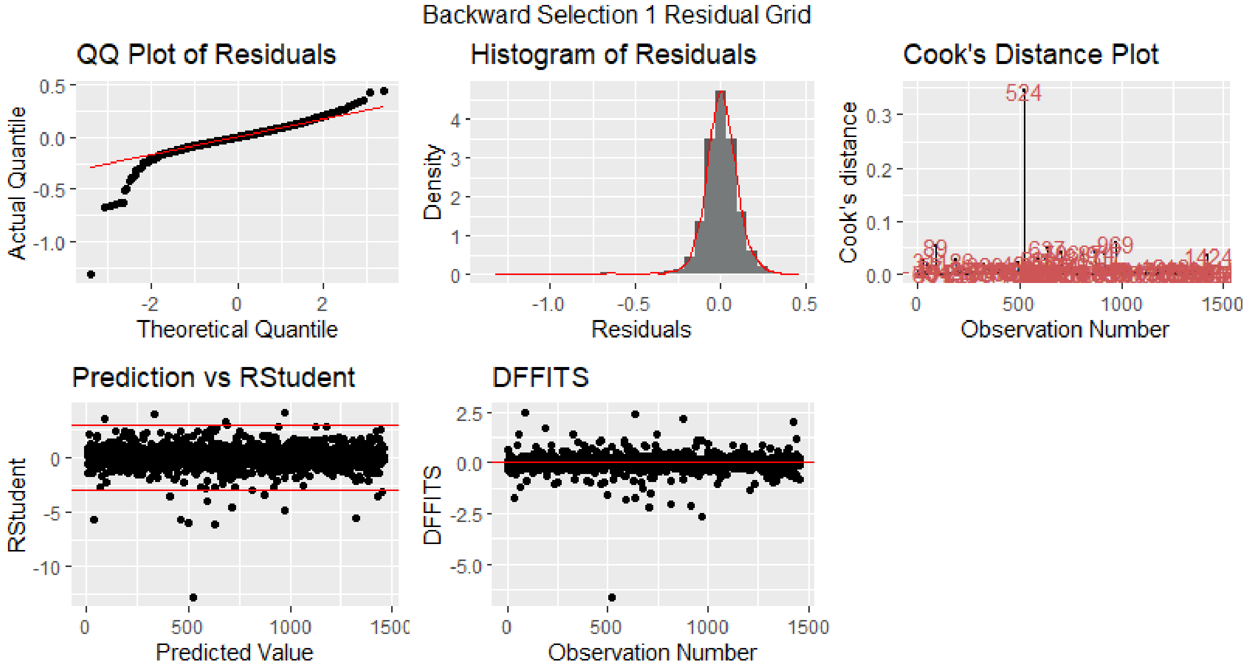
Before Transformation



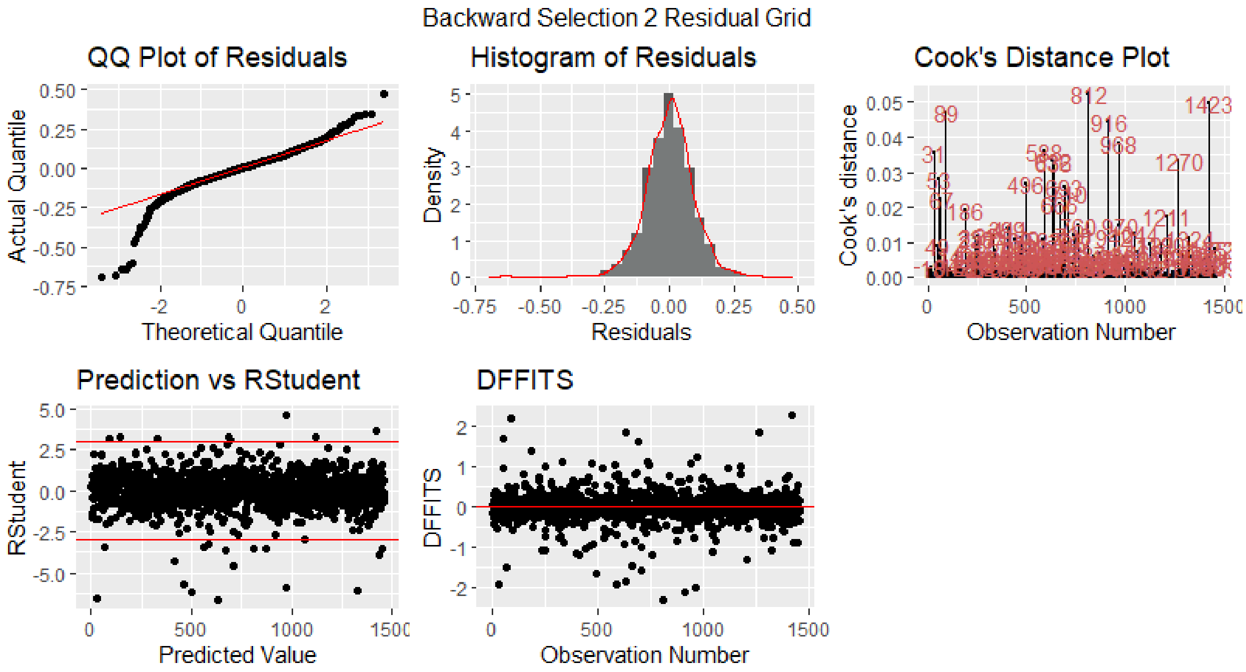
After Transformation



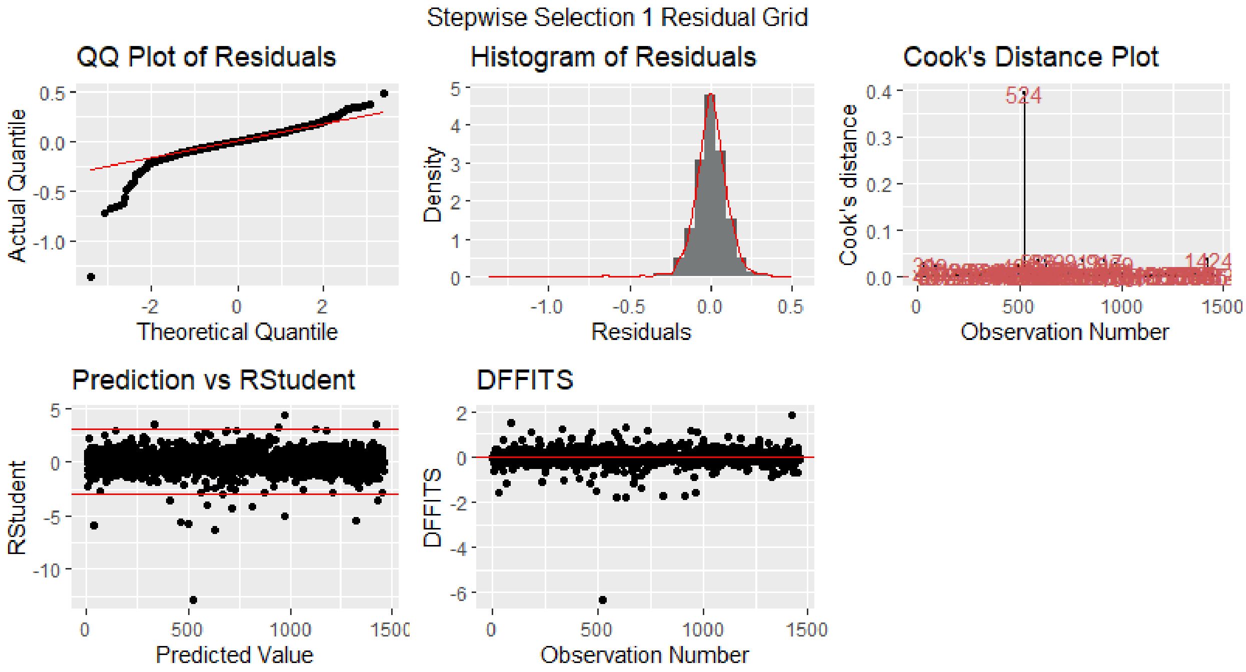
Before Transformation



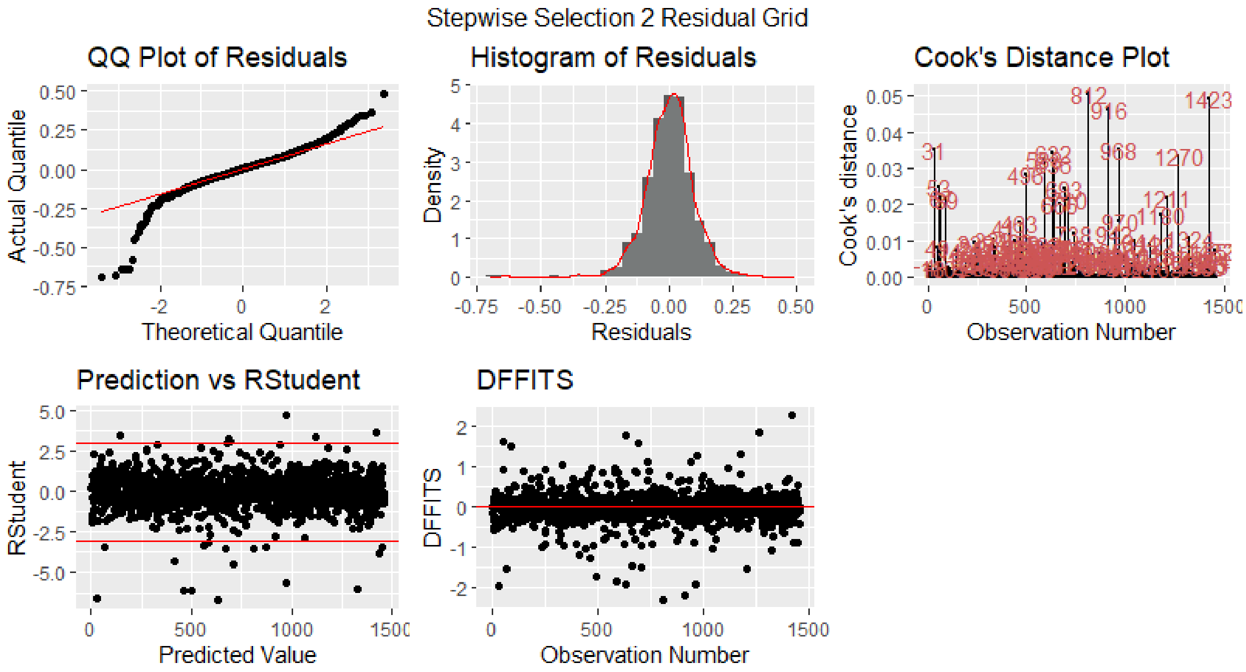
After Transformation



Before Transformation



After Transformation



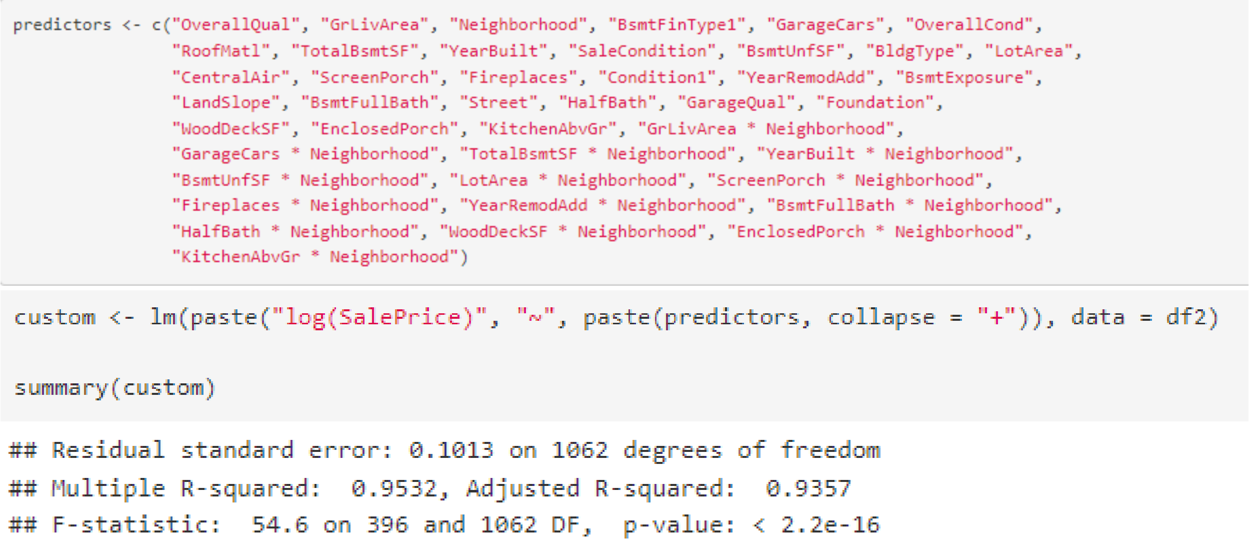
5.10 Forward & Stepwise Model

$$log(SalePrice) = \hat{\beta}_0 + \hat{\beta}_1 OverallQual + \hat{\beta}_2 GrLivArea + \hat{\beta}_3 Neighborhood + \hat{\beta}_4 BsmtFinType1 + \hat{\beta}_5 GarageCars + \hat{\beta}_6 OverallCond + \hat{\beta}_7 RoofMatl + \hat{\beta}_8 TotalBsmtSF + \hat{\beta}_9 YearBuilt + \hat{\beta}_{10} SaleCondition + \hat{\beta}_{11} BsmtUnfSF + \hat{\beta}_{12} BldgType + \hat{\beta}_{13} LotArea + \hat{\beta}_{14} CentralAir + \hat{\beta}_{15} ScreenPorch + \hat{\beta}_{16} Fireplaces + \hat{\beta}_{17} Condition1 + \hat{\beta}_{18} YearRemodAdd + \hat{\beta}_{19} BsmtExposure + \hat{\beta}_{20} LandSlope + \hat{\beta}_{21} BsmtFullBath + \hat{\beta}_{22} Street + \hat{\beta}_{23} HalfBath + \hat{\beta}_{24} GarageQual + \hat{\beta}_{25} Foundation + \hat{\beta}_{26} WoodDeckSF + \hat{\beta}_{27} EnclosedPorch + \hat{\beta}_{28} KitchenAbvGr$$

5.11 Backward Model

$$log(SalePrice) = \hat{\beta}_0 + \hat{\beta}_1 LotArea + \hat{\beta}_2 Street + \hat{\beta}_3 LandSlope + \hat{\beta}_4 Neighborhood + \hat{\beta}_5 Condition1 + \hat{\beta}_6 BldgType + \hat{\beta}_7 OverallQual + \hat{\beta}_8 OverallCond + \hat{\beta}_9 YearBuilt + \hat{\beta}_{10} YearRemodAdd + \hat{\beta}_{11} RoofMatl + \hat{\beta}_{12} Foundation + \hat{\beta}_{13} BsmtExposure + \hat{\beta}_{14} BsmtFinSF1 + \hat{\beta}_{15} BsmtFinSF2 + \hat{\beta}_{16} BsmtUnfSF + \hat{\beta}_{17} TotalBsmtSF + \hat{\beta}_{18} CentralAir + \hat{\beta}_{19} X1stFlrSF + \hat{\beta}_{20} X2ndFlrSF + \hat{\beta}_{21} LowQualFinSF + \hat{\beta}_{22} GrLivArea + \hat{\beta}_{23} HalfBath + \hat{\beta}_{24} KitchenAbvGr + \hat{\beta}_{25} Fireplaces + \hat{\beta}_{26} GarageCars + \hat{\beta}_{27} GarageQual + \hat{\beta}_{28} WoodDeckSF + \hat{\beta}_{28} ScreenPorch + \hat{\beta}_{28} SaleCondition$$

5.12 Custom Model



```
predictors <- c("OverallQual", "GrLivArea", "Neighborhood", "BsmtFinType1", "GarageCars", "OverallCond",
                "RoofMatl", "TotalBsmtSF", "YearBuilt", "SaleCondition", "BsmtUnfSF", "BldgType", "LotArea",
                "CentralAir", "ScreenPorch", "Fireplaces", "Condition1", "YearRemodAdd", "BsmtExposure",
                "LandSlope", "BsmtFullBath", "Street", "HalfBath", "GarageQual", "Foundation",
                "WoodDeckSF", "EnclosedPorch", "KitchenAbvGr", "GrLivArea * Neighborhood",
                "GarageCars * Neighborhood", "TotalBsmtSF * Neighborhood", "YearBuilt * Neighborhood",
                "BsmtUnfSF * Neighborhood", "LotArea * Neighborhood", "ScreenPorch * Neighborhood",
                "Fireplaces * Neighborhood", "YearRemodAdd * Neighborhood", "BsmtFullBath * Neighborhood",
                "HalfBath * Neighborhood", "WoodDeckSF * Neighborhood", "EnclosedPorch * Neighborhood",
                "KitchenAbvGr * Neighborhood")
```

```
custom <- lm(paste("log(SalePrice)", "~", paste(predictors, collapse = "+")), data = df2)

summary(custom)
```

```
## Residual standard error: 0.1013 on 1062 degrees of freedom
## Multiple R-squared:  0.9532, Adjusted R-squared:  0.9357
## F-statistic:  54.6 on 396 and 1062 DF,  p-value: < 2.2e-16
```

5.13 Analysis 1 Code

~~~
---
title: "MSDS 6371 Project Analysis 1"
author: "Duy Nguyen"
date: "4/3/2022"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

```{r, include=FALSE}
####Libaries####

library(tidyverse)
library(ggplot2)
library(caret)    #createDataPartition
library(DAAG)     #CVlm
library(car)      #leverage.plots
library(lindia)   #gg_cooksd
library(gridExtra) #grid.arrange
library(kableExtra)
library(olsrr)    #ols_step_forward_aic
```

```{r}
####Import Data####

getwd()
train <- read.csv("train.csv")
test <- read.csv("test.csv")
```

```{r}
####Analysis 1 Wrangle Data####
df <- dplyr::filter(train, Neighborhood =="Edwards" | Neighborhood =="NAmes" |
                    Neighborhood == "BrkSide")
df$Neighborhood <- as.factor(df$Neighborhood)
str(df)
```

```{r}
~~~

~~~
####Analysis 1 EDA####

#Normal Data Scatterplot
ggplot(df, aes(x = GrLivArea, y = SalePrice, color = Neighborhood)) +
  geom_point() +
  ggtitle("Normal Data ScatterLot")

#Log-transformed Data Scatterplot
ggplot(df, aes(x = log(GrLivArea), y = log(SalePrice), color = Neighborhood)) +
  geom_point() +
  ggtitle("Log-transformed Data ScatterLot")
```

```{r}
####Analysis 1 Modeling and Assumptions####
model0 <- lm(log(SalePrice) ~ log(GrLivArea), data = df)
paste(summary(model0)$r.squared, "  | ", summary(model0)$adj.r.squared)
#0.420370, 0.418849

library(lindia)
#Cook's Distance Plot
gg_cooksd(model0)

#Identify influential points
#From the above graph, those points are greater than 0.1 Cook's D
as.numeric(names(cooks.distance(model0))[(cooks.distance(model0) > 0.1)])
#And they are 131, 136, 339

#Start removing outliers one by one and a combo of outliers
#And find the most desired R-Squared and Adj R-Squared
A1Data01 = df[-131,]
model01 <- lm(log(SalePrice) ~ log(GrLivArea), data = A1Data01)
paste(summary(model01)$r.squared, "  | ", summary(model01)$adj.r.squared)

A1Data02 = df[-136,]
model02 <- lm(log(SalePrice) ~ log(GrLivArea), data = A1Data02)
paste(summary(model02)$r.squared, "  | ", summary(model02)$adj.r.squared)

A1Data03 = df[-339,]
model03 <- lm(log(SalePrice) ~ log(GrLivArea), data = A1Data03)
paste(summary(model03)$r.squared, "  | ", summary(model03)$adj.r.squared)

A1Data04 = df[-c(131, 136, 339),]
model04 <- lm(log(SalePrice) ~ log(GrLivArea), data = A1Data04)
~~~

~~~
paste(summary(model04)$r.squared, "  | ", summary(model04)$adj.r.squared)

A1Data05 = df[-c(131, 136),]
model05 <- lm(log(SalePrice) ~ log(GrLivArea), data = A1Data05)
paste(summary(model05)$r.squared, "  | ", summary(model05)$adj.r.squared)

A1Data06 = df[-c(131, 339),]
model06 <- lm(log(SalePrice) ~ log(GrLivArea), data = A1Data06)
paste(summary(model06)$r.squared, "  | ", summary(model06)$adj.r.squared)

#The most desired is A1Data06 and model06
A1Data = A1Data06
model1 = model06

str(A1Data)
str(model1)

#Log-transformed Data Scatterplot without indexes 131 and 339
ggplot(A1Data, aes(x = log(GrLivArea), y = log(SalePrice), color = Neighborhood)) +
  geom_point() +
  ggtitle("Log-transformed Data ScatterLot -c(131, 339)")
```


```{r}
####Model 2 Modeling####
model21 <- lm(log(SalePrice) ~ log(GrLivArea) + as.factor(Neighborhood), data = A1Data)
paste(summary(model21)$r.squared, "  | ", summary(model21)$adj.r.squared)
#"BrkSide" is read first so it was used as reference

model22 <- lm(log(SalePrice) ~ log(GrLivArea) + as.factor(Neighborhood) +
              as.factor(Neighborhood)*log(GrLivArea), data = A1Data)
paste(summary(model22)$r.squared, "  | ", summary(model22)$adj.r.squared)
press(model22)

model23 <- lm(log(SalePrice) ~ log(GrLivArea) * as.factor(Neighborhood), data = A1Data)
paste(summary(model23)$r.squared, "  | ", summary(model23)$adj.r.squared)
press(model23)

model24 <- lm(log(SalePrice) ~ log(GrLivArea) + as.factor(Neighborhood), data = A1Data)
paste(summary(model24)$r.squared, "  | ", summary(model24)$adj.r.squared)
press(model24)
summary(model24)

#The most desired is model23
~~~

~~~
model2 = model22
paste(summary(model2)$r.squared, "  | ", summary(model2)$adj.r.squared)
press(model2)
```

```{r}
####Model 2 EDA####

#Residuals QQ Plot
residuals = resid(model2)
p1 = ggplot(A1Data, aes(sample = residuals)) +
  geom_qq() +
  geom_qq_line(color = "red") +
  labs(title = "QQ Plot of Residuals", x = "Theoretical Quantile", y = "Actual Quantile")

#Residuals Histogram
p2 = ggplot(A1Data, aes(residuals)) +
  geom_histogram(aes(y = ..density..)) +
  geom_density(alpha = .2, color = "red", fill = "azure") +
  labs(title = "Histogram of Residuals", x = "Residuals", y = "Density")

#Cook's Distance Plot
library(lindia)
p3 = gg_cooksd(model2)

stdres2 <- rstandard(model2)
#Neighborhood vs RStudent
p4 = ggplot(A1Data, aes(as.factor(Neighborhood), stdres2)) +
  geom_boxplot() +
  labs(title = "    RStudent Boxplot", x = "Neighborhood", y = "RStudent")

#Standardized Residuals Plot
p5 = ggplot(A1Data, aes(x = seq(stdres2), y = stdres2)) +
  geom_point() +
  geom_hline(yintercept = 3, color = "red") +
  geom_hline(yintercept = -3, color = "red") +
  labs(title = "Prediction vs RStudent", x = "Predicted Value", y = "RStudent")

#Standardized Residuals vs Leverage
p6 = gg_resleverage(model2, method = "loess", se = FALSE, scale.factor = 1) +
  labs(title = "Leverage vs RStudent", x = "Leverage", y = "RStudent")

grid.arrange(p1, p2, p3, p4, p5, p6, ncol = 3)
```
~~~

~~~
```{r}
####Last defense between competing models####

#Extra sums of square test
anova(model1, model2)

#Repeated Cross-validation
set.seed(760397)
train.control <- trainControl(method = "repeatedcv", number = 10, repeats = 5)

#Train CV model1
CVModel1 <- train(log(SalePrice) ~ log(GrLivArea),
            data = A1Data, method = 'lm',
            trControl = train.control)
CVModel1$results$RMSE
CVModel1$results$Rsquared
press(model1)

#Train CV model1
CVModel2 <- train(log(SalePrice) ~ log(GrLivArea) + as.factor(Neighborhood) +
as.factor(Neighborhood)*log(GrLivArea),
            data = A1Data, method = 'lm',
            trControl = train.control)
CVModel2$results$RMSE
CVModel2$results$Rsquared
press(model2)

#Parameters
summary(model2)
confint(model2)
```
~~~

5.14 Analysis 2 Code

~~~
---
title: "forward selection"
author: "Leonardo Leal Filho"
date: "4/7/2022"
output: html_document
---
```{r TheLibraries, echo=FALSE, warning=FALSE, include=FALSE}
library(tidyverse)
library(ggplot2)
library(plotly)
library(GGally)
library(olsrr)
library(gridExtra)
library(lindia)
```



```{r TheDataSet, echo = FALSE}
# Reading the data
df <- read.csv("train.csv")

# This nested loop will take care of the NA values.
for(i in names(df)){
  for(j in 1:dim(df)[1]){
    if(is.na(df[j, i])){
      # Changing the NAs of the categorical features with the string "0"
      if(is.character(df[,i])){
        df[j,i] <- "0"
      }
      # Changing the NAs of the numerical features with the number 0
      else{
        df[j,i] <- 0
      }
    }
  }
}

# Making the categorical variables into factors
for(i in names(df)){
  if(is.character(df[, i])){
    df[, i] <- as.factor(df[, i])
  }
~~~

~~~
}

df <- subset(df, select = -c(Id, MSSubClass, MSZoning, Utilities, Condition2, Exterior1st,
              Exterior2nd, MasVnrType, KitchenQual, Functional, SaleType, PoolQC))
```

# Adjusting the Levels

```{r}
# Adjusting the Levels of condition2
df$OverallCond <- factor(df$OverallCond, levels = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10))

# Adjusting the Levels of BsmtQual
df$BsmtQual <- factor(df$BsmtQual, levels = c("Ex", "Gd", "TA", "Fa", "Po", "0"))


# Adjusting the Levels of BsmtCond
df$BsmtCond <- factor(df$BsmtCond, levels = c("Ex", "Gd", "TA", "Fa", "Po", "0"))


# Adjusting the Levels of PoolQC
df$PoolQC <- factor(df$PoolQC, levels = c("Ex", "Gd", "TA", "Fa", "0"))


# Making OverallCond into factor
df$OverallCond <- as.factor(df$OverallCond)

# Making OverallQual into factor
df$OverallQual <- as.factor(df$OverallQual)

```

```{r}


# Creating the full model
price_fit <- lm(log(SalePrice)~., data = df)



```
~~~

~~~
```{r ForwardSelectionModel1, echo = FALSE}

# Building the forward selection model
fwm <- ols_step_forward_p(price_fit, penter = 0.01, details = FALSE)


# Summary of the forward model
summary(fwm$model)
```



```{r ForwardSelectionModelGrid, echo = FALSE}
#Residuals QQ Plot
residuals = resid(fwm$model)
p1 = ggplot(df, aes(sample = residuals)) +
  geom_qq() +
  geom_qq_line(color = "red") +
  labs(title = "QQ Plot of Residuals", x = "Theoretical Quantile", y = "Actual Quantile")

#Residuals Histogram
p2 = ggplot(df, aes(residuals)) +
  geom_histogram(aes(y = ..density..), bins = 30) +
  geom_density(alpha = .2, color = "red", fill = "azure") +
  labs(title = "Histogram of Residuals", x = "Residuals", y = "Density")

#Cook's Distance Plot
p3 = gg_cooksd(fwm$model)


stdres2 <- rstandard(fwm$model)
#Standardized Residuals Plot
p5 = ggplot(df, aes(x = seq(stdres2), y = stdres2)) +
  geom_point() +
  geom_hline(yintercept = 3, color = "red") +
  geom_hline(yintercept = -3, color = "red") +
  labs(title = "Prediction vs RStudent", x = "Predicted Value", y = "RStudent")

#DFFITS
p6 = ggplot(df, aes(x = seq(dffits(fwm$model)), y = dffits(fwm$model))) +
  geom_point() +
  geom_hline(color="red", yintercept=0) +
  labs(title = "DFFITS", x = "Observation Number", y = "DFFITS")
  ylim(-5,5)
~~~

~~~
grid.arrange(p1, p2, p3, p5, p6, ncol = 3, top=("Forward Selection 1 Residual Grid"))

#Standardized Residuals vs Leverage
plot(fwm$model, which = 5, main = ("Forward Selection 1"))
```





```{r BackwardSelectionModel, echo=FALSE}

# Building the backward selection model
bwm <- ols_step_backward_p(price_fit, prem = 0.01, details = FALSE)

# Summary of the backward model
summary(bwm$model)
```





```{r BackwardSelectionModelGrid1, echo = F}
#Residuals QQ Plot
residuals = resid(bwm$model)
p1 = ggplot(df, aes(sample = residuals)) +
  geom_qq() +
  geom_qq_line(color = "red") +
  labs(title = "QQ Plot of Residuals", x = "Theoretical Quantile", y = "Actual Quantile")

#Residuals Histogram
p2 = ggplot(df, aes(residuals)) +
  geom_histogram(aes(y = ..density..), bins = 30) +
  geom_density(alpha = .2, color = "red", fill = "azure") +
  labs(title = "Histogram of Residuals", x = "Residuals", y = "Density")

#Cook's Distance Plot
p3 = gg_cooksd(bwm$model)


stdres2 <- rstandard(bwm$model)
#Standardized Residuals Plot
p5 = ggplot(df, aes(x = seq(stdres2), y = stdres2)) +
  geom_point() +
  geom_hline(yintercept = 3, color = "red") +
  geom_hline(yintercept = -3, color = "red") +
~~~

~~~
  labs(title = "Prediction vs RStudent", x = "Predicted Value", y = "RStudent")

#DFFITS
p6 = ggplot(df, aes(x = seq(dffits(bwm$model)), y = dffits(bwm$model))) +
  geom_point() +
  geom_hline(color="red", yintercept=0) +
  labs(title = "DFFITS", x = "Observation Number", y = "DFFITS")
  ylim(-5,5)

grid.arrange(p1, p2, p3, p5, p6, ncol = 3, top=("Backward Selection 1 Residual Grid"))

#Standardized Residuals vs Leverage
plot(bwm$model, which = 5, main = ("Backward Selection 1"))
```




```{r StepwiseSelectionModel1, echo=FALSE}


# Building the stepwise selection model
swm <- ols_step_both_p(price_fit, pent = 0.01, prem = 0.2, details = FALSE)




# Summary of the stepwise model
summary(swm$model)
```


```{r StepwiseSelectionModelGrid1, echo = F}
#Residuals QQ Plot
residuals = resid(swm$model)
p1 = ggplot(df, aes(sample = residuals)) +
  geom_qq() +
  geom_qq_line(color = "red") +
  labs(title = "QQ Plot of Residuals", x = "Theoretical Quantile", y = "Actual Quantile")

#Residuals Histogram
p2 = ggplot(df, aes(residuals)) +
  geom_histogram(aes(y = ..density..), bins = 30) +
  geom_density(alpha = .2, color = "red", fill = "azure") +
  labs(title = "Histogram of Residuals", x = "Residuals", y = "Density")

#Cook's Distance Plot
~~~

~~~
p3 = gg_cooksd(swm$model)


stdres2 <- rstandard(swm$model)
#Standardized Residuals Plot
p5 = ggplot(df, aes(x = seq(stdres2), y = stdres2)) +
  geom_point() +
  geom_hline(yintercept = 3, color = "red") +
  geom_hline(yintercept = -3, color = "red") +
  labs(title = "Prediction vs RStudent", x = "Predicted Value", y = "RStudent")

#DFFITS
p6 = ggplot(df, aes(x = seq(dffits(swm$model)), y = dffits(swm$model))) +
  geom_point() +
  geom_hline(color="red", yintercept=0) +
  labs(title = "DFFITS", x = "Observation Number", y = "DFFITS")
  ylim(-5,5)

grid.arrange(p1, p2, p3, p5, p6, ncol = 3, top=("Stepwise Selection 1 Residual Grid"))

#Standardized Residuals vs Leverage
plot(swm$model, which = 5, main = ("Stepwise Selection 1"))
```
~~~

## Initial Thoughts

When analyzing the models we can see that there are outliers that have a high cooks D and high leverage for all three of the models. Because of that we will remove the two datapoints (524 and 826) and redo all three model selections.

~~~
```{r}


# Removing the outliers
df2 <- df[-c(524),]

# The new full model
price_fit2 <- lm(log(SalePrice)~., data = df2)
~~~

~~~
```




```{r ForwardSelectionModel2, echo = FALSE}


# Building the forward selection model
fwm2 <- ols_step_forward_p(price_fit2, penter = 0.01, details = FALSE)

# Summary of the forward model
summary(fwm2$model)

# The features used for the fwm2 model
fwm2Predictors <- fwm2$predictors
```




```{r ForwardSelectionModelGrid2, echo = FALSE}
#Residuals QQ Plot
residuals = resid(fwm2$model)
p1 = ggplot(df2, aes(sample = residuals)) +
  geom_qq() +
  geom_qq_line(color = "red") +
  labs(title = "QQ Plot of Residuals", x = "Theoretical Quantile", y = "Actual Quantile")

#Residuals Histogram
p2 = ggplot(df2, aes(residuals)) +
  geom_histogram(aes(y = ..density..), bins = 30) +
  geom_density(alpha = .2, color = "red", fill = "azure") +
  labs(title = "Histogram of Residuals", x = "Residuals", y = "Density")

#Cook's Distance Plot
p3 = gg_cooksd(fwm2$model)

stdres2 <- rstandard(fwm2$model)
#Standardized Residuals Plot
p5 = ggplot(df2, aes(x = seq(stdres2), y = stdres2)) +
  geom_point() +
  geom_hline(yintercept = 3, color = "red") +
~~~

```
  geom_hline(yintercept = -3, color = "red") +
  labs(title = "Prediction vs RStudent", x = "Predicted Value", y = "RStudent")

#DFFITS
p6 = ggplot(df2, aes(x = seq(dffits(fwm2$model)), y = dffits(fwm2$model))) +
  geom_point() +
  geom_hline(color="red", yintercept=0) +
  labs(title = "DFFITS", x = "Observation Number", y = "DFFITS")
  ylim(-5,5)

grid.arrange(p1, p2, p3, p5, p6, ncol = 3, top=("Forward Selection 2 Residual Grid"))

#Standardized Residuals vs Leverage
plot(fwm2$model, which = 5, main = "Forward Selection 2")
```

~~~
```{r BackwardSelectionModel2, echo=FALSE}


# Building the backward selection model
bwm2 <- ols_step_backward_p(price_fit2, prem = 0.01, details = FALSE)



# Summary of the backward model
summary(bwm2$model)

# Because the backward selection gives the removed variables, to collect the variables used in
the
# model I need to first create an empty vector
bwm2Predictors <- c()

# Now I add all of the features from the df2 dataframe not included in the removed list from the
# bwm2 selection model
for(i in names(df2)){
  if(!i %in% bwm2$removed){
    bwm2Predictors <- c(bwm2Predictors, i) # filling the vector with column names that was not
removed
  }
}
```
~~~

~~~
```{r BackwardSelectionModelGrid2, echo = F}
#Residuals QQ Plot
residuals = resid(bwm2$model)
p1 = ggplot(df2, aes(sample = residuals)) +
  geom_qq() +
  geom_qq_line(color = "red") +
  labs(title = "QQ Plot of Residuals", x = "Theoretical Quantile", y = "Actual Quantile")

#Residuals Histogram
p2 = ggplot(df2, aes(residuals)) +
  geom_histogram(aes(y = ..density..), bins = 30) +
  geom_density(alpha = .2, color = "red", fill = "azure") +
  labs(title = "Histogram of Residuals", x = "Residuals", y = "Density")

#Cook's Distance Plot
p3 = gg_cooksd(bwm2$model)


stdres2 <- rstandard(bwm2$model)
#Standardized Residuals Plot
p5 = ggplot(df2, aes(x = seq(stdres2), y = stdres2)) +
  geom_point() +
  geom_hline(yintercept = 3, color = "red") +
  geom_hline(yintercept = -3, color = "red") +
  labs(title = "Prediction vs RStudent", x = "Predicted Value", y = "RStudent")

#DFFITS
p6 = ggplot(df2, aes(x = seq(dffits(bwm2$model)), y = dffits(bwm2$model))) +
  geom_point() +
  geom_hline(color="red", yintercept=0) +
  labs(title = "DFFITS", x = "Observation Number", y = "DFFITS")
  ylim(-5,5)

grid.arrange(p1, p2, p3, p5, p6, ncol = 3, top=("Backward Selection 2 Residual Grid"))

#Standardized Residuals vs Leverage
plot(bwm2$model, which = 5, main = ("Backward Selection 2"))
```
~~~

~~~
```{r StepwiseSelectionModel2, echo=FALSE}


# Building the stepwise selection model
swm2 <- ols_step_both_p(price_fit2, pent = 0.01, prem = 0.2, details = FALSE)



# Summary of the stepwise model
summary(swm2$model)

```



```{r StepwiseSelectionModelGrid2, echo = F}
#Residuals QQ Plot
residuals = resid(swm2$model)
p1 = ggplot(df2, aes(sample = residuals)) +
  geom_qq() +
  geom_qq_line(color = "red") +
  labs(title = "QQ Plot of Residuals", x = "Theoretical Quantile", y = "Actual Quantile")

#Residuals Histogram
p2 = ggplot(df2, aes(residuals)) +
  geom_histogram(aes(y = ..density..), bins = 30) +
  geom_density(alpha = .2, color = "red", fill = "azure") +
  labs(title = "Histogram of Residuals", x = "Residuals", y = "Density")

#Cook's Distance Plot
p3 = gg_cooksd(swm2$model)

stdres2 <- rstandard(swm2$model)
#Standardized Residuals Plot
p5 = ggplot(df2, aes(x = seq(stdres2), y = stdres2)) +
  geom_point() +
  geom_hline(yintercept = 3, color = "red") +
  geom_hline(yintercept = -3, color = "red") +
  labs(title = "Prediction vs RStudent", x = "Predicted Value", y = "RStudent")

#DFFITS
p6 = ggplot(df2, aes(x = seq(dffits(swm2$model)), y = dffits(swm2$model))) +
  geom_point() +
  geom_hline(color="red", yintercept=0) +
~~~

~~~
  labs(title = "DFFITS", x = "Observation Number", y = "DFFITS")
  ylim(-5,5)

grid.arrange(p1, p2, p3, p5, p6, ncol = 3, top=("Stepwise Selection 2 Residual Grid"))

#Standardized Residuals vs Leverage
plot(swm2$model, which = 5, main = ("Stepwise Selection 2"))
```



## Now none of the models seem to have datapoints with a high leverage factor.  Cook's D also
show that the datapoints that seem high, are actually not that high having a cook's d score of a
little above 0.1.
~~~

After deleting the outliers, the forward and stepwise selection have essentially the same model, consequently the exactly same statistics. The Backward model have a higher R-square so thus far it seems to be the best model.

~~~
```{r}
# The swm2 predictors
swm2Predictors <- swm2$predictors

# The swm2 dataframe
dfSwm <- data.frame(lSalePrice = log(df2$SalePrice))
dfSwm[, swm2Predictors] <- df2[, swm2Predictors]

swmFit <-
lm(lSalePrice~OverallQual+GrLivArea+Neighborhood+BsmtFinType1+GarageCars+OverallCon
d+RoofMatl+

TotalBsmtSF+YearBuilt+SaleCondition+BsmtUnfSF+MSZoning+MSSubClass+LotArea+Functio
nal+

CentralAir+ScreenPorch+Condition1+YearRemodAdd+Fireplaces+LandSlope+Exterior1st+Bsm
tExposure+
             Heating+GarageQual+KitchenQual+WoodDeckSF+OpenPorchSF+EnclosedPorch,
data = dfSwm)
ols_press(swmFit)

PRESS <- function(linear.model) {
  #' calculate the predictive residuals
  pr <- round(residuals(linear.model)/(1-lm.influence(linear.model)$hat), 5)
~~~

```
  #' calculate the PRESS
  PRESS <- sum(pr^2)

  return(PRESS)
}

PRESS(swmFit)
```
```

```
```{r CreatingNewModel, echo = FALSE}
# Selecting the predictors of the stepwise model selection
predictors <- swm2$predictors

# Adding the variables that will adjust the coefficient of each numerical variable with the
# Neighborhood factor
for(i in predictors){
  if(is.numeric(df2[, i])){
    predictors <- c(predictors, paste(i, "*", "Neighborhood")) # Adding the Adjustment variables
  }
}

# Creating the regression model
model12 <- lm(paste("log(SalePrice)", "~",paste(predictors, collapse =  "+")), data = df2)

# Summarizing the model
summary(model12)
```

```{r PlottingNewModelResiduals}
#Residuals QQ Plot
residuals = resid(model12)
p1 = ggplot(df2, aes(sample = residuals)) +
  geom_qq() +
  geom_qq_line(color = "red") +
  labs(title = "QQ Plot of Residuals", x = "Theoretical Quantile", y = "Actual Quantile")

#Residuals Histogram
p2 = ggplot(df2, aes(residuals)) +
  geom_histogram(aes(y = ..density..), bins = 30) +
  geom_density(alpha = .2, color = "red", fill = "azure") +
  labs(title = "Histogram of Residuals", x = "Residuals", y = "Density")

#Cook's Distance Plot
p3 = gg_cooksd(model12)
```

~~~
stdres2 <- rstandard(model12)
#Standardized Residuals Plot
p5 = ggplot(df2, aes(x = seq(stdres2), y = stdres2)) +
  geom_point() +
  geom_hline(yintercept = 3, color = "red") +
  geom_hline(yintercept = -3, color = "red") +
  labs(title = "Prediction vs RStudent", x = "Predicted Value", y = "RStudent")

#DFFITS
p6 = ggplot(df2, aes(x = seq(dffits(model12)), y = dffits(model12))) +
  geom_point() +
  geom_hline(color="red", yintercept=0) +
  labs(title = "DFFITS", x = "Observation Number", y = "DFFITS")
  ylim(-5,5)

grid.arrange(p1, p2, p3, p5, p6, ncol = 3, top=("New Model Residual Grid"))

#Standardized Residuals vs Leverage
plot(model12, which = 5, main = ("New Model"))
```
~~~

Importing the test set and making the predictions.

~~~
```{r}
#loading the test data
test <- read.csv("test.csv")

#Selecting the predictors for the test data
test2 <- test[, swm2$predictors]
```
~~~

~~~
```{r}
# Checking which of the predictors have NA values
l <- c()
for(i in names(test2)){
  if(sum(is.na(test2[, i]))){
    l <- c(l, i)
  }
}
```
~~~

~~~
```{r}
~~~

~~~~
# Taking care of the predictors NA values
for(i in l){
  for(j in 1:dim(test2)[1]){
    if(is.character(test2[,i])){
      test2[j, i] <- "0"
    }
    else{
      test2[j, i] <- 0
    }
  }
}

```


```{r}

# Transforming the character variables into factors
for(i in names(test2)){
  if(is.character(test2[, i])){
    test2[, i] <- as.factor(test2[, i])
  }
}
```


```{r}

# Making OverallCond into factor
test2$OverallCond <- as.factor(test2$OverallCond)

# Making OverallQual into factor
test2$OverallQual <- as.factor(test2$OverallQual)

```



```{r ForwardPred}

# Forward Prediction
pred <- data.frame(Id = seq(1461, 2919), SalePrice = round(exp(predict(fwm2$model, test2)),
2))

write.csv(pred, "kaggle_forward.csv", row.names = FALSE)
```
~~~~

~~~
```{r StepWisePred}

# Stepward Prediction
pred <- data.frame(Id = seq(1461, 2919), SalePrice = round(exp(predict(swm2$model, test2)),
2))

write.csv(pred, "kaggle_stepwise.csv", row.names = FALSE)
```


```{r CustomPred}

# Custom Prediction
pred <- data.frame(Id = seq(1461, 2919), SalePrice = round(exp(predict(model12, test2)), 2))

write.csv(pred, "kaggle_custom.csv", row.names = FALSE)
```



```{r}
# Adapting testing dataset for the backward prediction
test3 <- test[, predictors1]

```



```{r}
# Checking which predictors have NA values
l <- c()

for(i in names(test3)){
  if(sum(is.na(test3[, i]))){
    l <- c(l, i)
  }
}
```



```{r}
# Taking care of the predictors NA values
for(i in l){
  for(j in 1:dim(test3)[1]){
    if(is.character(test3[,i])){
      test3[j, i] <- "0"
~~~

~~~
    }
    else{
      test3[j, i] <- 0
    }
  }
}
```



```{r}
# Transforming the necessary variables into factors
test3$OverallQual <- as.factor(test3$OverallQual)
test3$OverallCond <- as.factor(test3$OverallCond)
```


```{r}
# Backward Prediction
pred <- data.frame(Id = seq(1461, 2919), SalePrice = round(exp(predict(bwm2$model, test3)), 2))

write.csv(pred, "kaggle_backward.csv", row.names = FALSE)
```
~~~

References

[1] Cock, D. D. (2011). Ames, Iowa: Alternative to the Boston Housing Data as an End of Semester Regression Project. *Journal of Statistics Education*, from http://jse.amstat.org/v19n3/decock.pdf

[2] Kaggle (2016). *House prices - advanced regression techniques*. Data retrieved April 5, 2022, from https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques/data