Predicting Residential House Price in Ames,Iowa

Dan Derieg(dderi2@illinois.edu)

08/07/2020

Introduction

The purpose of this project is to predict the residential house price in Ames, Iowa. To do this we create a linear model which can predict house prices in Iowa based on different predictor variables. This approach is similar to how companies like Redfin and Zillow calculate their price estimate for their websites

Our final project will try to touch some of the concepts mentioned below:

- Multiple linear regression
- Dummy variables
- Interaction
- Residual diagnostics
- Outlier diagnostics
- Transformations
- Polynomial regression
- Model selection

Dataset Source

The dataset is provided by Dean De Cock from Truman State University and initiative was inspired by Kaggle problem https://www.kaggle.com/c/house-prices-advanced-regression-techniques/overview.

The data consists of 2930 observations and 82 variables. It includes of a mix of nominal, discrete, ordinal and continuous variables.

The raw dataset is rather large consisting of 82 potential features. As such we make a subset of the data consisting of the 25 predictors we consider the most relevant to our study. The dataset is further refined to only include residential properties.

Loding Data

We loaded `housing_data` into R.

Below are a few of the variables we considered to be important for our regression analysis.

- `MS Zoning`: Identifies the general zoning classification of the sale.
- `Lot Area`: Lot size in square feet.
- `Overall Qual`: Overall material and finish quality.
- `Total Bsmt SF`: Total square feet of basement area.
- `Year Built`: Original construction date.
- `Bedroom AbvGr`: Bedrooms above grade (does NOT include basement bedrooms).
- `Bsmt Full Bath`: Full bathrooms above grade.
- `GarageCars`: Size of garage in car capacity.
- `Street`: Type of road access to property

```r
#Install all packages
library(readr)
library(stringr)
library(ggplot2)
library(dplyr)
library(corrplot)
library(tidyverse)
library("VIM")
library(car)
library(lmtest)
library(caret)
library(rpart)
library(rpart.plot)
library(PerformanceAnalytics)
library(trafo)
#install.packages("stringr")
#install.packages("corrplot")
#install.packages("tidyverse")
#install.packages("car")
```

Data Sanity Checking

```r
housing_data = read_csv("housing_data.csv")
names(housing_data) = str_replace_all(names(housing_data), c(" " = "." , "," = "" ))
```

```r
housing_data[c(1:10),c(1:5)]
```

```
## # A tibble: 10 x 5
##    Order       PID MS.SubClass MS.Zoning Lot.Frontage
##    <dbl>     <dbl>       <dbl> <chr>            <dbl>
##  1     1 526301100          20 RL                 141
##  2     2 526350040          20 RH                  80
##  3     3 526351010          20 RL                  81
##  4     4 526353030          20 RL                  93
##  5     5 527105010          60 RL                  74
##  6     6 527105030          60 RL                  78
##  7     7 527127150         120 RL                  41
##  8     8 527145080         120 RL                  43
##  9     9 527146030         120 RL                  39
## 10    10 527162130          60 RL                  60
```

Methods

To begin the analysis, data cleaning techniques were applied. On the basis of this some predictors were selected and a subset created as opposed to using all of the variables. The data is also examined for variables which have too many 'NA's, which could impact the results.

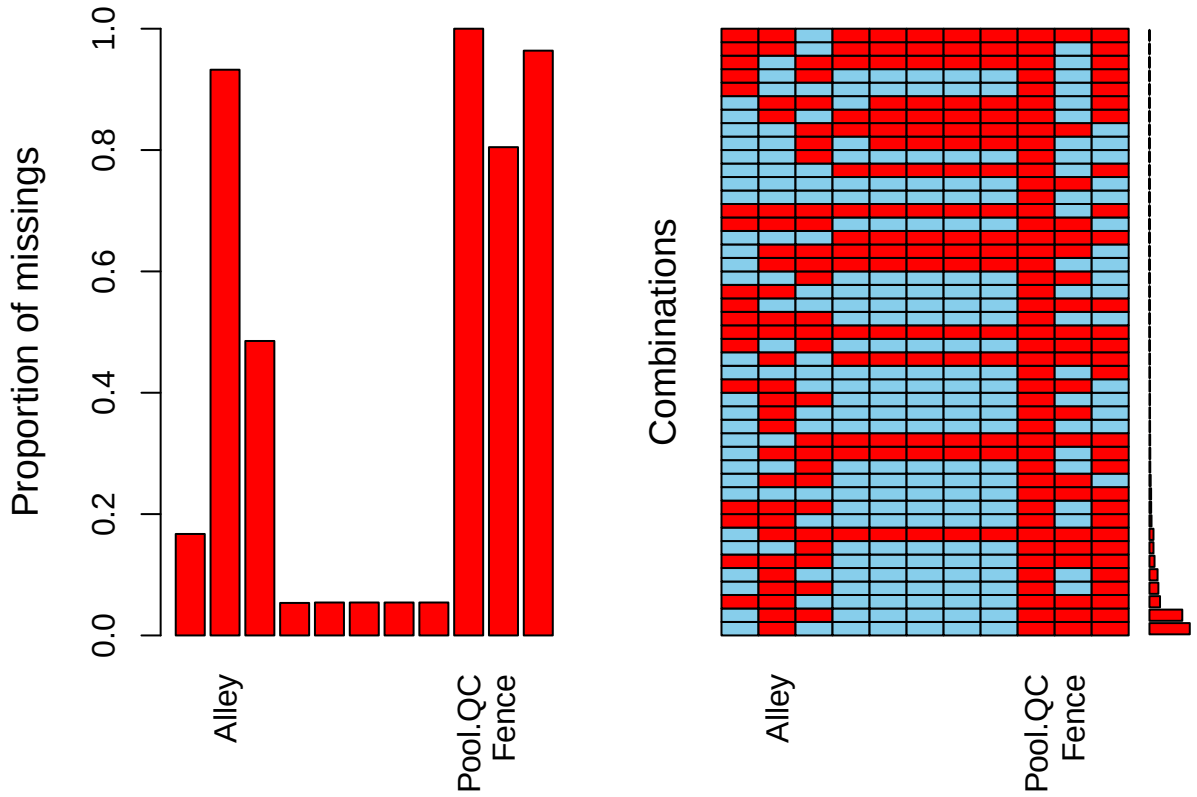
Data Cleansing and Preprocessing

Analyse Missing Values for predictors

```r
sapply(housing_data, function(x) sum(is.na(x)))
```

Below is the graphical representation of NA values.

```
missing_data = housing_data[sapply(housing_data, function(x) sum(is.na(x))) > 100]
aggr(missing_data)
```



Data Transformations

2 columns which have a lot of `NAs` are `Alley` and `Lot Frontage`.

- 2732 out of 2930 records for `Alley` were 'NA's, making it useless as a predictor, so it was removed. Other columns such as `Fireplace.Qu`,`Pool.QC`,`Fence`,`Misc.Feature` contained a lot of NA values and were removed also.
- Lot Frontage is an important feature and replaces the missing values with the mean of the data. SWe will do the same with `Garage.Cars` and `Total.Bsmt.SF` as well.

```
NA_values=data.frame(no_of_na_values=colSums(is.na(housing_data)))
#Replace NA values  with mean
housing_data$Lot.Frontage[which(is.na(housing_data$Lot.Frontage))] = mean(housing_data$Lot.Frontage,na.
housing_data$Total.Bsmt.SF[which(is.na(housing_data$Total.Bsmt.SF))] = mean(housing_data$Total.Bsmt.SF,
housing_data$Garage.Area[which(is.na(housing_data$Garage.Area))] = mean(housing_data$Garage.Area,na.rm =
```

Data Creation with Relevant predictors

```
names(housing_data)[names(housing_data) == "Year.Remod/Add"] = "Yr.Renovated"
house_data_subset = subset(housing_data,
                    select = c (SalePrice,
                                Lot.Area,
                                Gr.Liv.Area,
                                Garage.Cars,
```

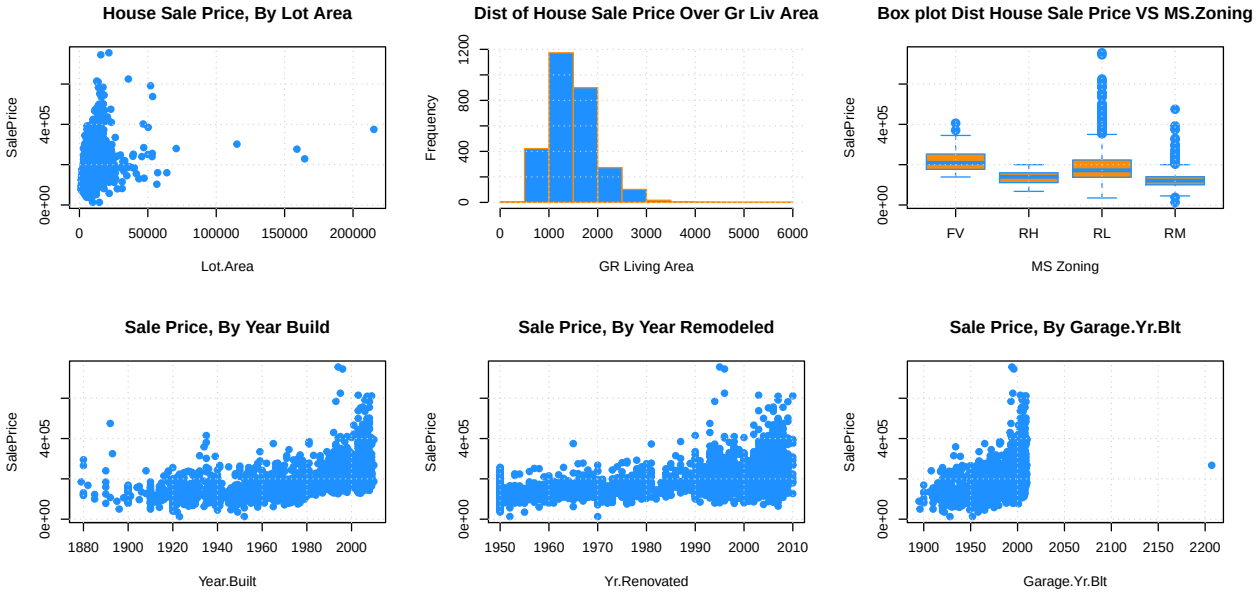
```
                                Total.Bsmt.SF,
                                Bedroom.AbvGr,
                                TotRms.AbvGrd,
                                Full.Bath,
                                Half.Bath,
                                Overall.Qual,
                                Overall.Cond ,
                                Year.Built ,
                                Yr.Renovated,
                                Fireplaces,
                                Bsmt.Full.Bath,
                                Bsmt.Half.Bath,
                                Lot.Frontage,
                                Garage.Area,
                                MS.Zoning,
                                Bsmt.Qual,
                                Kitchen.Qual,
                                Paved.Drive,
                                Foundation,
                                Central.Air,
                                Garage.Type
                                ))
```

The following predictors have been identified as factor predictors.

```
#Filter out and keep only Residential properties
target =  c("A (agr)", "C (all)", "I (all)")
house_data_subset = filter(house_data_subset, !MS.Zoning %in% target)
#Coerce as.factor for categorical variables
house_data_subset$Paved.Drive = as.factor(house_data_subset$Paved.Drive)
house_data_subset$MS.Zoning = as.factor(house_data_subset$MS.Zoning)
house_data_subset$Foundation = as.factor(house_data_subset$Foundation)
house_data_subset$Kitchen.Qual = as.factor(house_data_subset$Kitchen.Qual)
# add NA as factor
house_data_subset$Bsmt.Qual = factor(house_data_subset$Bsmt.Qual)
house_data_subset$Bsmt.Qual = addNA(house_data_subset$Bsmt.Qual, ifany = TRUE)
house_data_subset$Garage.Type = factor(house_data_subset$Garage.Type)
house_data_subset$Garage.Type = addNA(house_data_subset$Garage.Type, ifany = TRUE)
```

Exploratory Data Analysis

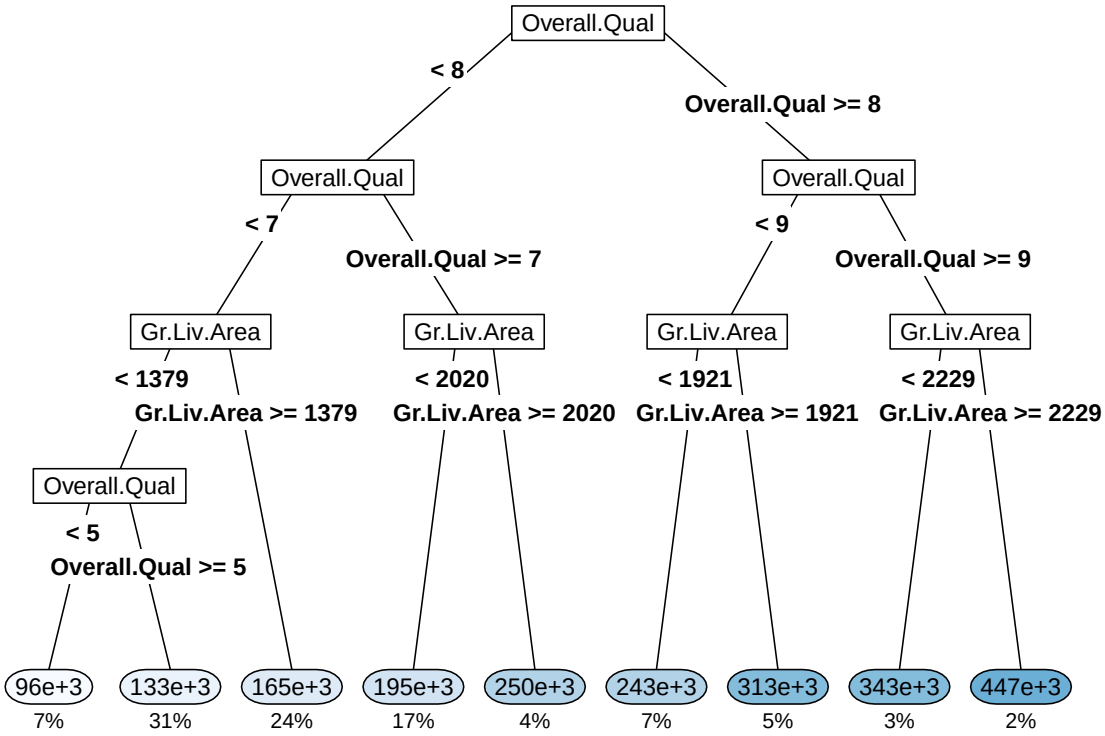
As a part of the exploratory data analysis, various variables are pltted against SalePrice to assess their potential as useful predictors and to get an overall perspective for the dataset.



Model Selection

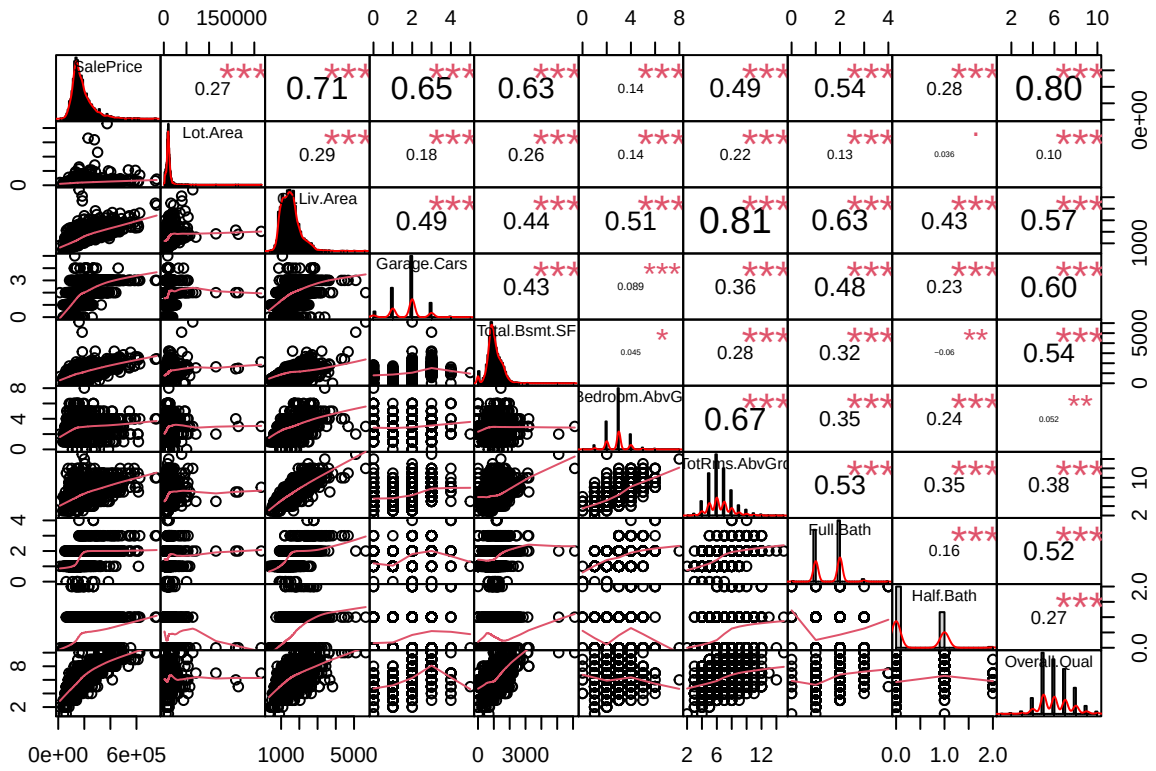
Predictor RelationShip

```
Test_data = house_data_subset
form = as.formula(SalePrice ~ Fireplaces + Overall.Cond + Overall.Qual +
    Lot.Area + Bedroom.AbvGr + Year.Built + Gr.Liv.Area +
    Garage.Area )
mdl = rpart(form, data = Test_data)
rpart.plot(mdl, type = 5, clip.right.labs = FALSE, branch = .3, under = TRUE)
```



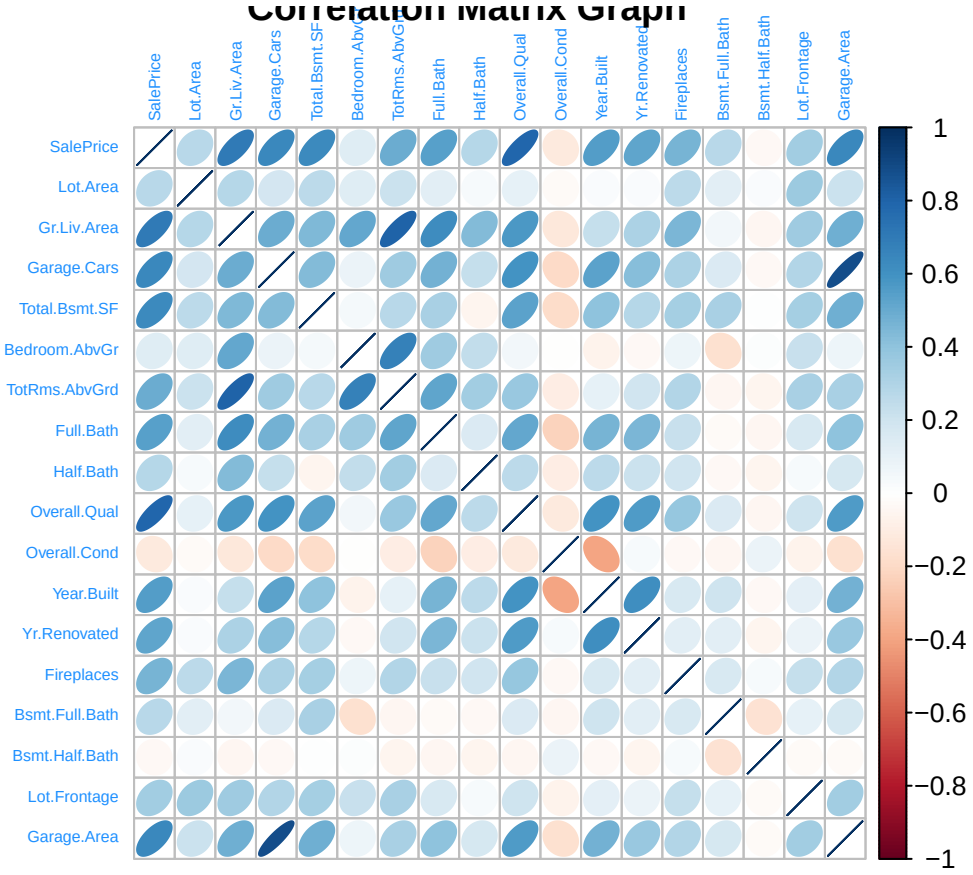
Below a pair plot like function is used, subselecting only numerical variables.

```
chart.Correlation(house_data_subset[,1:10])
```



The correlation between numerical variables is examined, this highlights which variables will potentially be significant in the regression model.

```
data = na.omit(house_data_subset)
corrplot(cor(data[sapply(data, is.numeric)]), method ="ellipse",
         title = "Correlation Matrix Graph",tl.cex = .5, tl.pos ="lt", tl.col ="dodgerblue" )
```




```
house_correlation_var=cor(data.frame(house_data_subset[,1:18]), use = "complete.obs")
sort(abs(house_correlation_var["SalePrice", ]), decreasing = TRUE)
```

```
##      SalePrice   Overall.Qual    Gr.Liv.Area    Garage.Cars    Garage.Area
##     1.00000000     0.79741564     0.70712246     0.64634514     0.64109841
##  Total.Bsmt.SF     Year.Built      Full.Bath   Yr.Renovated  TotRms.AbvGrd
##     0.63003325     0.55131301     0.54105888     0.52605103     0.49315068
##     Fireplaces   Lot.Frontage      Half.Bath       Lot.Area Bsmt.Full.Bath
##     0.46925240     0.34609586     0.28139553     0.27211261     0.27031218
##  Bedroom.AbvGr   Overall.Cond Bsmt.Half.Bath
##     0.13666648     0.11567870     0.03871955
```

High levels of correlation among multiple predictors is confirmed by the above corrplot.

Model Creation

We will now create a few models from the model selection process and perform Model diagnostics. We will check the Adjusted R2 for the models, and also check if any of the features have collinearity issues and high VIF.

- *This is our first attempt to create a Full Additive model and a Stepwise AIC*

```
full_model = lm(SalePrice ~ . , data = house_data_subset)
selected_model_aic = step(full_model, data = house_data_subset, trace = 0)
summary(selected_model_aic)$adj.r
```

```
## [1] 0.8494521
```

This Model has an adjusted R2 of 0.8494521

- *The analysis was started with a two way interactions model of all possible combinations, in order to identify which predictors makes more sense to us. From this assessment a couple of predictors were selected with a high p value. This analysis then utilizes those predictors as the base additive model.*

```
FirstModel = lm(
  SalePrice ~ Fireplaces + TotRms.AbvGrd + Total.Bsmt.SF + Overall.Cond + Overall.Qual + Lot.Area + Kit
  data = house_data_subset
)
```

This Model has adjusted R2 of 0.8366836

- *Will use a Quadratic Relationship here*

```
SecondModel_Quadratic = lm(SalePrice ~ Fireplaces + TotRms.AbvGrd + Total.Bsmt.SF + Overall.Cond + Overa
```

- *AIC Backward selection for the Quadratic model*

```
SecondModel = step(SecondModel_Quadratic,trace = 0)
```

This Model has adjusted R2 of 0.8790733

- *Third model is based on Two Interaction relationships*

```
ThirdModel = lm(SalePrice ~ (Fireplaces + TotRms.AbvGrd + Total.Bsmt.SF + Overall.Cond + Overall.Qual +
```

This Model has adjusted R2 of 0.924139

- *One further model is checked by performing predictor and response transformation, also using an interaction term.*

```
ForthModel = lm(
  log(SalePrice) ~ Fireplaces * Total.Bsmt.SF + Overall.Cond + Overall.Qual +
    log(Lot.Area) + Kitchen.Qual + Bedroom.AbvGr + Year.Built + log(Gr.Liv.Area) +
     Garage.Area + Garage.Cars, data = house_data_subset
)
summary(ForthModel)$adj.r
```

```
## [1] 0.8849385
```

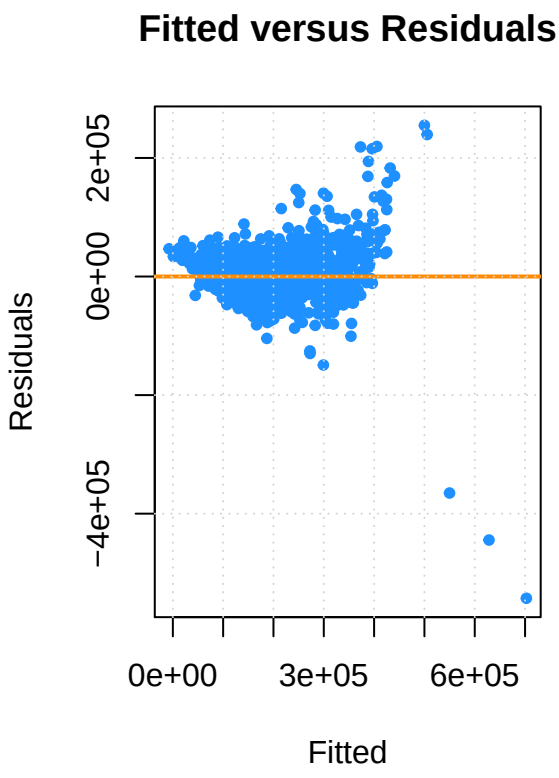
This Model has adjusted R2 of 0.8849385

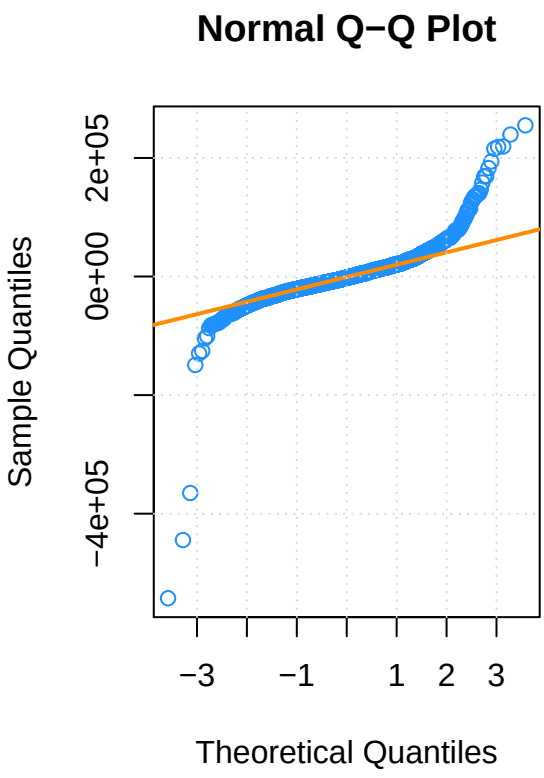
Model Assumptions

The following analyzes the FirstModel, SecondModel , ThirdModel and ForthModel selection process.

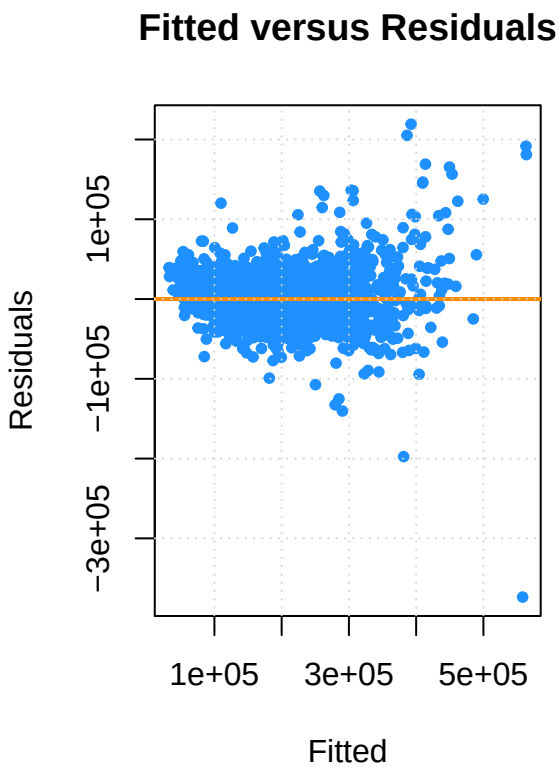
- **Linearity & Constant Variance**

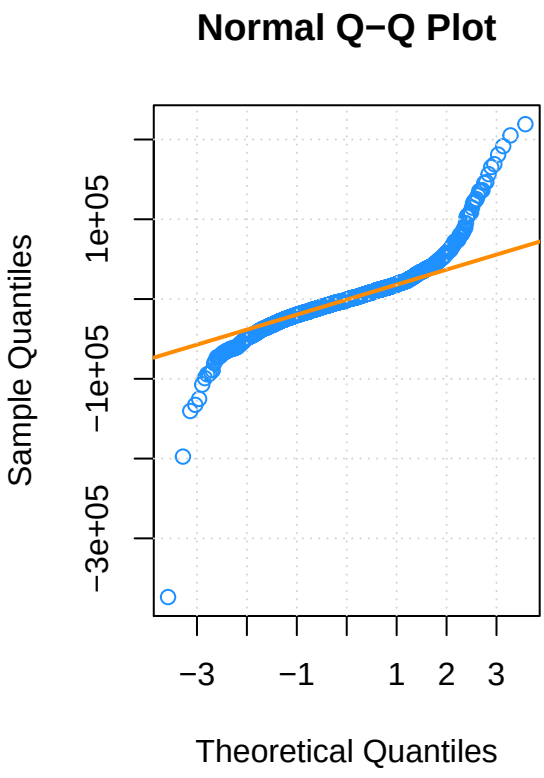
```
par(mfrow=c(4,2))
diagnostics(FirstModel, plotit = TRUE, testit = FALSE)
```



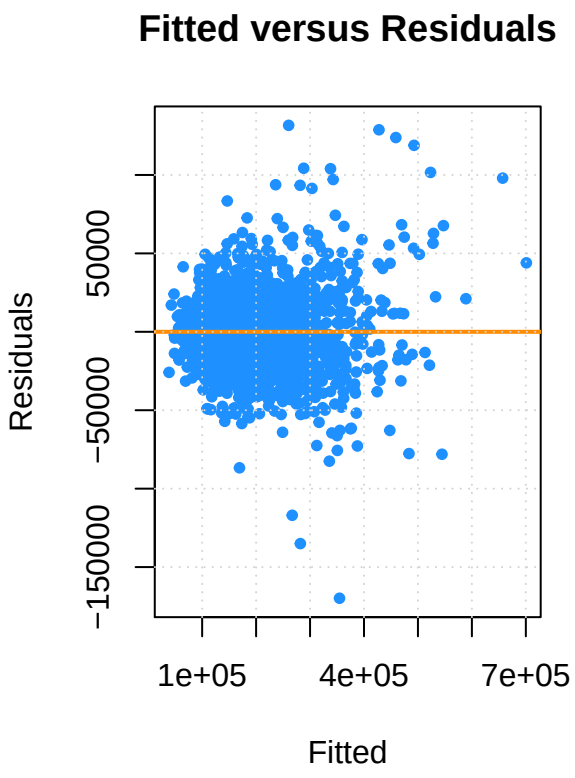


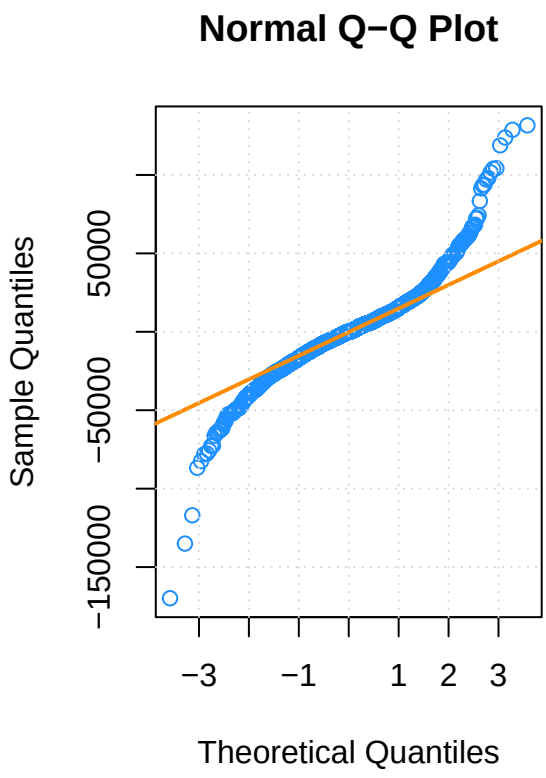

```
diagnostics(SecondModel, plotit = TRUE, testit = FALSE)
```



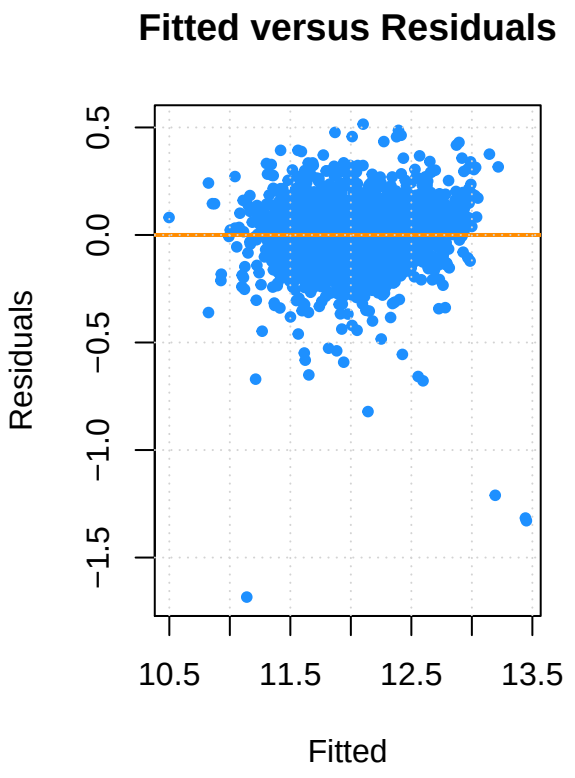


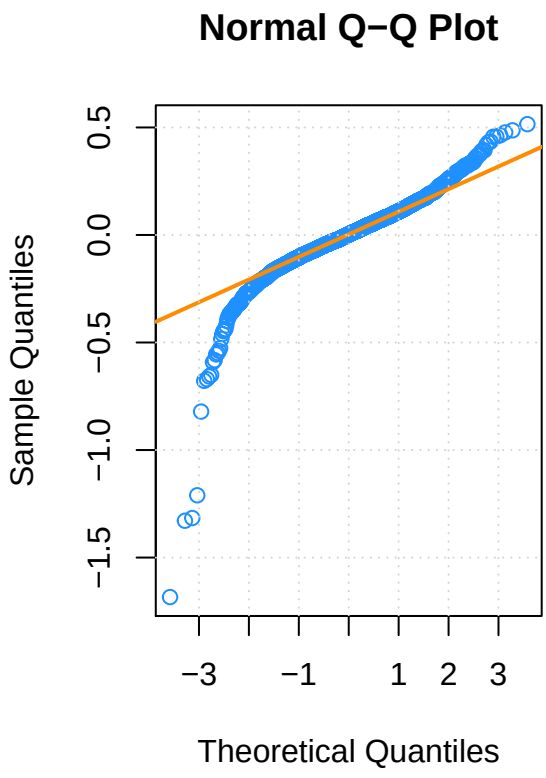

```
diagnostics(ThirdModel, plotit = TRUE, testit = FALSE)
```






```
diagnostics(ForthModel, plotit = TRUE, testit = FALSE)
```





- BP Test Results

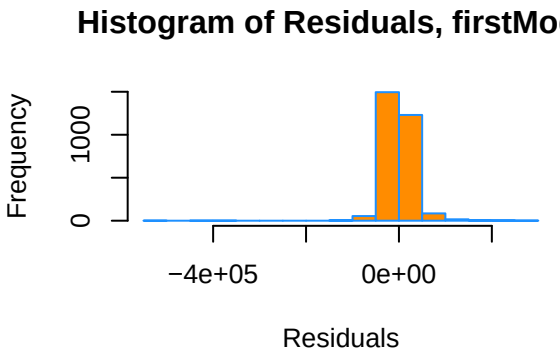
```
M1 = as.numeric(bptest(FirstModel)$p.value)
M2 = bptest(SecondModel)$p.value
```

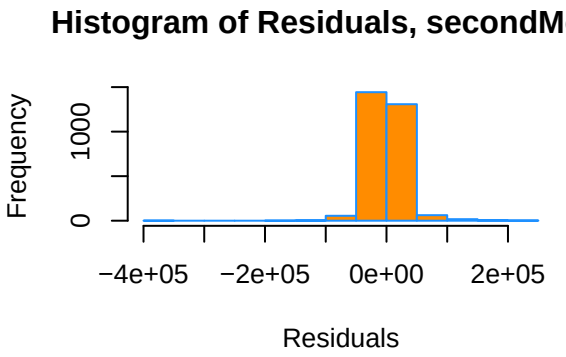
```
M3 = bptest(ThirdModel)$p.value
M4 = bptest(ForthModel)$p.value
val= c(M1,M2,M3,M4)
output = data.frame(
  "ModelName" = c("`Model 1`", "`Model 2`", "`Model 3`", "`Model 4`"),"`BPTestPval`" = val
)
output
```

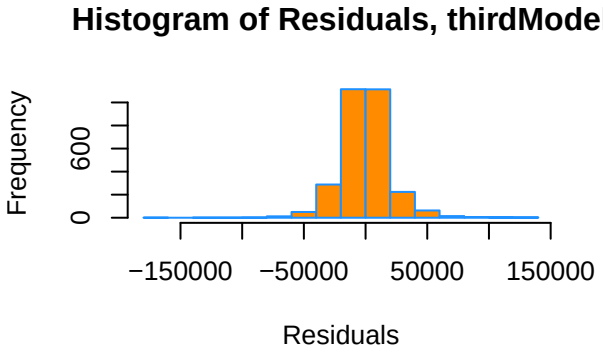
```
##   ModelName X.BPTestPval.
## 1 `Model 1` 1.048867e-153
## 2 `Model 2` 1.683046e-125
## 3 `Model 3`  6.070092e-65
## 4 `Model 4`  1.615120e-78
```

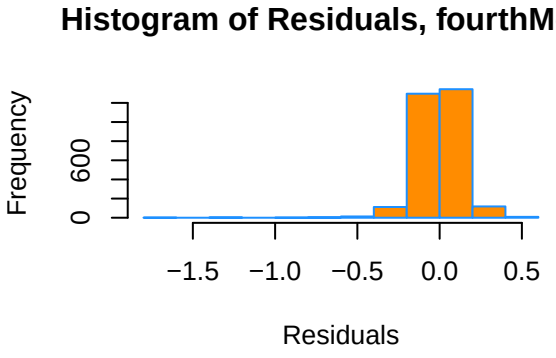
The above results shows that the P value is very low, hence homoscedasticity is rejected (the assumption of constant variance is not violated). In addition the BP value for the fourth model is better when compared to rest of the models.

- **Normailty of errors**









- Shapiro Wilk Test

```
diagnostics(FirstModel, plotit = FALSE, testit = TRUE)
```

```
## $p_val
## [1] 2.535269e-53
##
## $decision
## [1] "Reject"
```

```
diagnostics(SecondModel, plotit = FALSE, testit = TRUE)
```

```
## $p_val
## [1] 5.921164e-45
##
## $decision
## [1] "Reject"
```

```
diagnostics(ThirdModel, plotit = FALSE, testit = TRUE)
```

```
## $p_val
## [1] 2.221852e-34
##
## $decision
## [1] "Reject"
```

```
diagnostics(ForthModel, plotit = FALSE, testit = TRUE)
```

```
## $p_val
## [1] 8.738884e-41
##
## $decision
## [1] "Reject"
```

The above Shapiro Test results shows that the P value is very low, hence we reject the null hypothesis (linearity is suspect).

Unusual Observations

- *Looking at Leverage, Outliers and Influential Data Points*

```
#Leverage
lv_m1 = length(hatvalues(FirstModel)[hatvalues(FirstModel) > 2 * mean(hatvalues(FirstModel))])
lv_m2 = length(hatvalues(SecondModel)[hatvalues(SecondModel) > 2 * mean(hatvalues(SecondModel))])
lv_m3 = length(hatvalues(ThirdModel)[hatvalues(ThirdModel) > 2 * mean(hatvalues(ThirdModel))])
lv_m4 = length(hatvalues(ForthModel)[hatvalues(ForthModel) > 2 * mean(hatvalues(ForthModel))])
# Outliers
ol_m1 = length(rstandard(FirstModel)[abs(rstandard(FirstModel)) > 2])
ol_m2 = length(rstandard(SecondModel)[abs(rstandard(SecondModel)) > 2])
ol_m3 = length(rstandard(ThirdModel)[abs(rstandard(ThirdModel)) > 2])
ol_m4 = length(rstandard(ForthModel)[abs(rstandard(ForthModel)) > 2])
# Influential
inf_obs_m1 = length(cooks.distance(FirstModel)[cooks.distance(FirstModel) > 4 / length(cooks.distance(F
inf_obs_m2 = length(cooks.distance(SecondModel)[cooks.distance(SecondModel) > 4 / length(cooks.distance
inf_obs_m3 = length(cooks.distance(ThirdModel)[cooks.distance(ThirdModel) > 4 / length(cooks.distance(T
inf_obs_m4 = length(cooks.distance(ForthModel)[cooks.distance(ForthModel) > 4 / length(cooks.distance(F
#Summarizing The Result
output = data.frame(
  "ModelName" = c("`Model 1`", "`Model 2`", "`Model 3`", "`Model 4`"),
  "TotalUnfluentialObs" = c(
    lv_m1 ,
    lv_m2,
    lv_m3,
    lv_m4
  ),
  "TotalOutliers" = c(
```

```
    ol_m1 ,
    ol_m2,
    ol_m3,
    ol_m3
  ),
  "InfluentialObs" = c(
    inf_obs_m1 ,
    inf_obs_m2,
    inf_obs_m3,
    inf_obs_m4
  )

)
knitr::kable(output)
```

ModelName	TotalUnfluentialObs	TotalOutliers	InfluentialObs
Model 1	230	88	132
Model 2	258	125	165
Model 3	365	191	272
Model 4	195	191	153

Remove influential observation & Fix normality issue

The Residual vs Residual plot and Normal QQ plot show that that they are not perfect and the test performed above also proves that there is something wrong with the models. and that a variance assumption is violated.

- Next a Box Cox Plot is performed on the model, after which transformations of Response and other variables are performed, to see if BP test values change.

```
salepriceBCMod = caret::BoxCoxTrans(house_data_subset$SalePrice)
house_data_subset = cbind(house_data_subset, salePrice_new=predict(salepriceBCMod, house_data_subset$Sa
LotAreaBCMod = caret::BoxCoxTrans(house_data_subset$Lot.Area)
house_data_subset = cbind(house_data_subset, LotArea_new=predict(LotAreaBCMod, house_data_subset$Lot.Ar
GrLivAreaBCMod = caret::BoxCoxTrans(house_data_subset$Gr.Liv.Area)
house_data_subset = cbind(house_data_subset, GrLivArea_new=predict(GrLivAreaBCMod, house_data_subset$Gr
```

- Final Model Stats

```
cd_model4 = cooks.distance(ForthModel)
# Remove influential observation and use variable above created from Box Cox Plot
ForthModel_fixed = lm(
  salePrice_new ~ Fireplaces + Overall.Cond + Overall.Qual +
    LotArea_new + Bedroom.AbvGr + Year.Built + GrLivArea_new +
    Garage.Area ,
  data = house_data_subset,
  subset = cd_model4 <= 4 / length(cd_model4)
)
summary(ForthModel_fixed)$adj.r
```

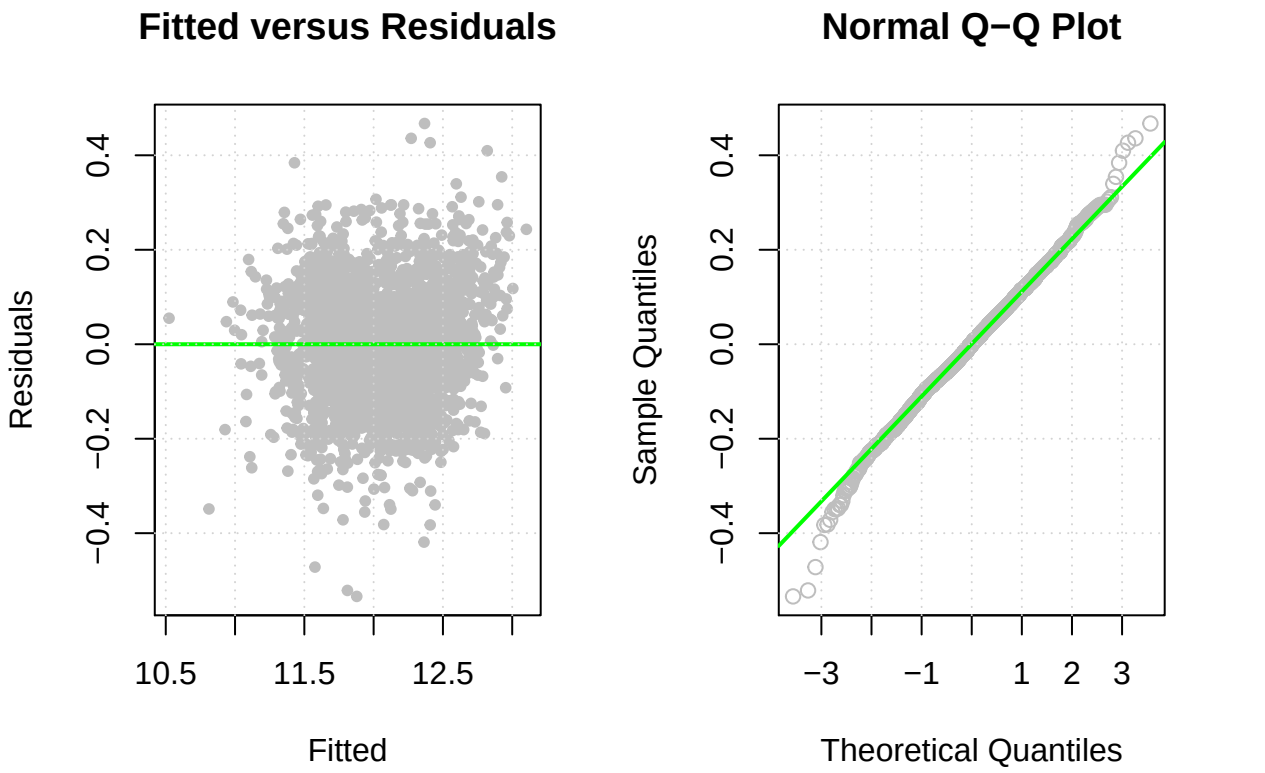
```
## [1] 0.9063674
```

```
bptest(ForthModel_fixed)
```

```
##
##  studentized Breusch-Pagan test
```

```
##
## data:  ForthModel_fixed
## BP = 66.836, df = 8, p-value = 2.089e-11
```

```
diagnostics(ForthModel_fixed, pcol = "grey", lcol = "green")
```



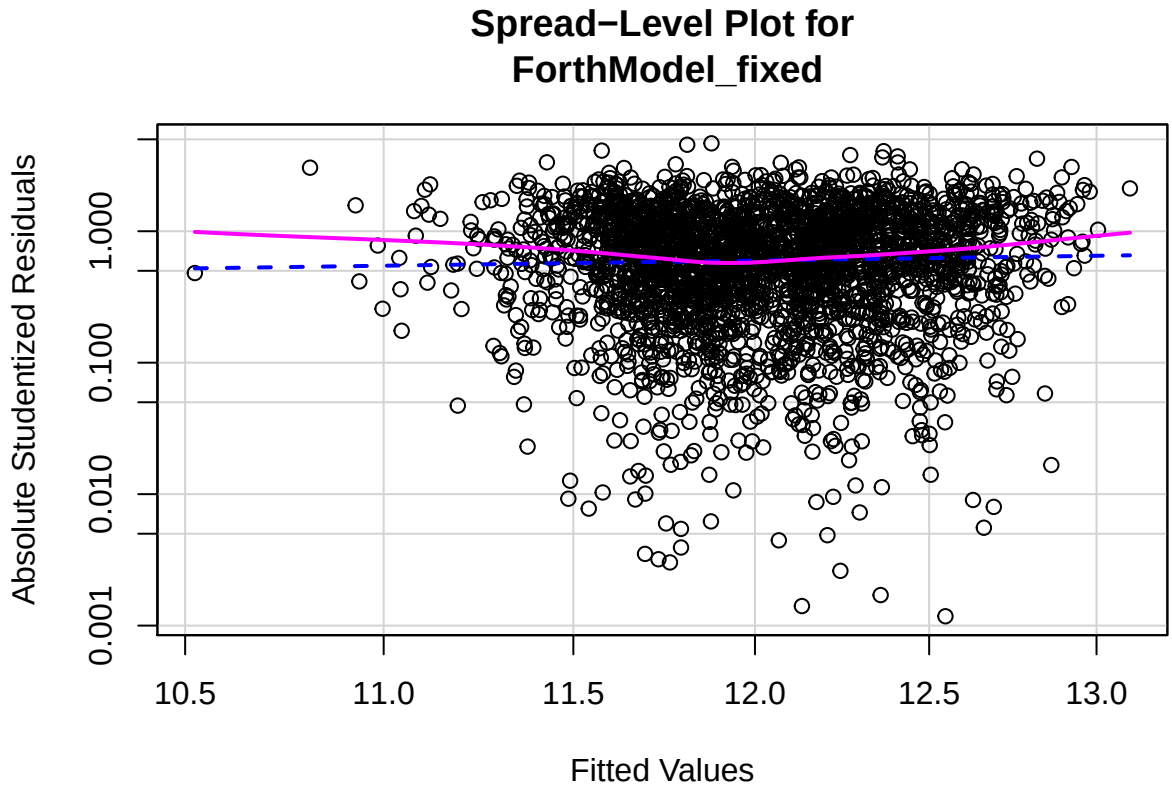

```
## $p_val
## [1] 2.885935e-05
##
## $decision
## [1] "Reject"
```

The 'FixedModel' looks very promising in terms of constant variance and linear relationship as compared to previous models.

```
ncvTest(ForthModel_fixed)
```

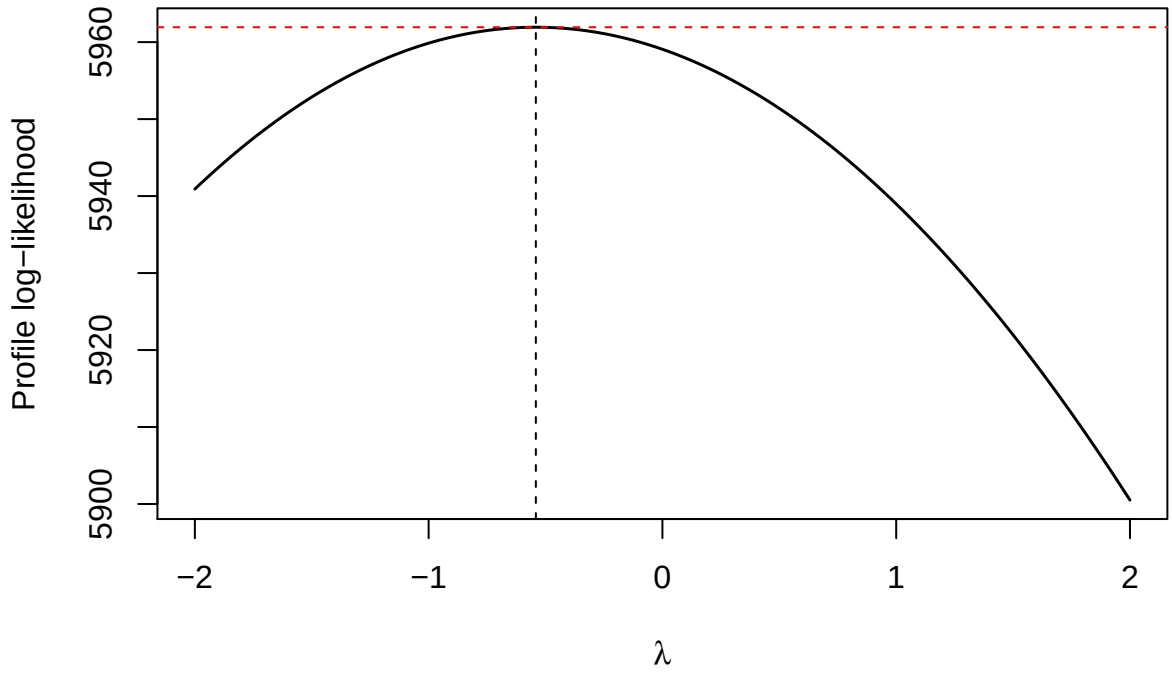
```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 4.234469, Df = 1, p = 0.039611
```

```
spreadLevelPlot(ForthModel_fixed)
```




```
##
## Suggested power transformation:  -0.04237718
```

```
boxcox(ForthModel_fixed, plotit = TRUE)
```




```
## Box-Cox Transformation
##
## Estimation method:  ml
```

```
## Optimal parameter:  -0.5414022
## Loglike:  -5961.935
##
## Summary of transformed variables
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.329   1.361   1.366   1.367   1.372   1.393
```

Based on the Results from box cox plot and ncvTest, it suggested us to do a power transformation of the Reponse Variable

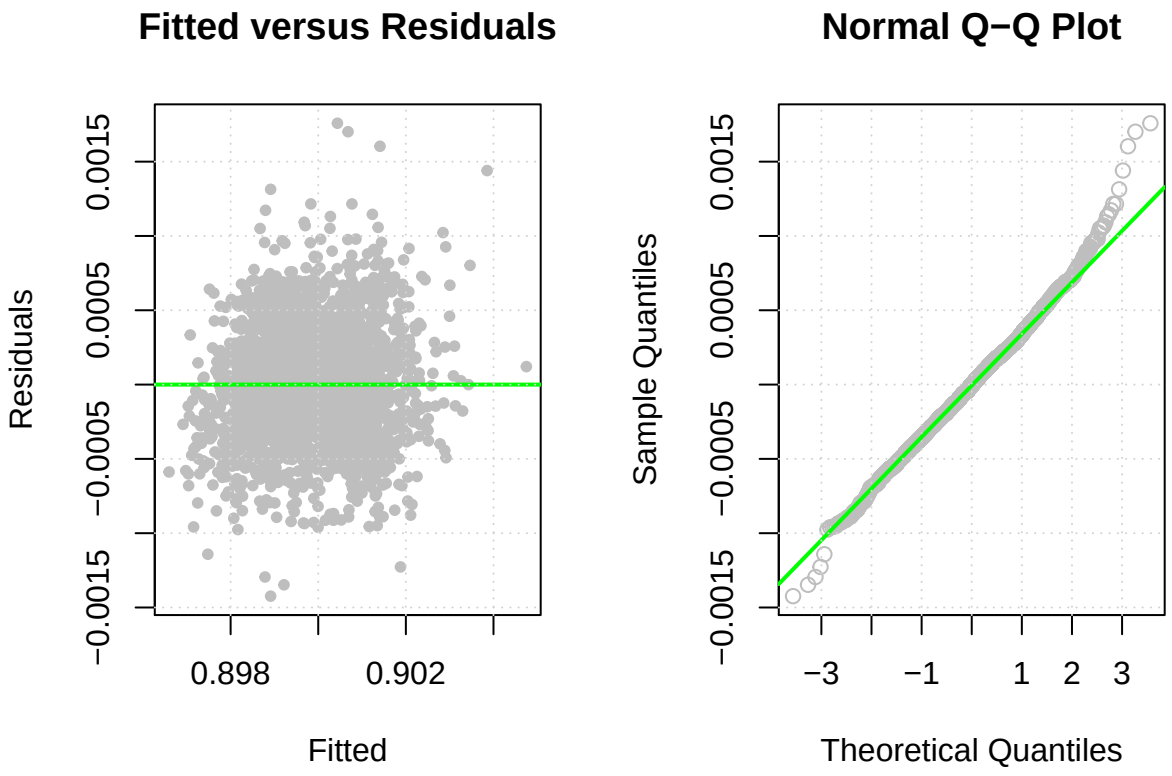
```
cd_model4 = cooks.distance(ForthModel)
# Remove influential observation and use variable above created from Box Cox Plot
ForthModel_fixed = lm(
  (salePrice_new ^ -0.04237718) ~ Fireplaces + Overall.Cond + Overall.Qual +
    LotArea_new + Bedroom.AbvGr + Year.Built + GrLivArea_new +
    Garage.Area ,
  data = house_data_subset,
  subset = cd_model4 <= 4 / length(cd_model4)
)
summary(ForthModel_fixed)$adj.r
```

```
## [1] 0.9071952
```

```
bptest(ForthModel_fixed)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  ForthModel_fixed
## BP = 64.379, df = 8, p-value = 6.401e-11
```

```
diagnostics(ForthModel_fixed, pcol = "grey", lcol = "green")
```



```
## $p_val
## [1] 2.299218e-07
##
## $decision
## [1] "Reject"
```

Evaluations

- *Reporting Adjusted R2 for each model*

```
output = data.frame(
  "ModelName" = c("`Model 1`", "`Model 2`", "`Model 3`", "`Model 4`", "Fixed Model"),
  "AdjustedR2" = c(
    summary(FirstModel)$adj.r.squared ,
    summary(SecondModel)$adj.r.squared,
    summary(ThirdModel)$adj.r.squared,
    summary(ForthModel)$adj.r.squared,
    summary(ForthModel_fixed)$adj.r.squared
  )

)
knitr::kable(output)
```

ModelName	AdjustedR2
Model 1	0.8366836
Model 2	0.8790733
Model 3	0.9241390
Model 4	0.8849385
Fixed Model	0.9071952

Fixed model is the leading model in terms of Adjusted R2.

- *Calculate RMSE for each model*

```
calc_loocv_rmse = function(model) {
sqrt(mean((resid(model) / (1 - hatvalues(model))) ^ 2))
}
output = data.frame(
  "ModelName" = c("`Model 1`", "`Model 2`", "`Model 3`", "`Model 4`", "`Fixed Model`"),
  "RMSE" = c(
    calc_loocv_rmse(FirstModel) ,
    calc_loocv_rmse(SecondModel),
    calc_loocv_rmse(ThirdModel),
   calc_loocv_rmse(ForthModel),
   calc_loocv_rmse(ForthModel_fixed)
  )

)
knitr::kable(output)
```

ModelName	RMSE
Model 1	Inf
Model 2	Inf
Model 3	Inf

ModelName	RMSE
Model 4	Inf
Fixed Model	0.0003638

The fixed Model has a much better LOOCV RMSE, as compared to other models which show infinity.

- *Checking Variance Inflation for each of the model*

```
# VIF
sort(vif(ForthModel_fixed))
```

```
##   LotArea_new  Overall.Cond    Fireplaces Bedroom.AbvGr   Garage.Area
##      1.288926      1.291559      1.367961      1.744734      1.785370
##    Year.Built  Overall.Qual GrLivArea_new
##      2.084127      2.663422      3.110250
```

```
sum(vif(FirstModel)>5)/length(coef(FirstModel))
```

```
## [1] 0.15625
```

```
sum(vif(SecondModel)>5)/length(coef(SecondModel))
```

```
## [1] 0.825
```

```
sum(vif(ForthModel)>5)/length(coef(ForthModel))
```

```
## [1] 0.25
```

```
sum(vif(ForthModel_fixed)>5)/length(coef(ForthModel_fixed))
```

```
## [1] 0
```

- *AIC of all Models*

```
output = data.frame(
  "ModelName" = c("`Model 1`", "`Model 2`", "`Model 3`", "`Model 4`", "`Fixed Model`"),
  "AIC" = c(AIC(FirstModel) ,
            AIC(SecondModel),
            AIC(ThirdModel),
            AIC(ForthModel),
            AIC(ForthModel_fixed)
            )
)
knitr::kable(output)
```

ModelName	AIC
Model 1	68409.047
Model 2	67546.083
Model 3	66509.444
Model 4	-3376.961
Fixed Model	-35723.186

In addition to the above statistics, the model needs to be validated via fit and predictions on a training and testing set.

Train VS Test Split for RMSE

```
house_data_subset_mod = subset(house_data_subset,  cd_model4 <=
    4/length(cd_model4))
```

```
set.seed(108)
n = nrow(house_data_subset_mod)
house_data_subset_idx=sample(nrow(house_data_subset_mod),round(nrow(house_data_subset) / 2))
house_data_subset_trn = house_data_subset_mod[house_data_subset_idx, ]
house_data_subset_tst = house_data_subset_mod[-house_data_subset_idx, ]
```

- Recreated forth Model on Train Data

```
ForthModel_fixed_train = lm(
  salePrice_new  ~ Fireplaces + Overall.Cond + Overall.Qual +
    LotArea_new + Bedroom.AbvGr + Year.Built + GrLivArea_new +
    Garage.Area ,
  data = house_data_subset_trn

)
```

RMSE for the train and test Model

```
sqrt(mean((house_data_subset_trn$salePrice_new - predict(ForthModel_fixed_train, house_data_subset_trn)
```

```
## [1] 0.1174843
```

```
sqrt(mean((house_data_subset_tst$salePrice_new - predict(ForthModel_fixed_train, house_data_subset_tst)
```

```
## [1] 0.1129396
```

The results delivered a train RMSE of 0.1148181 and a test RMSE 0.1159567. These are very close and do not seem to infer over or under fitting. To further analyze ForthModel_fixed, a forward and backward AIC and BIC is ran on ForthModel_fixed. All four methods returned the same model. Calling this model ForthModel_AIC_BIC an anova test is conducted (between ForthModel_fixed and ForthModel_AIC_BIC) and the RMSEs from the training and the testing sets of both models are compared.

```
ForthModel_AIC_BIC = lm(salePrice_new ~ Fireplaces + Overall.Cond + Overall.Qual + LotArea_new + Bedroo
  data = house_data_subset_trn)
```

```
anova(ForthModel_fixed_train, ForthModel_AIC_BIC)
```

```
## Analysis of Variance Table
##
## Model 1: salePrice_new ~ Fireplaces + Overall.Cond + Overall.Qual + LotArea_new +
##     Bedroom.AbvGr + Year.Built + GrLivArea_new + Garage.Area
## Model 2: salePrice_new ~ Fireplaces + Overall.Cond + Overall.Qual + LotArea_new +
##     Bedroom.AbvGr + Year.Built + GrLivArea_new
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1   1441 20.014
## 2   1442 21.868 -1    -1.854 133.49 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ANOVA Test

The ANOVA test returns a very low p-value, consequently the larger model, aka the ForthModel_fixed, is preferred.

Final Output

```
output = data.frame(
  "ModelName" = c("`ForthModel`", "`ForthModel_AIC_BIC`"),
  "Train_RMSE" = c(
     forth_trn,
    forthaic_trn
  ),
  "Test_RMSE" = c(
    forth_tst,
    forthaic_tst
  )

)
knitr::kable(output)
```

ModelName	Train_RMSE	Test_RMSE
ForthModel	0.1174843	0.1129396
ForthModel_AIC_BIC	0.1228056	0.1168950

Results & Discussion

Based on the results above, we conclude that it is not possible to make any further improvements to the model. Hence Final_Model is the final and best model.

The following presents the final predictors:

```
Final_Model = ForthModel_fixed
length(coef(Final_Model))
```

```
## [1] 9
```

```
names(coef(Final_Model))
```

```
## [1] "(Intercept)"   "Fireplaces"    "Overall.Cond"  "Overall.Qual"
## [5] "LotArea_new"   "Bedroom.AbvGr" "Year.Built"    "GrLivArea_new"
## [9] "Garage.Area"
```

```
summary(Final_Model)
```

```
##
## Call:
## lm(formula = (salePrice_new^-0.04237718) ~ Fireplaces + Overall.Cond +
##     Overall.Qual + LotArea_new + Bedroom.AbvGr + Year.Built +
##     GrLivArea_new + Garage.Area, data = house_data_subset, subset = cd_model4 <=
##     4/length(cd_model4))
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -1.424e-03 -2.405e-04 -3.750e-06  2.275e-04  1.758e-03
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    9.420e-01  7.552e-04 1247.41   <2e-16 ***
## Fireplaces    -1.453e-04  1.273e-05  -11.41   <2e-16 ***
```

```
## Overall.Cond  -1.788e-04  7.395e-06  -24.18   <2e-16 ***
## Overall.Qual  -2.925e-04  8.337e-06  -35.08   <2e-16 ***
## LotArea_new   -4.032e-04  1.569e-05  -25.70   <2e-16 ***
## Bedroom.AbvGr  1.169e-04  1.141e-05   10.24   <2e-16 ***
## Year.Built    -1.275e-05  3.406e-07  -37.45   <2e-16 ***
## GrLivArea_new -1.430e-03  3.885e-05  -36.80   <2e-16 ***
## Garage.Area   -6.730e-07  4.481e-08  -15.02   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0003631 on 2739 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.9075, Adjusted R-squared:  0.9072
## F-statistic:  3358 on 8 and 2739 DF,  p-value: < 2.2e-16
```

The significant predictors.

```
summary(Final_Model)$coefficients[summary(Final_Model)$coefficients[ ,4] < 0.05,]
```

```
##                    Estimate   Std. Error    t value      Pr(>|t|)
## (Intercept)    9.420242e-01 7.551853e-04 1247.40792  0.000000e+00
## Fireplaces    -1.452644e-04 1.273207e-05  -11.40933  1.721612e-29
## Overall.Cond  -1.788365e-04 7.395033e-06  -24.18333 2.864186e-117
## Overall.Qual  -2.924783e-04 8.336759e-06  -35.08297 5.051918e-223
## LotArea_new   -4.031799e-04 1.568608e-05  -25.70305 1.051763e-130
## Bedroom.AbvGr  1.168619e-04 1.140782e-05   10.24401  3.418634e-24
## Year.Built    -1.275450e-05 3.406098e-07  -37.44609 3.503978e-248
## GrLivArea_new -1.429505e-03 3.884768e-05  -36.79771 3.189816e-241
## Garage.Area   -6.729665e-07 4.481273e-08  -15.01731  4.802337e-49
```

Below is the accuracy of the model

```
summary(Final_Model)$r.squared
```

```
## [1] 0.9074655
```