ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ

ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS

BUSINESS
ANALYTICS
Master of Science

# MSc in Business Analytics - Part time

# Statistics for Business Analytics I

## Course Assignment January 2020

## Ames Iowa Housing Dataset 29

**Professor: Mr. Ntzoufras Ioannis**

**Student: Arseniou Evangelia (p2822026)**

# Contents

# Abstract

This paper refers to a data set of 1500 property sales in Ames, Iowa from the Ames Assessor's Office between 2006 and 2010. The data set contains 82 explanatory variables which are focused on the quality and quantity of many physical attributes of the property. The main goal is to compare the predictive performance of multiple regression models and to identify the best model for predicting the prices of the properties. Data cleaning and variable transformation were mandatory as the data set includes many missing values with not well – specified variables. The multicollinearity was also a problem as many attributes were highly correlated. Before the creation of the multiple regression model, categorical variables were converted in dummies. Lasso was used in order to reduce the number of columns. Then, the first model according to AIC in the Stepwise procedure was estimated. At this point, we have to refer that only variables with correlation greater than the absolute value of 0.25 were kept. Based on various visualizations of the variables, we could detect some possible transformations of the response and predictor variables in order to face the violation of residual's assumptions of the model. On the whole three models were estimated and based on Leave One Out and 10 – Fold cross validation methods we kept the model with the minimum RMSE. Finally, a test data set was used to assess the out – of – sample predictive ability of the model.

## 1. Introduction

A home is a major purchase. Customers have different criteria to decide their dream house. For example, a family with young children might need a fenced yard for safety with many bedrooms and bathrooms while a person who uses a wheelchair might need an easily accessible flat house. On the other hand, a great amount of people wants a garage area especially in urban areas. After all, the determining factor to decide a potential customer the purchase of the house is the price. But what are the factors that define the price of the house? In this paper we are going to answer this question based on a data set from the Ames Assessor's Office.

The data set has 82 explanatory variables which include 23 nominal, 23 ordinal, 14 discrete, and 20 continuous variables (and 2 additional observation identifiers). Continuous variables determine the various area dimensions such as the size of the living area, the basement, the garage area while discrete variables contain the year built / remodelled, the number of bedrooms, bathrooms, kitchens, fireplaces etc. Nominal variables describe the name of the neighbourhoods, the garage type, the sale type etc. Ordinal variables refer to the quality and condition of different house parts and utilities. This project dataset will help us to identify the most important variables and to define the best regression model for predicting the housing prices in Ames, Iowa.

## 2. Descriptive analysis and exploratory data analysis

The first step is to import the dataset in R program. Some variables are not well – specified and contain missing values. The data cleaning and transformation seems to be crucial before we continue in further analysis.

**Data cleaning**: 19 variables were found with missing values.

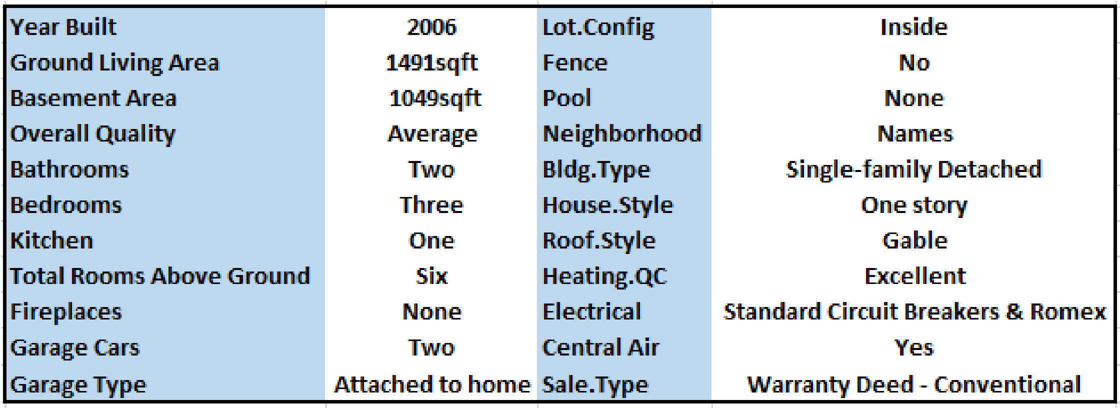
Most of the categorical variables have NA as level which actually means No – and the characteristic of each variable. (e.g. 'No Garage'). So, the only thing was to define a new 'NA' level for every categorical variable which contains the missing values. An exception was 'Mas.Vnr.Type' which does not contain NA as level so a new 'Missing' level was created. Also, 'Electrical' has only one missing value so it was replaced with the most common level.

Regarding numeric variables, missing values where replaced with zero. Only for ‘Lot.Frontage’ as zero is illogical and it had to be replaced with the median. and for ‘Garage.Yr.Blt’ NAs were replaced with the values of ‘YearBuilt’  (the original construction date of house).

Finally, the first three columns (X, Order, PID) were dropped as they do not add any further information in our analysis. Ordinal variables were also converted in integers so actually the data set contains 52 numeric and 28 categorical variables.

## Descriptive Analysis

After defining the correct type of all variables, the data set was separated in two data frames, one for numeric variables and the other for factors. Taking the descriptive measures for numeric variables, we observe an unusual max value in ‘Garage.Yr.Bld’ (max = 2027). This could be a typographic error so we have to replace it with the ‘Year.Remod.Add’ which is 2007 for this row. Table 1 shows some of the average house characteristics, based on the mean for numeric variables and the mode for categorical variables.



| | | | |
|---|---|---|---|
| Year Built | 2006 | Lot.Config | Inside |
| Ground Living Area | 1491sqft | Fence | No |
| Basement Area | 1049sqft | Pool | None |
| Overall Quality | Average | Neighborhood | Names |
| Bathrooms | Two | Bldg.Type | Single-family Detached |
| Bedrooms | Three | House.Style | One story |
| Kitchen | One | Roof.Style | Gable |
| Total Rooms Above Ground | Six | Heating.QC | Excellent |
| Fireplaces | None | Electrical | Standard Circuit Breakers & Romex |
| Garage Cars | Two | Central Air | Yes |
| Garage Type | Attached to home | Sale.Type | Warranty Deed - Conventional |

**Table 1. Average House Characteristics**

## Visualizations of the most important variables

We create Q – Q Plots and Histograms for all numeric variables and Bar plots for categorical variables in order to gain a better insight of our data. Based on Histograms we observed that some numeric variables are positively skewed. These variables are Overall.Qual, Lot.Frontage, BsmtUnfSF, TotalBsmtSF, X1stFlrSF, GarageArea, Gr.Liv.Area, and SalePrice and we have to keep them in mind as they might be log – transformed later. Moreover, from Bar plots Garage.Cars – Garage.Area, Garage.Qual – Garage.Cont and Exter.Qual – Kitchen.Qual seem to be highly correlated which will probably lead to multicollinearity problems. In statistics, multicollinearity (also collinearity) is a phenomenon in which one predictor variable in a multiple regression model can be linearly predicted from the others with a substantial degree of accuracy.

Some of the most important visualizations are shown below.

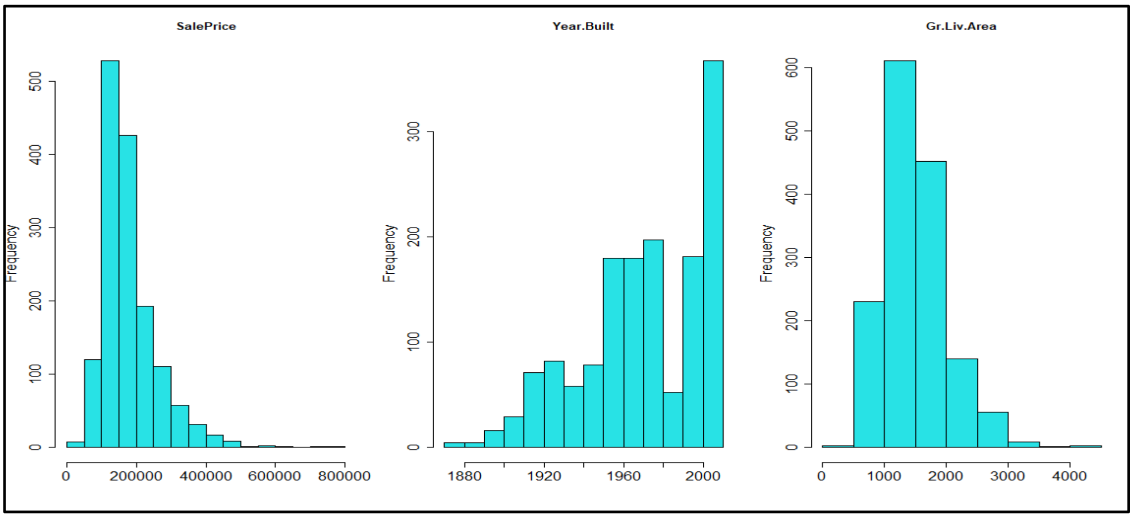


**Figure 2. Histograms of the most important variables**

As we have already mentioned Sale.Price is positively skewed which means that the majority of people choose to buy not very expensive houses as they cannot afford them. Gr.Liv.Area is also positively skewed with the living area of house to be mostly from 1000 to 1500 square ft. Moreover, a high percentage of houses were built after 1950 with an obvious reduction between 1980 to 1990.

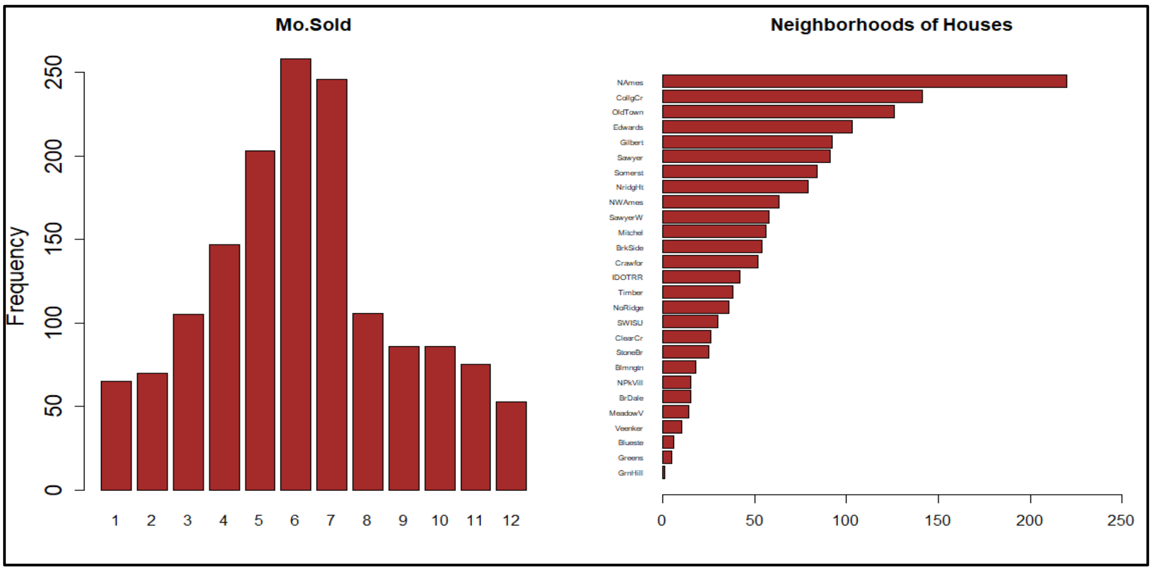


**Figure 3. Bar Plots of the most important variables**

Based on Bar Plots we could easily conclude that most of the houses were sold during summer seasons and especially between May and July with most of them located in North Ames.

## 3. Pairwise comparisons

In this part we would like to examine the relationship of variables in pairs. For this purpose, scatter plots were created and Pearson's coefficient was calculated to quantify the degree of correlation between the numeric variables. In order to examine the relationship between SalesPrice and categorical variables we will use appropriate hypothesis tests for two independent samples and box – plots. In Figure 3 we could see the correlations between the **numeric variables** and our response variable SalesPrice.

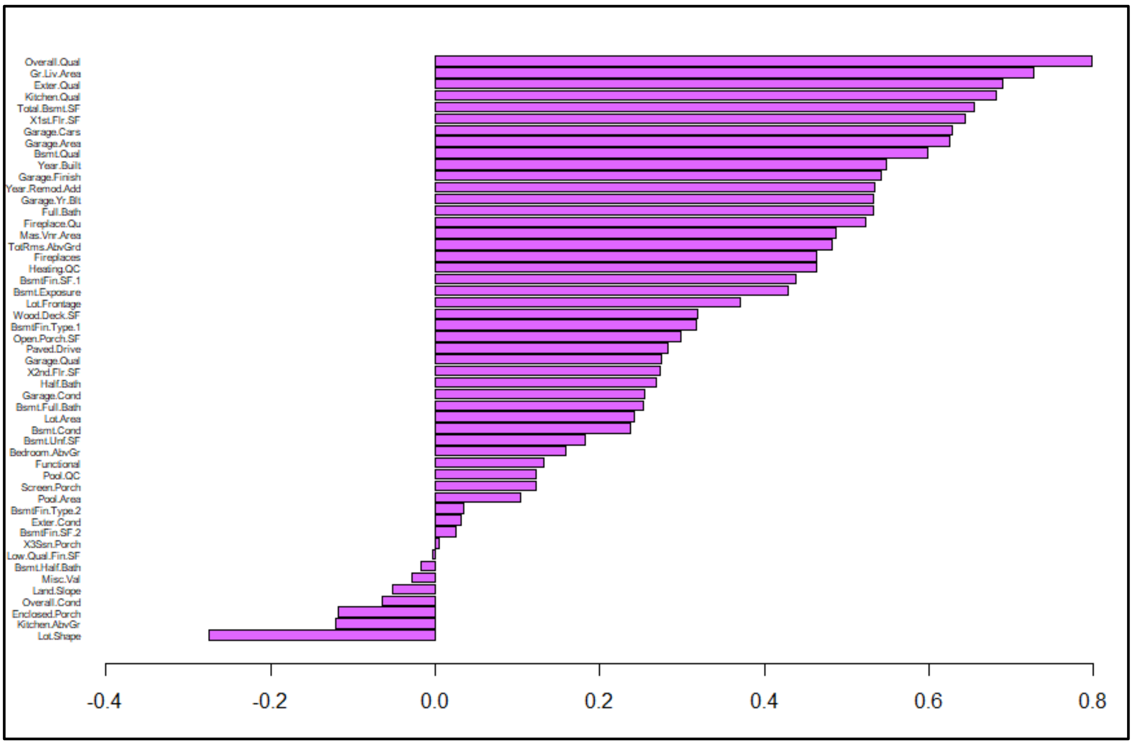


**Figure 4. Correlations between Sale.Price and numeric variables**

The features most correlated with SalePrice are Overall.Qual and GrLivArea, followed by ExterQual, KitchenQual, and Total.Bsmt.Sf. As we already notice some of those variables are highly correlated with each other so a further investigation is necessary.

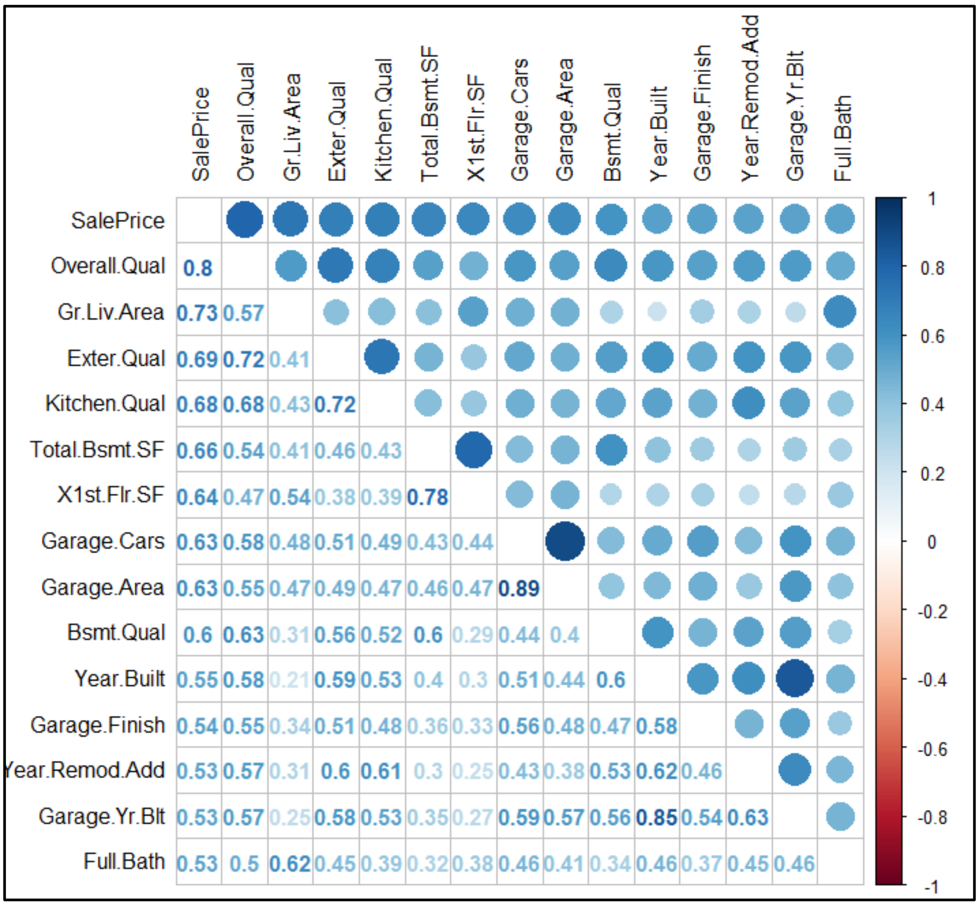


**Figure 5. Correlations between top 15 numeric variables and SalePrice**

From Figure 4 we could easily conclude that: Garage.Yr.Blt – Year.Built and Garage.Area – Garage.Cars are the most highly correlated variables (correlation > 0.80). In order to avoid multicollinearity in regression we have to drop one of them, the one that is less correlated with our response variable. In particular, the following variables will be dropped: Garage.Yr.Blt and Garage.Cars. If we repeat this procedure, we exclude from our data set three more variables: Garage.Cond (Garage.Cond - Garage.Qual, cor= 0.94), Pool.Area (Pool.Area - Pool.QC, cor = 0.98), BsmtFin.SF.2 (BsmtFin.SF.2 - BsmtFin.Type.2, cor = 0.80).

Regarding the categorical variables, it will be interesting to examine SalePrice with Neighborhood, YrSold, Street and Central Air. For this purpose, we carry out the appropriate tests for 2 independent samples with 1 quantitative (SalePrice) and 1 binary variable (Street and Central Air accordingly). After the tests we have concluded that there is a significant difference in the median of the SalePrice between Gravel and Paved (wilcox.test, reject Ho, p.value = 0.005 < 0.05) and in the median of the SalePrice between those houses who have Central Air and those who do not have (wilcox.test, reject

Ho, p.value < 2.2e-16 < 0.05). The first two are factors with many levels so Kruskal – Wallis test was used for non – normal variables. After the implementation of the above tests, it was emerged that there is a significant difference in the median of the SalePrice between some Neighbourhood groups (Kruskal.wallis test, reject Ho, p.value < 2.2e-16 < 0.05) but there is no significant difference in the median of the SalePrice between YrSold groups (Kruskal.wallis test, do not reject Ho, p.value = 0.377 > 0.05). Our conclusions are more easily visible in Figure 6 below.

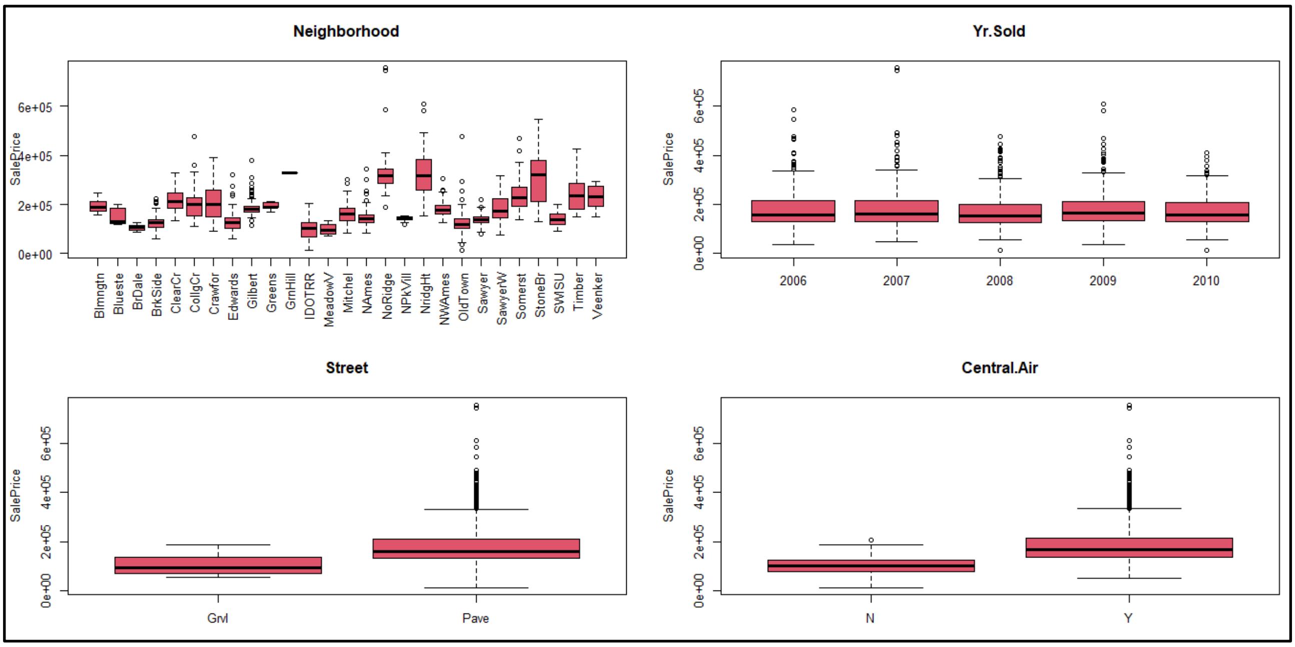


**Figure 6. Box – Plots of SalePrice and some of the most important Categorical variables**

## 4. Predictive models

In this chapter we are going to create a multiple regression model having as response variable the SalePrice. Before we continue, we have first to convert all variables into numeric. Thus, categorical variables will be transformed to dummies. Dummy variables or binary variables are commonly used in statistical analyses and in more simple descriptive statistics. A dummy column is one which has a value of one when a categorical event occurs and a zero when it does not occur. After the transformation the two data frames where concatenated again in one numeric dataset. The new merged dataset's columns have increased as it was expected after dummy transformation and contains 235 variables. At this point, we would like to keep only the variables that are correlated with SalePrice greater than the absolute value of 0,25 so we end up having only 46 attributes. Lasso was then used for further attribute reduction. When we have an extremely large number of variables the best strategy is Lasso with $\lambda_{lse}$. Lasso kept only 25 variables. As the focus is on the prediction of the SalePrice we used Stepwise method according to AIC. It seems that the best model includes only 19 variables which appear to be more significant.

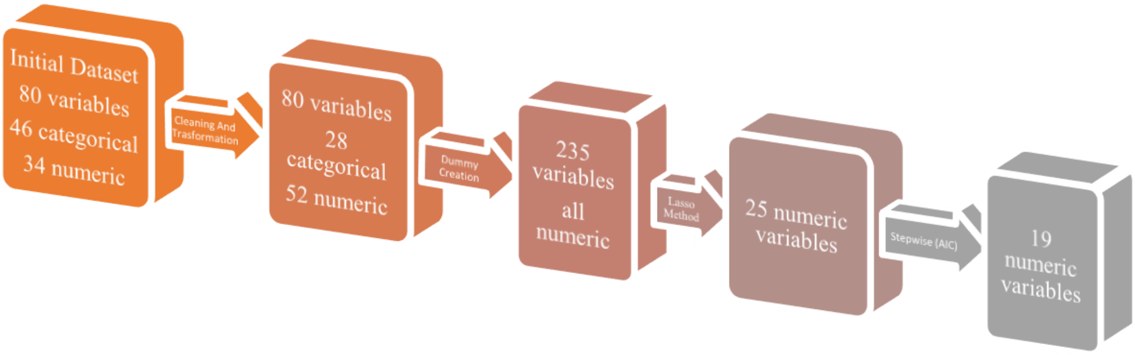
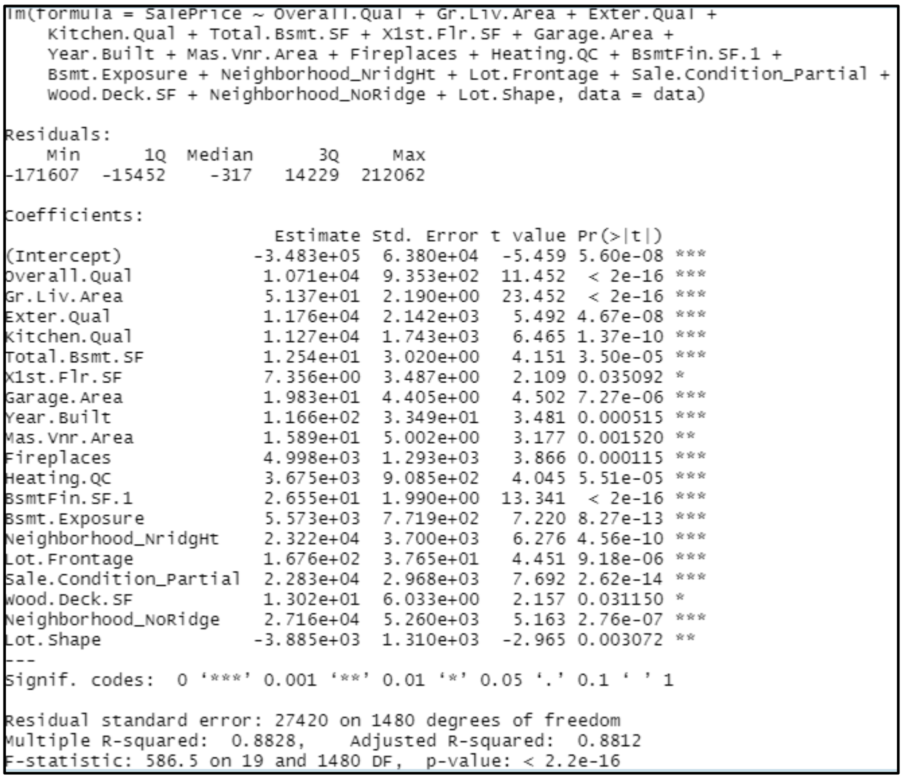


**Figure 7. Data Procedure to the final variables**

Table 2 includes the estimated model of stepwise:



```
lm(formula = SalePrice ~ Overall.Qual + Gr.Liv.Area + Exter.Qual +
    Kitchen.Qual + Total.Bsmt.SF + X1st.Flr.SF + Garage.Area +
    Year.Built + Mas.Vnr.Area + Fireplaces + Heating.QC + BsmtFin.SF.1 +
    Bsmt.Exposure + Neighborhood_NridgHt + Lot.Frontage + Sale.Condition_Partial +
    Wood.Deck.SF + Neighborhood_NoRidge + Lot.Shape, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-171607  -15452    -317   14229  212062

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)            -3.483e+05  6.380e+04  -5.459 5.60e-08 ***
Overall.Qual            1.071e+04  9.353e+02  11.452  < 2e-16 ***
Gr.Liv.Area             5.137e+01  2.190e+00  23.452  < 2e-16 ***
Exter.Qual              1.176e+04  2.142e+03   5.492 4.67e-08 ***
Kitchen.Qual            1.127e+04  1.743e+03   6.465 1.37e-10 ***
Total.Bsmt.SF           1.254e+01  3.020e+00   4.151 3.50e-05 ***
X1st.Flr.SF             7.356e+00  3.487e+00   2.109 0.035092 *
Garage.Area             1.983e+01  4.405e+00   4.502 7.27e-06 ***
Year.Built              1.166e+02  3.349e+01   3.481 0.000515 ***
Mas.Vnr.Area            1.589e+01  5.002e+00   3.177 0.001520 **
Fireplaces              4.998e+03  1.293e+03   3.866 0.000115 ***
Heating.QC              3.675e+03  9.085e+02   4.045 5.51e-05 ***
BsmtFin.SF.1            2.655e+01  1.990e+00  13.341  < 2e-16 ***
Bsmt.Exposure           5.573e+03  7.719e+02   7.220 8.27e-13 ***
Neighborhood_NridgHt    2.322e+04  3.700e+03   6.276 4.56e-10 ***
Lot.Frontage            1.676e+02  3.765e+01   4.451 9.18e-06 ***
Sale.Condition_Partial  2.283e+04  2.968e+03   7.692 2.62e-14 ***
Wood.Deck.SF            1.302e+01  6.033e+00   2.157 0.031150 *
Neighborhood_NoRidge    2.716e+04  5.260e+03   5.163 2.76e-07 ***
Lot.Shape              -3.885e+03  1.310e+03  -2.965 0.003072 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 27420 on 1480 degrees of freedom
Multiple R-squared:  0.8828,    Adjusted R-squared:  0.8812
F-statistic: 586.5 on 19 and 1480 DF,  p-value: < 2.2e-16
```

Table 2. Model 1 - Stepwise regression model

The mathematical formulation of Model 1 is shown below:

$$\begin{aligned} SalePrice = {} & -348.300 + 10.710 * Overall.Qual + 51{,}37 * Gr.Liv.Area + 11.760 * Exter.Qual \\ & + 11.270 * Kitchen.Qual + 12{,}54 * Total.Bsmt.SF + 7{,}36 * X1st.Flr.SF + 19{,}83 \\ & * Garage.Area + 116{,}60 * Year.Built + 15{,}89 * Mas.Vnr.Area + 4.998 * Fireplaces \\ & + 3.675 * Heating.QC + 26{,}55 * BsmtFin.SF.1 + 5.573 * Bsmt.Exposure + 23.220 \\ & * Neighborhood_{NridgHt} + 167{,}60 * Lot.Frontage + 22.830 * Sale.Condition_{Partial} \\ & + 13{,}02 * Wood.Deck.SF + 27.160 * Neighborhood_{NoRidge} - 3.885 * Lot.Shape + \varepsilon \end{aligned}$$

$$\varepsilon \sim N(0, 27.420^2)$$

The expected value of SalePrice for Neighborhood Blmngth (the reference group) with Sale.Condition Abnorml (the reference group) and all the other variables zero is -348.300$ (= -284.569€). However, this interpretation is not sensible. If we try to remove the constant and predict the model again with the rest of the variables, we end up having $R_{adj}^2$ = 88,03% which is lower than the model with constant ($R_{adj}^2$ = 88,12%). After all, we decide to keep the constant in our model as it is statistically significant and $R_{adj}^2$ is better. Regarding the estimations of the variable’s coefficients, if we increase the living area by 1sqft with all the other variables constant and Neighborhood to be

Blmngth with Sale.Condition Abnorml then the SalePrice will increase by 51,37$ (= 41,99€). The expected difference for Neighborhood NoRidge and Neighborhood Blmngth for with Sale.Condition Abnorml and all the other variables constant is 27.160$ (=22.203€).

Based on $R_{adj}^2$ the model explains 88,12% of the variability of the response data around its mean.

Before we end up in Model 1, we need to verify that the residual's assumptions are not violated. **Normality** is the first assumptions we would like to check. The assumption of normality has not to be violated so as to the significance tests of models to be accurate and to get the best estimates of parameters. As the data set is large (n = 1500), Q - Q plot of standardized residuals can be used to visually check the normality assumption. The normal probability plot of residuals should approximately follow a straight line.

Secondly, **homoscedasticity** is also of great importance. Homogeneity of variance occurs when the variance for all observations is equal. When this assumption is satisfied, parameter estimates will be optimal. In the presence of heteroskedasticity, there are two main consequences on the least squares' estimators: the least squares estimator is still a linear and unbiased estimator, but it is no longer best. That is, there is another estimator with a smaller variance. Secondly the standard errors computed for the least squares' estimators are incorrect. This can affect confidence intervals and hypothesis testing that use those standard errors, which could lead to misleading conclusions. Homoscedasticity could be checked with Non-constant Variance Score Test, Levene's Test and residual plot (least squares residuals against SalePrice).

Another assumption is **Linearity**. Linear regression is based on the assumption that the model is linear. Violation of this assumption is very serious meaning that linear model probably does not predict well the actual data. Linearity could be checked by inspecting the Rstudent Residuals vs Fitted plot and Tukey's Test.

Finally, **independence** means that the errors in the model are not related to each other. Computation of standard error relies on the assumption of independence, so if we do not have standard error, confidence intervals and significance tests are biased. Many tests could be done in order to check independence with the most important to be Durbin Watson's Test and Runs Test. In Figure 8 we could observe the appropriate plots for the residual assumptions of Model 1.

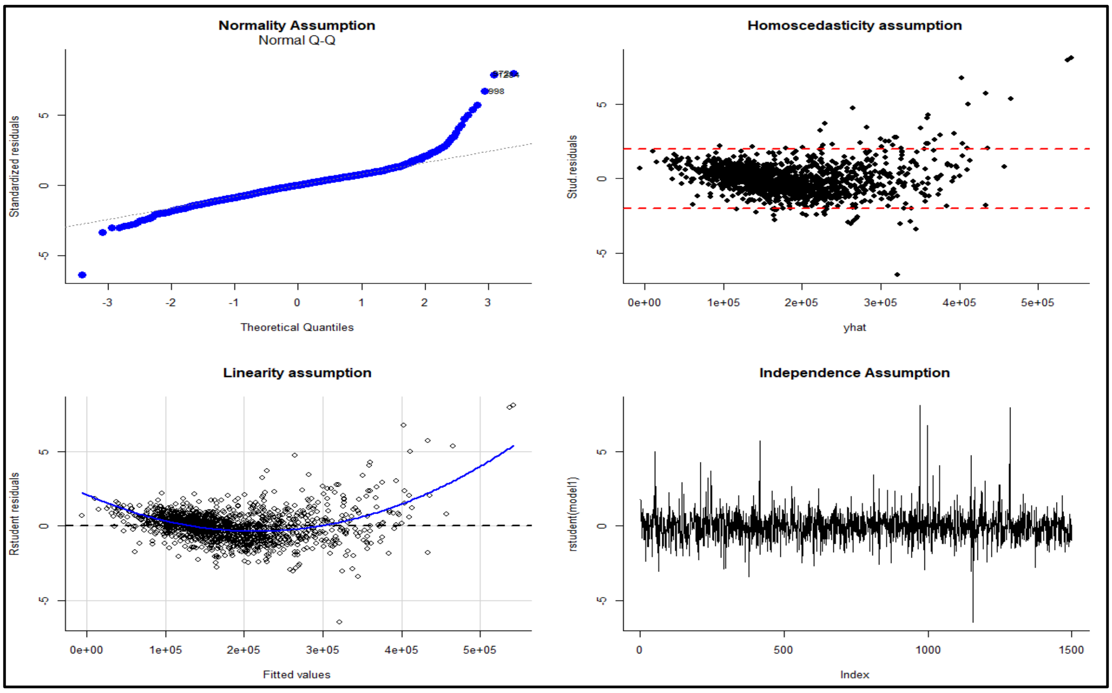


**Figure 8. Model 1 Residuals' Assumptions**

From the Q-Q plot, we could that normality assumption is violated because residuals have at the ends heavily tails and especially to the right. Homoscedasticity of Variance is also rejected as the normally distributed standardized residuals were expected to be centered around zero 0 and reach 2 – 3 standard deviations away from zero (ncvTest/ Levene's Test, reject Ho, p – value < 2.22e-16 < 0.05). Concerning the linearity assumption, it seems that there is a pattern in the residual plot. This suggests that we cannot assume linear relationship (Tukey test, reject Ho, p – value < 2.2e-16 < 0.05). The only assumption that Model 1 meets is the independence (Durbin Watson's Test, do not reject Ho, p – value = 0.114 > 0.05). Multicollinearity was also examined without indicates any problem.

As the assumptions were violated, we continued with logarithmic transformation. We observed earlier that SalePrice is positively skewed so log transformation would be appropriate.

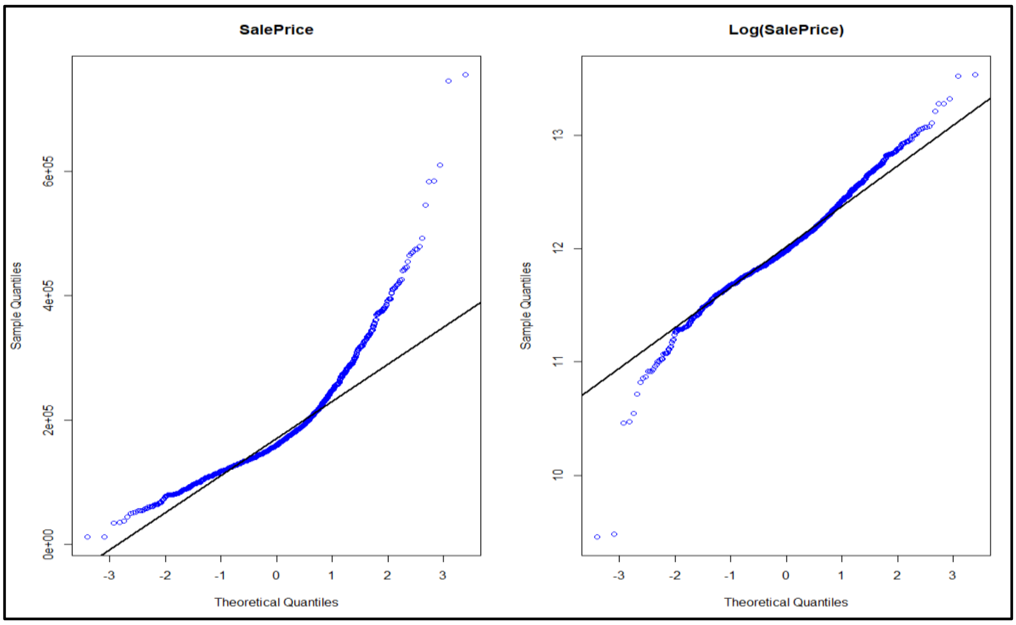
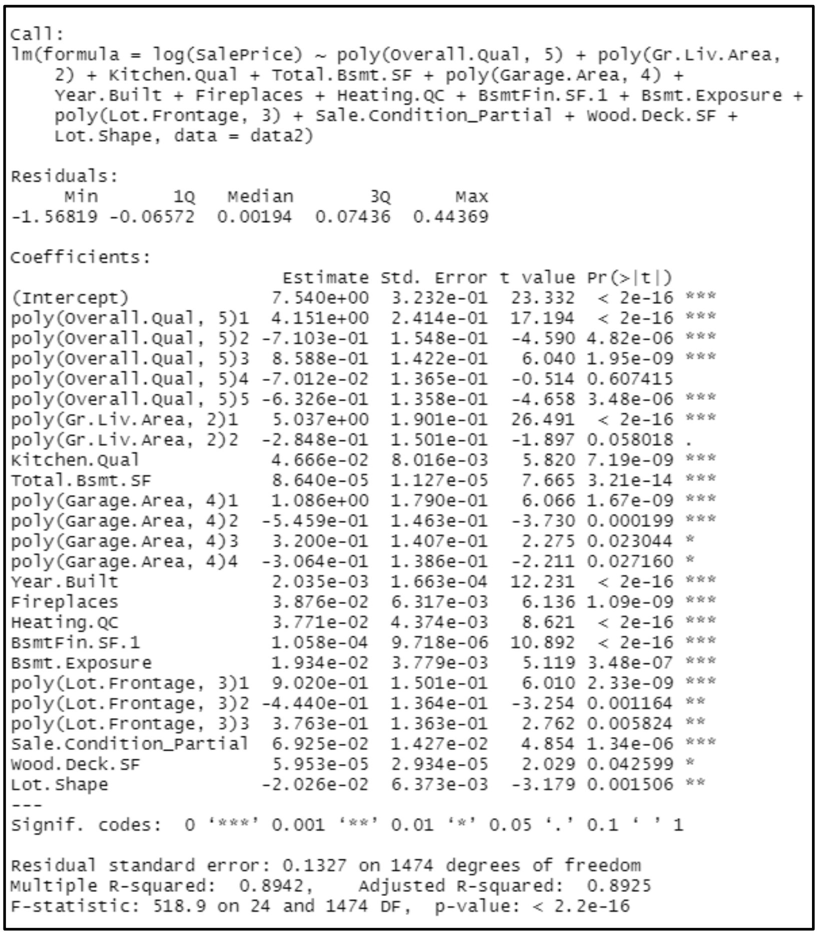


**Figure 9. Log - Transformation of SalePrice**

Based on Figure 9, the Q-Q plot of the log(SalePrice) is more linear than the Q-Q plot of SalePrice as the most points fall approximately along the reference line, signifying that the data is distributed more like a normal distribution than before the transformation.

After defining a new model (**Model2**) with log(SalePrice) at this time as response variable and the same predictors, we had to drop 5 variables that did not appear to be statistically significant (Exter.Qual, X1st.Flr.SF, Mas.Vnr.Area, Neighborhood_NridgHt and Neighborhood_NoRidge). We repeat the same procedure as before with the residual assumptions but still the most of them were violated.

The last thought was to transform our model with **orthogonal polynomials** as to avoid correlated variables. After many repeated trials, we ended up with the polynomial transformations as shown in Table 3. At this point we have to mention that at the first attempt of estimating the model an outlier where observed with cook's distance plot so we had to exclude it. Cook's distance Plot is used to identify influential data points.



```
Call:
lm(formula = log(SalePrice) ~ poly(Overall.Qual, 5) + poly(Gr.Liv.Area,
    2) + Kitchen.Qual + Total.Bsmt.SF + poly(Garage.Area, 4) +
    Year.Built + Fireplaces + Heating.QC + BsmtFin.SF.1 + Bsmt.Exposure +
    poly(Lot.Frontage, 3) + Sale.Condition_Partial + Wood.Deck.SF +
    Lot.Shape, data = data2)

Residuals:
     Min       1Q   Median       3Q      Max
-1.56819 -0.06572  0.00194  0.07436  0.44369

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)             7.540e+00  3.232e-01  23.332  < 2e-16 ***
poly(Overall.Qual, 5)1  4.151e+00  2.414e-01  17.194  < 2e-16 ***
poly(Overall.Qual, 5)2 -7.103e-01  1.548e-01  -4.590 4.82e-06 ***
poly(Overall.Qual, 5)3  8.588e-01  1.422e-01   6.040 1.95e-09 ***
poly(Overall.Qual, 5)4 -7.012e-02  1.365e-01  -0.514 0.607415
poly(Overall.Qual, 5)5 -6.326e-01  1.358e-01  -4.658 3.48e-06 ***
poly(Gr.Liv.Area, 2)1   5.037e+00  1.901e-01  26.491  < 2e-16 ***
poly(Gr.Liv.Area, 2)2  -2.848e-01  1.501e-01  -1.897 0.058018 .
Kitchen.Qual            4.666e-02  8.016e-03   5.820 7.19e-09 ***
Total.Bsmt.SF           8.640e-05  1.127e-05   7.665 3.21e-14 ***
poly(Garage.Area, 4)1   1.086e+00  1.790e-01   6.066 1.67e-09 ***
poly(Garage.Area, 4)2  -5.459e-01  1.463e-01  -3.730 0.000199 ***
poly(Garage.Area, 4)3   3.200e-01  1.407e-01   2.275 0.023044 *
poly(Garage.Area, 4)4  -3.064e-01  1.386e-01  -2.211 0.027160 *
Year.Built              2.035e-03  1.663e-04  12.231  < 2e-16 ***
Fireplaces              3.876e-02  6.317e-03   6.136 1.09e-09 ***
Heating.QC              3.771e-02  4.374e-03   8.621  < 2e-16 ***
BsmtFin.SF.1            1.058e-04  9.718e-06  10.892  < 2e-16 ***
Bsmt.Exposure           1.934e-02  3.779e-03   5.119 3.48e-07 ***
poly(Lot.Frontage, 3)1  9.020e-01  1.501e-01   6.010 2.33e-09 ***
poly(Lot.Frontage, 3)2 -4.440e-01  1.364e-01  -3.254 0.001164 **
poly(Lot.Frontage, 3)3  3.763e-01  1.363e-01   2.762 0.005824 **
Sale.Condition_Partial  6.925e-02  1.427e-02   4.854 1.34e-06 ***
Wood.Deck.SF            5.953e-05  2.934e-05   2.029 0.042599 *
Lot.Shape              -2.026e-02  6.373e-03  -3.179 0.001506 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1327 on 1474 degrees of freedom
Multiple R-squared:  0.8942,	Adjusted R-squared:  0.8925
F-statistic: 518.9 on 24 and 1474 DF,  p-value: < 2.2e-16
```

**Table 3. Model 3 - Log and Polynomial Transformation Model**

The mathematical formulation of **Model3** is shown below:

$$
\begin{aligned}
\log(SalePrice) = {} & 7.54 + 4.15 * Overall.Qual - 0.71 * Overall.Qual^2 + 0.86 * Overall.Qual^3 - 0.07 \\
& * Overall.Qual^4 - 0.63 * \ Overall.Qual^5 + 5.04 * Gr.Liv.Area - 0.28 * Gr.Liv.Area^2 \\
& + 4.67x10^{-2} * Kitchen.Qual + \ 8.64x10^{-5} * Total.Bsmt.SF + \ 1.09 * Garage.Area - 0.55 \\
& * Garage.Area^2 + 0.32 * Garage.Area^3 - 0.31 * Garage.Area^4 + 2.04x10^{-3} * Year.Built \\
& + 3.88x10^{-2} * Fireplaces + 3.77x10^{-2} * Heating.QC + \ 1.06x10^{-4} * BsmtFin.SF.1 \\
& + 1.93x10^{-2} * Bsmt.Exposure + 0.90 * Lot.Frontage - 0.44 * Lot.Frontage^2 + 0.38 \\
& * Lot.Frontage^3 + 6.93x10^{-2} * Sale.Condition_{Partial} + 5.95x10^{-5} * Wood.Deck.SF \\
& - \ 2.03x10^{-2} * Lot.Shape + \varepsilon
\end{aligned}
$$

$$\varepsilon \sim N(0, 0.1327^2)$$

The expected value of SalePrice for Sale.Condition Abnorml (the reference group) and all the other variables zero is $e^{7.54}$ =1.881,83$ (=1.530,56€). Regarding the estimate of the variable's coefficients, the expected difference for Sale.Condition Partial and Sale Condition Abnorml with all the other variables constant increases the value of the house 7% (1,07$) (=0,87€). If we increase the wood deck area by 1sqft with all the other variables constant and Sale.Condition Abnorml then the SalePrice will increase by 1$ (=0,81€). Interpretation of polynomials factors is quite difficult. The average increase in SalePrice for Sale.Condtion Abnormal when Gr.Liv.Area increases from 10 to 11sqft is $e^{5.04*(11-10)-0.28*(11^2-10^2)} = 0.43\$$.

Based on $R^2_{adj}$ the model explains 89.25% of the variability of the response data around its mean.

Now if we would like to find the typical profile of a property sale for Sale.Condtion Abnormal, we have to center all the predictor variables to their mean. The expected price of a house when all covariates are equal to their mean is $e^{12.02} = 166.043\$$. Generally, we need 166.043$ (= 135.236€) to buy a house with Sale.Condtion Abnormal and average characteristics.

After log and polynomial transformation, the residual assumptions are still violated. Only independence is satisfied (Durbin Watson's Test/ Runs Test, do not reject Ho, p – value = 0.576/ 0.5696 > 0.05).

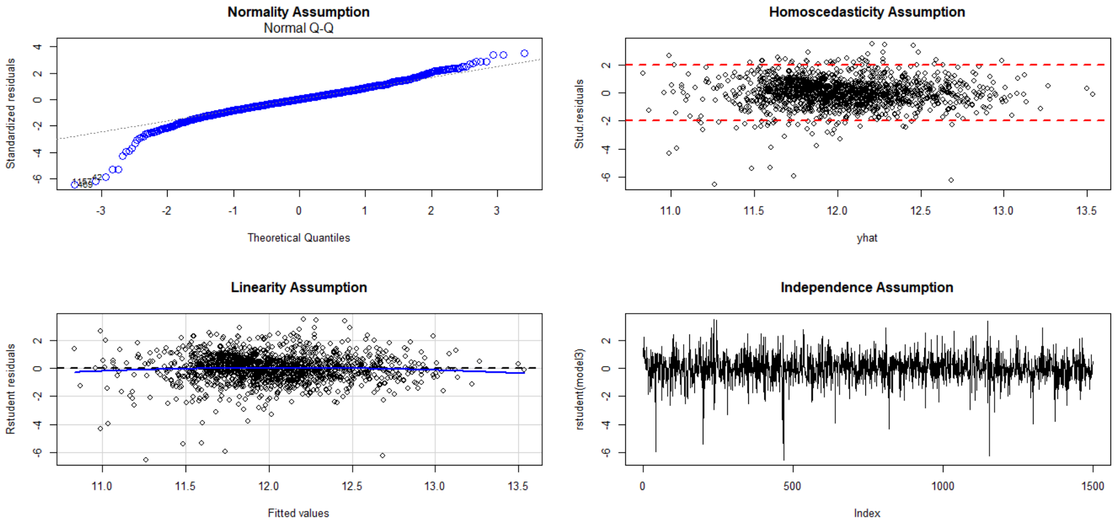


**Figure 10. Model 3 – Residuals' Assumptions**

Although linearity assumption from Figure 10 seems to satisfied as the residual plot has not any pattern, Tukey's Test reject the null Hypothesis (p – value = 0.0003 < 0.05).

## Out-of-sample prediction

A good way to test the assumptions of a model and to realistically compare its predictive performance against other models is to perform out-of-sample validation, which means to withhold some of the sample data from the model training process and then use the model to make predictions for the hold-out data (test data). One metric for comparing out-of-sample performance for multiple models is called root mean squared error (RMSE). In general, the better the model fit, the lower the RMSE. **Leave One Out Cross Validation** is a type of cross-validation approach in which each observation is considered as the validation set and the rest (N-1) observations are considered as the training set. In LOOCV, fitting of the model is done and predicting using one observation validation set. The advantage of the LOOCV method is that we make use all data points reducing potential bias. However, the process is repeated as many times as there are data points, resulting to a higher execution time when n is extremely large. Additionally, we test the model performance against one data point at each iteration. This might result to higher variation in the prediction error, if some data points are outliers. So, we need a good ratio of testing data points, a solution provided by the **k-fold cross-validation** method. The k-fold cross-validation method evaluates the model performance on different subset of the training data and then calculate the average prediction error rate. K-fold cross-validation (CV) is a robust method for estimating the accuracy of a model. The most obvious advantage of k-fold CV compared to LOOCV is computational. A less obvious but potentially more important advantage of k-fold CV is that it often gives more accurate estimates of the test error rate than does LOOCV. One typically performs k-fold cross-validation using k = 10, as this value has been shown empirically to yield test error rate estimates that suffer neither from excessively high bias nor from very high variance.

In our analysis we implemented leave – one – out and 10 – fold cross validation to the three above models (the initial model of stepwise, the log transformed SalePrice and the last model with log(SalePrice) and polynomials) by using our data set. The results are shown in Table 4.

**Table 4. Results of LOO & 10 - Fold CV for Training Dataset**

| | **RMSE** | |
|---|---|---|
| | **LOO** | **10 - Fold** |
| ***Model1*** | 27901.26 | 27283.15 |
| ***Model2*** | 0.1457 | 0.1421 |
| ***Model3*** | 0.1357 | 0.1339 |

Based on RMSE the best model with the minimum RMSE is the Model with log(SalePrice) and polynomials.

## 5. Further Analysis

In this part of our analysis, we would like to use a test dataset to check our models. After importing the dataset in R, we would like to choose only the variables that resulted from stepwise procedure. We noticed that the test dataset has also missing values so we repeat the cleaning procedure as from the initial dataset. We defined the correct types of variables and dummies were also created for categorical variables. Then, LOO and 10 – Fold cross validation were used to select the best model. The results are shown below:

**Table 5. Results of LOO & 10 - Fold CV for Test Dataset**

| | RMSE | |
|---|---|---|
| | **LOO** | **10 - Fold** |
| ***Model1*** | 43684.65 | 41080.06 |
| ***Model2*** | 0.2166 | 0.2047 |
| ***Model3*** | 0.1930 | 0,1876 |

The RMSE of Model 3 is significantly lower comparing to other two so the prediction is much better (RMSE = 0.1876). It is the same model that we end up by using the training data set.

## 6. Conclusion and Discussion

During our analysis we tried to find the best model for predicting the Prices of the properties. The variables that seem to affect the SalePrice of a house are the Overall Quality, ground living area, kitchen quality, total basement area, size of garage, year built, number of fireplaces, heating quality, type 1 finished sq feet, basement exposure, linear feet of street connected to property, wood deck area, sale condition and the general shape of property. However, throughout the estimation of multiple regression models it was not possible to find a model which satisfy the residuals assumptions. This might have been occurred due to the wrong replacement of missing values or the incorrect transformations of the model. Regarding other relevant studies the most important predictors of SalePrice differ as various methods were used with different samples. In the future it would be interesting to return in this case study and handle the data set from another point of view.

## 7. Bibliography

https://en.wikipedia.org/wiki/Multicollinearity#:~:text=Multicollinearity%20refers%20to%20a%20situation,equal%20to%201%20or%20%E2%88%921.

https://cran.r-project.org/web/packages/fastDummies/vignettes/making-dummy-variables.html

https://rpubs.com/cyobero/187387

https://ademos.people.uic.edu/Chapter12.html

https://www.geeksforgeeks.org/loocvleave-one-out-cross-validation-in-r-programming/#:~:text=LOOCV(Leave%20One%20Out%20Cross%2DValidation)%20is%20a%20type,using%20one%20observation%20validation%20set.

https://cran.r-project.org/web/packages/olsrr/vignettes/influence_measures.html#:~:text=Potential%20Residual%20Plot-,Cook's%20D%20Bar%20Plot,R%20Dennis%20Cook%20in%201977.&text=It%20depends%20on%20both%20the,y%20value%20of%20the%20observation

https://www.r-bloggers.com/2015/09/fitting-polynomial-regression-in-r/

## APPENDIX

## R - Code

```
#Reading the data
setwd("C:/Users/Eva/Desktop/aueb/Statistics 1/Main Assignment Ntzoufras")
data<-read.csv(file = 'ames_iowa_housing_29.csv', header= TRUE, sep=";")

head(data)
#--------Drop x order and pid variables
data<-subset(data,select=-c(X, Order, PID))
str(data)

require(psych)
############################################
#--------------------------------------------------------------#
#-------------Find missing values in the data set -------#
#--------------------------------------------------------------#
############################################

NAs <- which(colSums(is.na(data)) > 0)
sort(colSums(sapply(data[NAs], is.na)), decreasing = TRUE)
#Pool.QC is the variable with the most missing values
length(NAs)
#19 variables have NAs
require(plyr)
#-----------------------------------------------------------------------------------
data$Pool.QC[is.na(data$Pool.QC)] <- 'NA'
table(data$Pool.QC)
PoolQC <- c('NA' = 0, 'Po' = 1, 'Fa' = 2, 'TA' = 3, 'Gd' = 4, 'Ex' = 5)
data$Pool.QC<-as.integer(revalue(data$Pool.QC, PoolQC))
table(data$Pool.QC)
#------------------------------------------------------------------------------------
data$Misc.Feature[is.na(data$Misc.Feature)] <- 'NA'
data$Misc.Feature <- as.factor(data$Misc.Feature)
table(data$Misc.Feature)
#-----------------------------------------------------------------------------------
data$Alley[is.na(data$Alley)] <- 'NA'
data$Alley <- as.factor(data$Alley)
table(data$Alley)
#-----------------------------------------------------------------------------------
data$Fence[is.na(data$Fence)] <- 'NA'
table(data$Fence)
data$Fence <- as.factor(data$Fence)
#-----------------------------------------------------------------------------------
data$Fireplace.Qu[is.na(data$Fireplace.Qu)] <- 'NA'
Fireplace <- c('NA' = 0, 'Po' = 1, 'Fa' = 2, 'TA' = 3, 'Gd' = 4, 'Ex' = 5)
data$Fireplace.Qu<-as.integer(revalue(data$Fireplace.Qu, Fireplace))
table(data$Fireplace.Qu)
#-----------------------------------------------------------------------------------
```

```
for (i in 1:nrow(data)){
  if(is.na(data$Lot.Frontage[i]==TRUE)){
    data$Lot.Frontage[i] <-
as.integer(median(data$Lot.Frontage[data$Neighborhood==data$Neighborhood[i]], na.rm=TRUE))
  }}
i<-which(is.na(data$Lot.Frontage))
#there is only 1 GrnHill in our data set so we cannot take the median for this neighborhood
#we are going to use the mode for lot.frontage
require(modeest)
data$Lot.Frontage[is.na(data$Lot.Frontage)] <- mlv(data$Lot.Frontage[-i], method = "mfv")

Lot.Shape<-c('IR3'=0, 'IR2'=1, 'IR1'=2, 'Reg'=3)
data$Lot.Shape<-as.integer(revalue(data$Lot.Shape,Lot.Shape ))
table(data$Lot.Shape)

data$Lot.Config <- as.factor(data$Lot.Config)
table(data$Lot.Config)
#---------------------------------------------------------------------------
data$Garage.Yr.Blt[is.na(data$Garage.Yr.Blt)] <- data$Year.Built[is.na(data$Garage.Yr.Blt)]

length(which(is.na(data$Garage.Type) & is.na(data$Garage.Finish) & is.na(data$Garage.Cond) &
is.na(data$Garage.Qual)))

data$Garage.Type[is.na(data$Garage.Type)] <- 'NA'
data$Garage.Type<- as.factor(data$Garage.Type)
table(data$Garage.Type)

data$Garage.Finish[is.na(data$Garage.Finish)] <- 'NA'
GarageFinish <- c('NA'=0, 'Unf'=1, 'RFn'=2, 'Fin'=3)
data$Garage.Finish<-as.integer(revalue(data$Garage.Finish, GarageFinish))
table(data$Garage.Finish)

data$Garage.Qual[is.na(data$Garage.Qual)] <- 'NA'
GarageQual <- c('NA' = 0, 'Po' = 1, 'Fa' = 2, 'TA' = 3, 'Gd' = 4, 'Ex' = 5)
data$Garage.Qual<-as.integer(revalue(data$Garage.Qual, GarageQual))
table(data$Garage.Qual)

data$Garage.Cond[is.na(data$Garage.Cond)] <- 'NA'
GarageCond <- c('NA' = 0, 'Po' = 1, 'Fa' = 2, 'TA' = 3, 'Gd' = 4, 'Ex' = 5)
data$Garage.Cond<-as.integer(revalue(data$Garage.Cond, GarageCond))
table(data$Garage.Cond)
#----------------------------------------------------------------------------
#check if 47 NAs are the same for all basement variables
length(which(is.na(data$Bsmt.Qual) & is.na(data$Bsmt.Cond) & is.na(data$Bsmt.Exposure) &
             is.na(data$BsmtFin.Type.1) & is.na(data$BsmtFin.Type.2)))

#they are all the same so we have to find which 2 nas are in bsmt.exposure

same<-which(is.na(data$Bsmt.Qual) & is.na(data$Bsmt.Cond) &
            is.na(data$BsmtFin.Type.1) & is.na(data$BsmtFin.Type.2))
bsmtexposure_nas<-which(is.na(data$Bsmt.Exposure))
```

```
#compare to find which are the 2 extra NAs
y<-bsmtexposure_nas[!(bsmtexposure_nas %in% same)]
#rows 174 and 1154

#select only columns for basement with NAs
bmst<-subset(data,select=c(Bsmt.Exposure,Bsmt.Qual, Bsmt.Cond, BsmtFin.Type.1,BsmtFin.Type.2))
#show only the rows with extra 2 NAs
bmst[y,]

#filter the data set
bmst<-
bmst[which((data$Bsmt.Qual=='Gd')&(data$Bsmt.Cond=='TA')&(data$BsmtFin.Type.1=='Unf')&(data$Bs
mtFin.Type.2=='Unf')),]
#find the most common bsmt.exposure
table(bmst$Bsmt.Exposure)
#so we are going to replace NAs with NA

data$Bsmt.Qual[is.na(data$Bsmt.Qual)] <- 'NA'
BsmtQual <- c('NA' = 0, 'Po' = 1, 'Fa' = 2, 'TA' = 3, 'Gd' = 4, 'Ex' = 5)
data$Bsmt.Qual<-as.integer(revalue(data$Bsmt.Qual, BsmtQual))
table(data$Bsmt.Qual)

data$Bsmt.Cond[is.na(data$Bsmt.Cond)] <- 'NA'
BsmtCond <- c('NA' = 0, 'Po' = 1, 'Fa' = 2, 'TA' = 3, 'Gd' = 4, 'Ex' = 5)
data$Bsmt.Cond<-as.integer(revalue(data$Bsmt.Cond, BsmtCond))
table(data$Bsmt.Cond)

data$Bsmt.Exposure[is.na(data$Bsmt.Exposure)] <- 'NA'
BsmtExposure <- c('NA'=0, 'No'=1, 'Mn'=2, 'Av'=3, 'Gd'=4)
data$Bsmt.Exposure<-as.integer(revalue(data$Bsmt.Exposure, BsmtExposure))
table(data$Bsmt.Exposure)

data$BsmtFin.Type.1[is.na(data$BsmtFin.Type.1)] <- 'NA'
BsmtFinType <- c('NA'=0, 'Unf'=1, 'LwQ'=2, 'Rec'=3, 'BLQ'=4, 'ALQ'=5, 'GLQ'=6)
data$BsmtFin.Type.1<-as.integer(revalue(data$BsmtFin.Type.1,BsmtFinType))
table(data$BsmtFin.Type.1)

data$BsmtFin.Type.2[is.na(data$BsmtFin.Type.2)] <- 'NA'
BsmtFinType <- c('NA'=0, 'Unf'=1, 'LwQ'=2, 'Rec'=3, 'BLQ'=4, 'ALQ'=5, 'GLQ'=6)
data$BsmtFin.Type.2<-as.integer(revalue(data$BsmtFin.Type.2, BsmtFinType))
table(data$BsmtFin.Type.2)
#---------------------------------------------------------------------------------------
data$Mas.Vnr.Type[is.na(data$Mas.Vnr.Type)] <- 'Missing'
data$Mas.Vnr.Type<- as.factor(data$Mas.Vnr.Type)
table(data$Mas.Vnr.Type)
data$Mas.Vnr.Area[is.na(data$Mas.Vnr.Area)] <-0
#----------------------------------------------------------------------------------
which(is.na(data$Electrical))
#only one NA so it is better to replace it with the most common category and
#not create a new level missing
```

```
data$Electrical[is.na(data$Electrical)] <- names(sort(table(data$Electrical),decreasing=TRUE))[1]
data$Electrical <- as.factor(data$Electrical)
table(data$Electrical)
#---------------------------------------------------------------------------
which(is.na(data)==TRUE) #we have finished with NAs

#######################################################
#------------------------------------------------------------------------------#
#-------------Convert remaining Variables in the correct type------#
#------------------------------------------------------------------------------#
#######################################################

#-------------------------------------------------------------------------------
data$MS.SubClass <- as.factor(data$MS.SubClass)
#---------------------------------------------------------------------------------
data$MS.Zoning <- as.factor(data$MS.Zoning)
table(data$MS.Zoning)
#--------------------------------------------------------------------------------
data$Street <- as.factor(data$Street)
table(data$Street)
#--------------------------------------------------------------------------------
data$Land.Contour <- as.factor(data$Land.Contour)
table(data$Land.Contour)

landslope<-c('Sev'=0, 'Mod'=1, 'Gtl'=2)
data$Land.Slope<-as.integer(revalue(data$Land.Slope, landslope))
table(data$Land.Slope)
#--------------------------------------------------------------------------------
table(data$Utilities)
#utilities has only 2 nosewr. it seems that this variable does not add any significant
#information in our model. we keep this in mind for later on
#--------------------------------------------------------------------------------
data$Neighborhood <- as.factor(data$Neighborhood)
table(data$Neighborhood)
#--------------------------------------------------------------------------------
data$Condition.1 <- as.factor(data$Condition.1)
table(data$Condition.1)

data$Condition.2 <- as.factor(data$Condition.2)
table(data$Condition.2)
#-------------------------------------------------------------------------------
data$Bldg.Type <- as.factor(data$Bldg.Type)
table(data$Bldg.Type)

data$House.Style<- as.factor(data$House.Style)
table(data$House.Style)
#--------------------------------------------------------------------------------
data$Roof.Style <- as.factor(data$Roof.Style)
table(data$Roof.Style)

data$Roof.Matl <- as.factor(data$Roof.Matl)
```

```
table(data$Roof.Matl)
#------------------------------------------------------------------------
data$Exterior.1st <- as.factor(data$Exterior.1st)
table(data$Exterior.1st)

data$Exterior.2nd <- as.factor(data$Exterior.2nd)
table(data$Exterior.2nd)

ExterQual <- c('Po' = 1, 'Fa' = 2, 'TA' = 3, 'Gd' = 4, 'Ex' = 5)
data$Exter.Qual<-as.integer(revalue(data$Exter.Qual, ExterQual))

ExterCond <- c('Po' = 1, 'Fa' = 2, 'TA' = 3, 'Gd' = 4, 'Ex' = 5)
data$Exter.Cond<-as.integer(revalue(data$Exter.Cond, ExterCond))
#--------------------------------------------------------------------
data$Foundation <- as.factor(data$Foundation)
table(data$Foundation)
#--------------------------------------------------------------------
data$Heating <- as.factor(data$Heating)
table(data$Heating)

HeatingQC <- c('Po' = 1, 'Fa' = 2, 'TA' = 3, 'Gd' = 4, 'Ex' = 5)
data$Heating.QC<-as.integer(revalue(data$Heating.QC, HeatingQC))

data$Central.Air <- as.factor(data$Central.Air)
table(data$Central.Air)
#--------------------------------------------------------------------
kitchenqual <- c('Po' = 1, 'Fa' = 2, 'TA' = 3, 'Gd' = 4, 'Ex' = 5)
data$Kitchen.Qual<-as.integer(revalue(data$Kitchen.Qual, kitchenqual))
table(data$Kitchen.Qual)
#--------------------------------------------------------------------
functional<- c('Sal'=1, 'Sev'=2, 'Maj2'=3, 'Maj1'=4, 'Mod'=5, 'Min2'=6, 'Min1'=7, 'Typ'=8)
data$Functional <- as.integer(revalue(data$Functional,functional))
table(data$Functional)
#--------------------------------------------------------------------
paveddrive<-c('N'=1, 'P'=2, 'Y'=3)
data$Paved.Drive<-as.integer(revalue(data$Paved.Drive, paveddrive))
table(data$Paved.Drive)
#--------------------------------------------------------------------
data$Sale.Type <- as.factor(data$Sale.Type )
table(data$Sale.Type)

data$Sale.Condition <- as.factor(data$Sale.Condition)
table(data$Sale.Condition)

data$Yr.Sold <- as.factor(data$Yr.Sold)
table(data$Yr.Sold)

data$Mo.Sold<- as.factor(data$Mo.Sold)
table(data$Mo.Sold)
#------------------------------------------------------------------------------
#split data to numeric and categorical variables
```

```
index <- (sapply(data, class) == "integer") |(sapply(data, class) == "numeric")
datanum <- data[,index]
datafact <- data[,!index]

ncol(datanum) # 52 numeric variables
ncol(datafact) # 28 categorical variables

####################################################
#------------------------------------------------------------------------#
#-----------------------------univariate analysis -----------------------#
#------------------------------------------------------------------------#
####################################################

#####################################################
#-------------descriptive statistics-------------------------
#####################################################

require(summarytools)
dsc <- round(describe(datanum),2)
View(dsc)

#we found that garage.yr.build has a maximum value 2027. this is not reasonable
#we are going to replace it with the year.remod.add

row<-which((datanum$Garage.Yr.Blt)==2207)
datanum$Garage.Yr.Blt[row]<-datanum$Year.Remod.Add[row]

#--------------Separate Discrete variables------------------------------------
datadiscrete<-subset(datanum,select=c(Garage.Yr.Blt,Bsmt.Full.Bath,Bsmt.Half.Bath,Full.Bath,Half.Bath,
                              Bedroom.AbvGr,Kitchen.AbvGr,Kitchen.Qual,Fireplaces,Fireplace.Qu,
                              Garage.Cars,Year.Built,Year.Remod.Add,Lot.Shape, Land.Slope, Overall.Cond,
Exter.Qual,
                              Exter.Cond,Bsmt.Qual,Bsmt.Cond,Bsmt.Exposure,BsmtFin.Type.1,BsmtFin.Type.2,
                              Heating.QC,TotRms.AbvGrd, Functional, Garage.Finish, Garage.Qual,Garage.Cond,
                              Paved.Drive, Pool.QC,Overall.Qual))

#--------------Separate Continuous variables------------------------------------
datanumeric<-subset(datanum,select=-c(Garage.Yr.Blt,Bsmt.Full.Bath,Bsmt.Half.Bath,Full.Bath,Half.Bath,
                              Bedroom.AbvGr,Kitchen.AbvGr,Kitchen.Qual,Fireplaces,Fireplace.Qu,
                              Garage.Cars,Year.Built,Year.Remod.Add,Lot.Shape, Land.Slope, Overall.Cond,
Exter.Qual,
                              Exter.Cond,Bsmt.Qual,Bsmt.Cond,Bsmt.Exposure,BsmtFin.Type.1,BsmtFin.Type.2,
                              Heating.QC,TotRms.AbvGrd, Functional, Garage.Finish, Garage.Qual,Garage.Cond,
                              Paved.Drive, Pool.QC,Overall.Qual))

#find the mode for categorical variables and mean for numeric
meanvalues<-sapply(datanumeric,mean)
meanvalues<-as.data.frame(round(meanvalues,2))

Mode = function(x){
  tab = table(x)
  maxim = max(tab)
  if (all(tab == maxim))
    mod = NA
  else
    if(is.numeric(x))
      mod = as.numeric(names(tab)[tab == maxim])
  else
    mod = names(tab)[tab == maxim]
  return(mod)
}
modefact<-rep(NA,ncol(datafact))
for (i in 1:ncol(datafact)){
  modefact[i]<-Mode(datafact[,i])
}
modefact<-cbind(colnames(datafact),modefact)

modediscrete<-rep(NA,ncol(datadiscrete))
for (i in 1:ncol(datadiscrete)){
  modediscrete[i]<-Mode(datadiscrete[,i])
}
modediscrete<-cbind(colnames(datadiscrete),modediscrete)

##############################################
#--------------------Visual Analysis-------------------------
##############################################
#useful visuals for report
options(scipen = 999)

par(mfrow=c(1,3))
hist(datanum$SalePrice,main="SalePrice",xlab="",ylab="Frequency",col=5,cex.axis=1.5,cex.lab=1.5)
hist(datanum$Year.Built,main="Year.Built",xlab="",ylab="Frequency",col=5,cex.axis=1.5,cex.lab=1.5)
hist(datanum$Gr.Liv.Area,main="Gr.Liv.Area",xlab="",ylab="Frequency",col=5,cex.axis=1.5,cex.lab=1.5)

par(mfrow=c(1,2))
barplot(table(datafact$Mo.Sold),main="Mo.Sold",xlab="",ylab="Frequency",
col="brown",cex.axis=1.5,cex.lab=1.5)
barplot(sort(table(datafact$Neighborhood)),xlim=c(0,250),horiz=TRUE,las=1,ylab=names(datafact$Neighb
orhood), col="brown",cex.axis=1,cex.names=0.5, main="Neighborhoods of Houses")

#---------------discrete variables-----------
n<-nrow(datadiscrete)
par(mfrow=c(2,3));

for (i in 1:ncol(datadiscrete)) {
  plot(table(datadiscrete[,i])/n, type='h', main=names(datadiscrete)[i], ylab='Relative
frequency',col="deepskyblue4")
}

#-----------------continuous variables

par(mfrow=c(2,3));
```

```
for (i in 1:ncol(datanumeric)) {
  hist(datanumeric[,i], main=names(datanumeric)[i], xlab="",col="deepskyblue4")}
par(mfrow=c(2,3));
for (i in 1:ncol(datanumeric)) {
  qqnorm(datanumeric[,i],main=names(datanumeric)[i],col="magenta")
  qqline(datanumeric[,i], col="black")}

#-----------------categorical variables
#Visual Analysis for factors
par(mfrow=c(2,3));
for (i in 1:ncol(datafact)){
  barplot(table(datafact[,i]),ylab=names(datafact)[i], col="brown",cex.axis=1.5,cex.lab=1.5)
}
###################################################
#-------------------------------------------------------------------------#
#-----------------------------Bivariate analysis -----------------------#
#-------------------------------------------------------------------------#
###################################################

#---------------------Pairwise comparisons---------------------------------
#numeric variables
require(corrplot)
head(datanumeric)
par(mfrow=c(2,3))
for (i in 1:(ncol(datanumeric)-1)){
  plot(datanumeric[,i],datanum$SalePrice,ylab="SalePrice",xlab=names(datanumeric)[i],col="magenta")}

n<-which(colnames(datanum)=='SalePrice')
x<-cor(datanum$SalePrice, datanum[,-n])
x <- x[,order(x[1,])]
barplot(x, horiz=T, las = 1, xlim=c(-0.4,0.8), cex.axis=1, cex.names=0.5, col="mediumorchid1",)
#overallqual has the strongest correlation with saleprice

cor_numVar <- cor(datanum, use="pairwise.complete.obs") #correlations of all numeric variables
#sort on decreasing correlations with SalePrice
cor_sorted <- as.matrix(sort(cor_numVar[,'SalePrice'], decreasing = TRUE))
#select only high corelations
CorHigh <- names(which(apply(cor_sorted, 1, function(x) abs(x)>0.53)))
cor_numVar <- cor_numVar[CorHigh, CorHigh]
corrplot.mixed(cor_numVar, tl.col="black", tl.pos = "lt")
length(CorHigh)

#drop variables with cor>0.80
#drop garage.cars, Garage.Yr.Blt, Garage.Cond, Pool.Area, BsmtFin.SF.2

datanum<-subset(datanum,select=-c(Garage.Yr.Blt, Garage.Cars,
                                  Garage.Cond,Pool.Area,
                                  BsmtFin.SF.2))
```

```
#--------------------Saleprice~categorical variables-------------------------------
#SalePrice (our response) on factor variables
par(mfrow=c(2,3))
for(i in 1:ncol(datafact)){
  boxplot(datanum$SalePrice~datafact[,i], xlab=names(datafact)[i], ylab='SalePrice',cex.lab=1.5, col=4)}
#--------------------------------------------------------------------------------
#Saleprice - Neighborhood
options(scipen = 0)
#Tests for k>2 independent samples - 1 quantitative & 1 categorical factor
anova<-aov(data$SalePrice~data$Neighborhood)
summary(anova)
#Can we assume normality?
require(nortest)
shapiro.test(anova$res)
lillie.test(anova$res)
qqnorm(anova$res)
qqline(anova$res)
#no
#large sample?
#yes
#is the mean a sufficient descriptive measure for central location for all groups?
require(lawstat)
#symmetry.test(anova$res)
#no
#test for equality of medians (kruskall-wallis test)
kruskal.test(data$SalePrice~data$Neighborhood)
#reject Ho
boxplot(data$SalePrice~data$Neighborhood,ylab="SalePrice",xlab="",main="Neighborhood",boxcol=1,boxfill=2,las=2,medlwd=3,medcol="black")
#-------------------------------------------------------------------------------
#Saleprice - YrSold
#Tests for k>2 independent samples - 1 quantitative & 1 categorical factor
anova<-aov(data$SalePrice~data$Yr.Sold)
summary(anova)
#Can we assume normality?
require(nortest)
shapiro.test(anova$res)
lillie.test(anova$res)
qqnorm(anova$res)
qqline(anova$res)
#no
#large sample?
#yes
#is the mean a sufficient descriptive measure for central location for all groups?
require(lawstat)
#symmetry.test(anova$res)
#nO
#test for equality of medians (kruskall-wallis test)
kruskal.test(data$SalePrice~data$Yr.Sold)
#we do not reject Ho
```

```
boxplot(data$SalePrice~data$Yr.Sold, ylab="SalePrice",xlab="",main="Yr.Sold",boxcol=1,boxfill=2,medlwd=3,medcol="black")
#-----------------------------------------------------------------------------------
#Sales - Street
#test for 2 independent samples - 1 quantitative & 1 binary
#can we assume the normality?
table(data$Street)
by(data$SalePrice,data$Street,shapiro.test)
by(data$SalePrice,data$Street, lillie.test)
#no
#large samples?
#yes
#is the mean a sufficient descriptive measure for central location for both groups?
#symmetry.test(data$SalePrice)
#no
#test for zero difference between the medians
wilcox.test(data$SalePrice~data$Street)
#we reject Ho: M1=M2 significance difference is found about the median of the saleprice between grvl and pave
boxplot(data$SalePrice~data$Street,ylab="SalePrice",xlab="",main="Street",boxfill=2,medlwd=3)
#-----------------------------------------------------------------------------------
#Sales -Central Air
#test for 2 independent samples - 1 quantitative & 1 binary
#can we assume the normality?
table(data$Central.Air)
by(data$SalePrice,data$Central.Air,shapiro.test)
by(data$SalePrice,data$Central.Air, lillie.test)
#no
#large samples?
#yes
#is the mean a sufficient descriptive measure for central location for both groups?
#symmetry.test(data$SalePrice)
#no
#test for zero difference between the medians
wilcox.test(data$SalePrice~data$Central.Air)
#we reject Ho: M1=M2 significance difference is found about the median of the saleprice between grvl and pave
boxplot(data$SalePrice~data$Central.Air,ylab="SalePrice",xlab="",main="Central.Air",boxfill=2,medlwd=3)
par(mfrow=c(2,2))

#####################################################
#-----------------------------------------------------------------------#
#-------------------------------Model Selection ----------------------#
#-----------------------------------------------------------------------#
#####################################################
table(data$sa)
require(fastDummies)
datafact <- dummy_cols(datafact, remove_selected_columns = TRUE,remove_first_dummy = TRUE)
colnames(datafact)
sapply(datafact,as.numeric)
```

```
data<-cbind(datanum,datafact)
ncol(data)
#after dummy creation variables are 235
table(data$ne)
# Dropping highly correlated variables
cor_numVar <- cor(data, use="pairwise.complete.obs") #correlations of all numeric variables
#sort on decreasing correlations with SalePrice
cor_sorted <- as.matrix(sort(cor_numVar[,'SalePrice'], decreasing = TRUE))
#select only high corelations
CorHigh <- names(which(apply(cor_sorted, 1, function(x) abs(x)>0.25)))
cor_numVar <- cor_numVar[CorHigh, CorHigh]
corrplot.mixed(cor_numVar, tl.col="black", tl.pos = "lt")
length(CorHigh)

data<-subset(data,select=c(SalePrice, Overall.Qual, Gr.Liv.Area, Exter.Qual,Kitchen.Qual,
                           Total.Bsmt.SF, X1st.Flr.SF,
                           Garage.Area, Bsmt.Qual,Year.Built, Garage.Finish,
                           Year.Remod.Add, Full.Bath,
                           Fireplace.Qu, Foundation_PConc, Mas.Vnr.Area,
                           TotRms.AbvGrd, Fireplaces, Heating.QC,
                           BsmtFin.SF.1, Bsmt.Exposure, Neighborhood_NridgHt,
                           Lot.Frontage, MS.SubClass_60, Garage.Type_Attchd,
                           Sale.Type_New, Sale.Condition_Partial, Exterior.1st_VinylSd,
                           Exterior.2nd_VinylSd, Wood.Deck.SF, BsmtFin.Type.1,
                           Neighborhood_NoRidge, Mas.Vnr.Type_Stone, Open.Porch.SF,
                           Paved.Drive, Central.Air_Y, Garage.Qual,
                           X2nd.Flr.SF, Half.Bath, Roof.Style_Hip,
                           Bsmt.Full.Bath, MS.Zoning_RM,
                           Lot.Shape, Foundation_CBlock, Garage.Type_Detchd,
                           Mas.Vnr.Type_None ))
ncol(data)


#we end up with 46 variables

########################
#-----------------------------------#
#-------------LASSO-------------#
#-----------------------------------#
########################

mfull <- lm(data$SalePrice ~ . ,data)
n<-which(colnames(data)=="SalePrice")

require(glmnet)
X <- model.matrix(mfull)[,-n]

lasso <- glmnet(X, data$SalePrice)
plot(lasso, xvar = "lambda", label = T)

#Use cross validation to find a reasonable value for lambda - 10-fold CV
```

```
lasso1 <- cv.glmnet(X, data$SalePrice, alpha = 1)
lasso1$lambda
lasso1$lambda.min
lasso1$lambda.1se
plot(lasso1, cex.lab = 2, cex.axis = 2)

coef(lasso1, s = "lambda.min")
coef(lasso1, s = "lambda.1se") #keep lasso with 64 variables

coefs <- as.matrix(coef(lasso1)) # convert to a matrix (618 by 1)
ix <- which(abs(coefs[,1]) > 0)
length(ix)
coefs<-coefs[ix,1, drop=FALSE]
coefs
plot(lasso1$glmnet.fit, xvar = "lambda")
abline(v=log(c(lasso1$lambda.min, lasso1$lambda.1se)), lty =2)

#-----model with 25 variables of lasso---------
model<-lm(SalePrice~Overall.Qual+Gr.Liv.Area+Exter.Qual+Kitchen.Qual+
          Total.Bsmt.SF+X1st.Flr.SF+Garage.Area+Bsmt.Qual+
          Year.Built+Year.Remod.Add+Fireplace.Qu+Foundation_PConc+
          Mas.Vnr.Area+Fireplaces+Heating.QC+BsmtFin.SF.1+Bsmt.Exposure+
          Neighborhood_NridgHt+Lot.Frontage+Sale.Type_New+Sale.Condition_Partial+
          Wood.Deck.SF+Neighborhood_NoRidge+Lot.Shape,data)


#########################
#-----------------------------------##
#-------------STEPWISE--------##
#-----------------------------------##
#########################
step<-step(model, direction='both') #Stepwise

#after stepwise model has left with 19 variables

###create scatter plots to have a better insight
par(mfrow=c(2,2))
for (i in 2:ncol(data)){
  m<-lm(data$SalePrice~data[,i])
  plot(data[,i],data$SalePrice,xlab=colnames(data[i]))
  abline(m, col="blue")}

#---------------------------------------------------------------------#
#                 Model1 - Initial variables                    #
#---------------------------------------------------------------------#
model1<-lm(SalePrice ~ Overall.Qual + Gr.Liv.Area + Exter.Qual + Kitchen.Qual +
           Total.Bsmt.SF + X1st.Flr.SF + Garage.Area + Year.Built +
           Foundation_PConc + Mas.Vnr.Area + Fireplaces + Heating.QC +
           BsmtFin.SF.1 + Bsmt.Exposure + Neighborhood_NridgHt + Lot.Frontage +
           Sale.Condition_Partial + Wood.Deck.SF + Neighborhood_NoRidge +
           Lot.Shape, data=data)
```

```
summary(model1)
#------foundation.pcon not significant
model1<-lm(SalePrice ~ Overall.Qual + Gr.Liv.Area + Exter.Qual + Kitchen.Qual +
            Total.Bsmt.SF + X1st.Flr.SF + Garage.Area + Year.Built +
            Mas.Vnr.Area + Fireplaces + Heating.QC +
            BsmtFin.SF.1 + Bsmt.Exposure + Neighborhood_NridgHt + Lot.Frontage +
            Sale.Condition_Partial + Wood.Deck.SF + Neighborhood_NoRidge +
            Lot.Shape, data=data)
summary(model1)

#exlude constant
model1<-lm(SalePrice ~ -1 + Overall.Qual + Gr.Liv.Area + Exter.Qual + Kitchen.Qual +
            Total.Bsmt.SF + X1st.Flr.SF + Garage.Area + Year.Built +
            Mas.Vnr.Area + Fireplaces + Heating.QC +
            BsmtFin.SF.1 + Bsmt.Exposure + Neighborhood_NridgHt + Lot.Frontage +
            Sale.Condition_Partial + Wood.Deck.SF + Neighborhood_NoRidge +
            Lot.Shape, data=data)

summary(model1)

true.r2 <- 1-sum(model1$res^2)/((1500-1)*var(data$SalePrice))
true.r2
#radj lower so keep the constant

#after stepwise the model left with 19 variables

##########################################
###### Checking the assumptions of model1#####
##########################################
par(mfrow=c(2,2),pch=16,bty='l')
# -----------------------------------------------
# normality
# -----------------------------------------------
plot(model1,which=2,cex=1.5,col='blue',main="Normality Assumption")
#normality is rejected
# -----------------------------------------------
# Homoscedasticity
# -----------------------------------------------
Stud.residuals <- rstudent(model1)
yhat <- fitted(model1)
plot(yhat, Stud.residuals, main="Homoscedasticity assumption")
abline(h=c(-2,2),lty=2,col='red',lwd=2)

plot(yhat, Stud.residuals^2)
abline(h=4, col=2, lty=2,lwd=2)
# ------------------
library(car)
ncvTest(model1)
# ------------------
yhat.quantiles<-cut(yhat, breaks=quantile(yhat, probs=seq(0,1,0.25)), dig.lab=6)
table(yhat.quantiles)
```

```
leveneTest(rstudent(model1)~yhat.quantiles)
boxplot(rstudent(model1)~yhat.quantiles)
# ---------------------------------------------
# non linearity
# ---------------------------------------------
library(car)
plot(model1,1)
residualPlot(model1, type='rstudent', main="Linearity assumption")
residualPlots(model1, plot=F, type = "rstudent")
# -------------------
# Independence
# -------------------
plot(rstudent(model1), type='l',main="Independence Assumption")
library(randtests); runs.test(model1$res)
library(car); durbinWatsonTest(model1)
# -------------------
# multicollinearity
# -------------------
library(car)
vif(model1)
alias(model1)
#no collinearity problems
#-------------------
#cook's distance
plot(model1,pch=16,cex=2,col='blue',which=4)
abline(h=4/5,col='red',lty=2,lwd=2)
#-------------------------------------------------


###########################################
#                                         #
#            evaluate model1              #
#                                         #
###########################################
require(caret)
#################################################
#--------Leave one out cross validation - LOOCV----------
#################################################
set.seed(2822026)
# Define training control
train.control <- trainControl(method = "LOOCV")
# Train the model
model1a <- train(SalePrice ~ Overall.Qual + Gr.Liv.Area + Exter.Qual + Kitchen.Qual +
              Total.Bsmt.SF + X1st.Flr.SF + Garage.Area + Year.Built +
              Mas.Vnr.Area + Fireplaces + Heating.QC +
              BsmtFin.SF.1 + Bsmt.Exposure + Neighborhood_NridgHt + Lot.Frontage +
              Sale.Condition_Partial + Wood.Deck.SF + Neighborhood_NoRidge +
              Lot.Shape, data=data, method = "lm", trControl = train.control)
# Summarize the results
print(model1a)
#RMSE 27901.26
```

```
##########################################
#----------------10-fold cross-validation-----------------
##########################################
set.seed(2822026)
# Define training control
train.control <- trainControl(method = "cv", number = 10)
# Train the model
model1b <- train(SalePrice ~ Overall.Qual + Gr.Liv.Area + Exter.Qual + Kitchen.Qual +
                 Total.Bsmt.SF + X1st.Flr.SF + Garage.Area + Year.Built +
                 Mas.Vnr.Area + Fireplaces + Heating.QC +
                 BsmtFin.SF.1 + Bsmt.Exposure + Neighborhood_NridgHt + Lot.Frontage +
                 Sale.Condition_Partial + Wood.Deck.SF + Neighborhood_NoRidge +
                 Lot.Shape, data=data, method = "lm",trControl = train.control)
# Summarize the results
print(model1b)
#RMSE 27283.15

#------------------------------------------------------------#
#                 Model2 - Log(SalePrice)                    #
#------------------------------------------------------------#
par(mfrow=c(1,2))
qqnorm(data$SalePrice, col="blue",main="SalePrice")
qqline(data$SalePrice, lwd = 2)
qqnorm(log(data$SalePrice), col="blue",main="Log(SalePrice)")
qqline(log(data$SalePrice), lwd = 2)

model2<-lm(log(SalePrice) ~ Overall.Qual + Gr.Liv.Area + Exter.Qual + Kitchen.Qual +
           Total.Bsmt.SF + X1st.Flr.SF + Garage.Area + Year.Built +
           Mas.Vnr.Area + Fireplaces + Heating.QC +
           BsmtFin.SF.1 + Bsmt.Exposure + Neighborhood_NridgHt + Lot.Frontage +
           Sale.Condition_Partial + Wood.Deck.SF + Neighborhood_NoRidge +
           Lot.Shape, data=data)
summary(model2)

#drop exter.qual, X1st.Flr.SF, mas.vnr.area,Neighborhood_NridgHt,Neighborhood_NoRidge

model2<-lm(log(SalePrice) ~ Overall.Qual + Gr.Liv.Area + Kitchen.Qual +
           Total.Bsmt.SF +  Garage.Area + Year.Built +
            Fireplaces + Heating.QC +
           BsmtFin.SF.1 + Bsmt.Exposure + Lot.Frontage +
           Sale.Condition_Partial + Wood.Deck.SF +
           Lot.Shape, data=data)
summary(model2)

##############################################
###### Checking the assumptions ###################
##############################################
# normality
plot(model2,which=2,cex=1.5,col='blue')
#normality is rejected
# ---------------------------------------------
# Homoscedasticity
# ----------------------------------------------
Stud.residuals <- rstudent(model2)
yhat <- fitted(model2)
par(mfrow=c(1,2),pch=16,bty='l')
plot(yhat, Stud.residuals)
abline(h=c(-2,2),lty=2,col='red',lwd=2)
plot(yhat, Stud.residuals^2)
abline(h=4, col=2, lty=2,lwd=2)

# ------------------
library(car)
ncvTest(model2)
# ------------------
yhat.quantiles<-cut(yhat, breaks=quantile(yhat, probs=seq(0,1,0.25)), dig.lab=6)
table(yhat.quantiles)
leveneTest(rstudent(model2)~yhat.quantiles)
boxplot(rstudent(model2)~yhat.quantiles)
# ---------------------------------------------
# non linearity
# ---------------------------------------------
library(car)
plot(model2,1)
residualPlot(model2, type='rstudent')
residualPlots(model2, plot=F, type = "rstudent")
# -------------------
# Independence
# -------------------
plot(rstudent(model2), type='l')
library(randtests); runs.test(model2$res)
library(car); durbinWatsonTest(model2)
# -------------------
# multicollinearity
# -------------------
library(car)
vif(model2)
alias(model2)
#no collinearity problems
#-------------------
#cook's distance
plot(model2,pch=16,cex=2,col='blue',which=4)
abline(h=4/5,col='red',lty=2,lwd=2)
#-----------------------------------------------
#################################################
#--------Leave one out cross validation - LOOCV-----------
#################################################
set.seed(2822026)
# Define training control
train.control <- trainControl(method = "LOOCV")
# Train the model
model2a <- train(log(SalePrice) ~ Overall.Qual + Gr.Liv.Area + Kitchen.Qual +
                Total.Bsmt.SF +  Garage.Area + Year.Built +
                Fireplaces + Heating.QC +
                BsmtFin.SF.1 + Bsmt.Exposure + Lot.Frontage +
                Sale.Condition_Partial + Wood.Deck.SF +
                Lot.Shape, data=data, method = "lm", trControl = train.control)
# Summarize the results
print(model2a)
#RMSE 0.1457159

###########################################
#----------------10-fold cross-validation-----------------
###########################################
set.seed(2822026)
# Define training control
train.control <- trainControl(method = "cv", number = 10)
# Train the model
model2b <- train(log(SalePrice) ~ Overall.Qual + Gr.Liv.Area + Kitchen.Qual +
                Total.Bsmt.SF +  Garage.Area + Year.Built +
                Fireplaces + Heating.QC +
                BsmtFin.SF.1 + Bsmt.Exposure + Lot.Frontage +
                Sale.Condition_Partial + Wood.Deck.SF +
                Lot.Shape, data=data, method = "lm",trControl = train.control)
# Summarize the results
print(model2b)
#RMSE 0.1420607
#-----------------------------------------------------#
#                       Model3                        #
#---------------------------------------------------- #
########################################
##############Log and Polynomials########
########################################
data2<-data[-c(787),]
model3<-lm(log(SalePrice) ~ poly(Overall.Qual,5)+
           poly(Gr.Liv.Area,2)+ Kitchen.Qual + Total.Bsmt.SF +
           poly(Garage.Area,4)+
           Year.Built + Fireplaces + Heating.QC +
           BsmtFin.SF.1 + Bsmt.Exposure  +
           poly(Lot.Frontage,3) +
           Sale.Condition_Partial + Wood.Deck.SF  +
           Lot.Shape, data=data2)
summary(model3)
# model with centered covariates
#----------------------------------------------------------------------------
data2centered <- as.data.frame(scale(data2, center = TRUE, scale = F))
data2centered$SalePrice<-data2$SalePrice
model3centered<-lm(log(SalePrice) ~ poly(Overall.Qual,5)+
           poly(Gr.Liv.Area,2)+ Kitchen.Qual + Total.Bsmt.SF +
           poly(Garage.Area,4)+
           Year.Built + Fireplaces + Heating.QC +
           BsmtFin.SF.1 + Bsmt.Exposure  +
           poly(Lot.Frontage,3) +
            Sale.Condition_Partial + Wood.Deck.SF  +
            Lot.Shape, data=data2centered)
summary(model3centered)
######################################
###### Checking the assumptions ##########
######################################
par(mfrow=c(2,2))
# -----------------------------------------------
# normality
# -----------------------------------------------
plot(model3,which=2,cex=1.5,col='blue', main="Normality Assumption")
#normality is rejected
# -----------------------------------------------
# Homoscedasticity
# -----------------------------------------------
Stud.residuals <- rstudent(model3)
yhat <- fitted(model3)
par(mfrow=c(1,2),pch=16,bty='l')
plot(yhat, Stud.residuals,main="Homoscedasticity Assumption")
abline(h=c(-2,2),lty=2,col='red',lwd=2)
plot(yhat, Stud.residuals^2)
abline(h=4, col=2, lty=2,lwd=2)
# ------------------
library(car)
ncvTest(model3)
# ------------------
yhat.quantiles<-cut(yhat, breaks=quantile(yhat, probs=seq(0,1,0.25)), dig.lab=6)
table(yhat.quantiles)
leveneTest(rstudent(model3)~yhat.quantiles)
boxplot(rstudent(model3)~yhat.quantiles)
# ----------------------------------------------
# non linearity
# ----------------------------------------------
library(car)
residualPlot(model3, type='rstudent', main="Linearity Assumption")
residualPlots(model3, plot=F, type = "rstudent")
# -------------------
# Independence
# -------------------
plot(rstudent(model3), type='l',main="Independence Assumption")
library(randtests); runs.test(model3$res)
library(car); durbinWatsonTest(model3)
# -------------------
# multicollinearity
# -------------------
library(car)
vif(model3)
alias(model3)
#no collinearity problems
#-------------------
#cook's distance
```

```
plot(model3,pch=16,cex=2,col='blue',which=4)
abline(h=4/5,col='red',lty=2,lwd=2)
#---------------------------------------------------
################################################
#--------Leave one out cross validation - LOOCV-----------
################################################
set.seed(2822026)
# Define training control
train.control <- trainControl(method = "LOOCV")
# Train the model
model3a <- train(log(SalePrice) ~ poly(Overall.Qual,5)+
              poly(Gr.Liv.Area,2)+ Kitchen.Qual + Total.Bsmt.SF +
              poly(Garage.Area,4)+
              Year.Built + Fireplaces + Heating.QC +
              BsmtFin.SF.1 + Bsmt.Exposure  +
              poly(Lot.Frontage,3) +
              Sale.Condition_Partial + Wood.Deck.SF  +
              Lot.Shape, data=data2,
              method = "lm", trControl = train.control)
# Summarize the results
print(model3a)
#RMSE 0.1356781
############################################
#----------------10-fold cross-validation-----------------
############################################
set.seed(2822026)
# Define training control
train.control <- trainControl(method = "cv", number = 10)
# Train the model
model3b <- train(log(SalePrice) ~ poly(Overall.Qual,5)+
              poly(Gr.Liv.Area,2)+ Kitchen.Qual + Total.Bsmt.SF +
              poly(Garage.Area,4)+
              Year.Built + Fireplaces + Heating.QC +
              BsmtFin.SF.1 + Bsmt.Exposure  +
              poly(Lot.Frontage,3) +
              Sale.Condition_Partial + Wood.Deck.SF  +
              Lot.Shape, data=data2,
              method = "lm",trControl = train.control)
# Summarize the results
print(model3b)
#RMSE 0.1339403

##########################################################################
#########################TEST DATA FROM FILE#########################
##########################################################################
setwd("C:/Users/Administrator/Desktop/documents/Statistics 1/Main Assignment Ntzoufras")
testdata<-read.csv(file = "ames_iowa_housing_test.csv", header= TRUE, sep=";")
#keep only the remaining variables after lasso
testdata<-subset(testdata,select=c(SalePrice, Overall.Qual , Gr.Liv.Area , Exter.Qual , Kitchen.Qual ,
                         Total.Bsmt.SF , X1st.Flr.SF , Garage.Area , Year.Built ,
                         Foundation, Mas.Vnr.Area , Fireplaces , Heating.QC ,
                                  BsmtFin.SF.1 , Bsmt.Exposure , Neighborhood , Lot.Frontage ,
                                  Sale.Condition , Wood.Deck.SF,
                                  Lot.Shape))
#------------------------------------------------------------------------------
#-----find missing values and define the correct type as train data
#------------------------------------------------------------------------------
NAs <- which(colSums(is.na(testdata)) > 0)
sort(colSums(sapply(testdata[NAs], is.na)), decreasing = TRUE)
length(NAs)
#6 variables have NAs
#------------------------------------------------------------------------------
testdata$Bsmt.Exposure[is.na(testdata$Bsmt.Exposure)] <- 'NA'
BsmtExposure <- c('NA'=0, 'No'=1, 'Mn'=2, 'Av'=3, 'Gd'=4)
testdata$Bsmt.Exposure<-as.integer(revalue(testdata$Bsmt.Exposure, BsmtExposure))
table(testdata$Bsmt.Exposure)
#------------------------------------------------------------------------------
for (i in 1:nrow(testdata)){
  if(is.na(testdata$Lot.Frontage[i]==TRUE)){
    testdata$Lot.Frontage[i] <-
as.integer(median(testdata$Lot.Frontage[testdata$Neighborhood==testdata$Neighborhood[i]],
na.rm=TRUE)) }}
#------------------------------------------------------------------------------
testdata$Mas.Vnr.Area[is.na(testdata$Mas.Vnr.Area)] <-0
#------------------------------------------------------------------------------
testdata$Garage.Area[is.na(testdata$Garage.Area)]<-0
i<-which(is.na(testdata$Lot.Frontage))
require(modeest)
testdata$Lot.Frontage[is.na(testdata$Lot.Frontage)] <- mlv(testdata$Lot.Frontage[-i], method = "mfv")
#------------------------------------------------------------------------------
which(is.na(testdata))
#------------------------------------------------------------------------------
testdata$Sale.Condition <- as.factor(testdata$Sale.Condition)
table(testdata$Sale.Condition)
#------------------------------------------------------------------------------
ExterQual <- c('Po' = 1, 'Fa' = 2, 'TA' = 3, 'Gd' = 4, 'Ex' = 5)
testdata$Exter.Qual<-as.integer(revalue(testdata$Exter.Qual, ExterQual))
#------------------------------------------------------------------------------
kitchenqual <- c('Po' = 1, 'Fa' = 2, 'TA' = 3, 'Gd' = 4, 'Ex' = 5)
testdata$Kitchen.Qual<-as.integer(revalue(testdata$Kitchen.Qual, kitchenqual))
table(testdata$Kitchen.Qual)
#------------------------------------------------------------------------------
HeatingQC <- c('Po' = 1, 'Fa' = 2, 'TA' = 3, 'Gd' = 4, 'Ex' = 5)
testdata$Heating.QC<-as.integer(revalue(testdata$Heating.QC, HeatingQC))
#------------------------------------------------------------------------------
Lot.Shape<-c('IR3'=0, 'IR2'=1, 'IR1'=2, 'Reg'=3)
testdata$Lot.Shape<-as.integer(revalue(testdata$Lot.Shape,Lot.Shape ))
table(testdata$Lot.Shape)
#------------------------------------------------------------------------------
indextestdata <- (sapply(testdata, class) == "integer") |(sapply(testdata, class) == "numeric")
datanumtestdata <- testdata[,indextestdata]
datafacttestdata <- testdata[,!indextestdata]
```

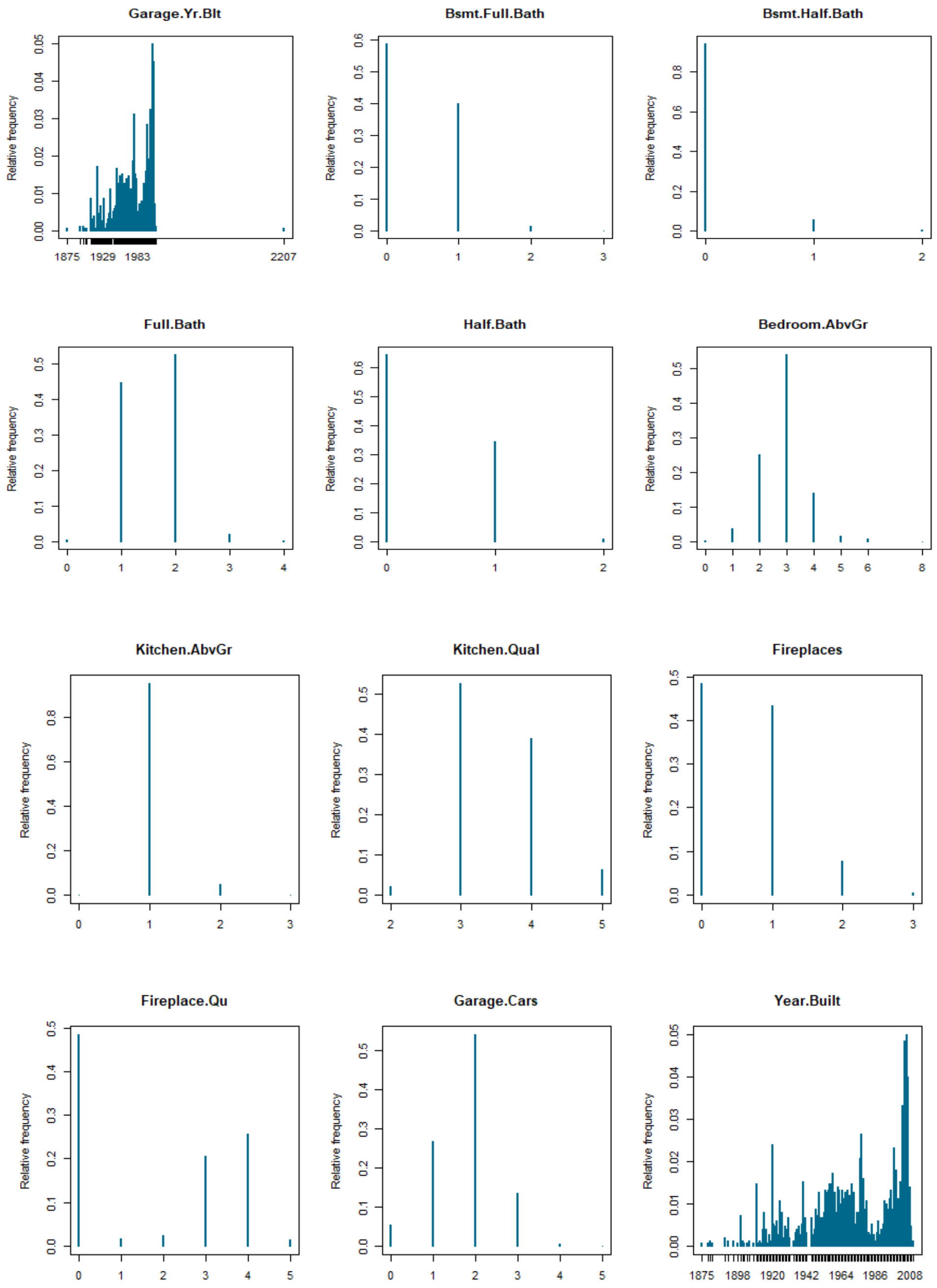
```
#------------------------------------------------------------------------------
#------------------Create Dummies-------------------------------------
#------------------------------------------------------------------------------
require(fastDummies)
datafacttestdata <- dummy_cols(datafacttestdata, remove_selected_columns = TRUE,remove_first_dummy
= TRUE)
colnames(datafacttestdata)
sapply(datafacttestdata,as.numeric)
testdata<-cbind(datanumtestdata,datafacttestdata)
#------------------------------------------------------------------------------
#keep only those which included in model1
testdata<-subset(testdata,select=c(SalePrice, Overall.Qual , Gr.Liv.Area , Exter.Qual , Kitchen.Qual ,
                              Total.Bsmt.SF , X1st.Flr.SF , Garage.Area , Year.Built ,
                              Foundation_PConc , Mas.Vnr.Area , Fireplaces , Heating.QC ,
                              BsmtFin.SF.1 , Bsmt.Exposure , Neighborhood_NridgHt , Lot.Frontage ,
                              Sale.Condition_Partial , Wood.Deck.SF , Neighborhood_NoRidge ,
                              Lot.Shape))
require(caret)
#########################################
#                                                         #
#                  evaluate model1                  #
#                                                         #
#########################################
###############################################
#--------Leave one out cross validation - LOOCV-----------
###############################################
set.seed(2822026)
# Define training control
train.control <- trainControl(method = "LOOCV")
# Train the model
model1aa <- train(SalePrice ~ Overall.Qual + Gr.Liv.Area + Exter.Qual + Kitchen.Qual +
                Total.Bsmt.SF + X1st.Flr.SF + Garage.Area + Year.Built +
                Mas.Vnr.Area + Fireplaces + Heating.QC +
                BsmtFin.SF.1 + Bsmt.Exposure + Neighborhood_NridgHt + Lot.Frontage +
                Sale.Condition_Partial + Wood.Deck.SF + Neighborhood_NoRidge +
                Lot.Shape, data=testdata, method = "lm", trControl = train.control)
# Summarize the results
print(model1aa)
#RMSE  43684.65
###########################################
#----------------10-fold cross-validation-----------------
###########################################
set.seed(2822026)
# Define training control
train.control <- trainControl(method = "cv", number = 10)
# Train the model
model1bb <- train(SalePrice ~ Overall.Qual + Gr.Liv.Area + Exter.Qual + Kitchen.Qual +
                Total.Bsmt.SF + X1st.Flr.SF + Garage.Area + Year.Built +
                Mas.Vnr.Area + Fireplaces + Heating.QC +
                BsmtFin.SF.1 + Bsmt.Exposure + Neighborhood_NridgHt + Lot.Frontage +
                Sale.Condition_Partial + Wood.Deck.SF + Neighborhood_NoRidge +
                Lot.Shape, data=testdata, method = "lm",trControl = train.control)
# Summarize the results
print(model1bb)
#RMSE 41080.06

#########################################
#                                                   #
#                       evaluate model2                  #
#                                                   #
#########################################
################################################
#--------Leave one out cross validation - LOOCV-----------
################################################
set.seed(2822026)
# Define training control
train.control <- trainControl(method = "LOOCV")
# Train the model
model2aa <- train(log(SalePrice) ~ Overall.Qual + Gr.Liv.Area + Kitchen.Qual +
                Total.Bsmt.SF +  Garage.Area + Year.Built +
                Fireplaces + Heating.QC +
                BsmtFin.SF.1 + Bsmt.Exposure + Lot.Frontage +
                Sale.Condition_Partial + Wood.Deck.SF +
                Lot.Shape, data=testdata, method = "lm", trControl = train.control)
# Summarize the results
print(model2aa)
#RMSE 0.2166
###########################################
#----------------10-fold cross-validation-----------------
###########################################
set.seed(2822026)
# Define training control
train.control <- trainControl(method = "cv", number = 10)
# Train the model
model2bb <- train(log(SalePrice) ~ Overall.Qual + Gr.Liv.Area + Kitchen.Qual +
                Total.Bsmt.SF +  Garage.Area + Year.Built +
                Fireplaces + Heating.QC +
                BsmtFin.SF.1 + Bsmt.Exposure + Lot.Frontage +
                Sale.Condition_Partial + Wood.Deck.SF +
                Lot.Shape, data=testdata, method = "lm",trControl = train.control)
# Summarize the results
print(model2bb)
#RMSE 0.2046978

####################################
#                                            #
#                   evaluate model3                #
#                                            #
####################################
################################################
#--------Leave one out cross validation - LOOCV-----------
################################################
```

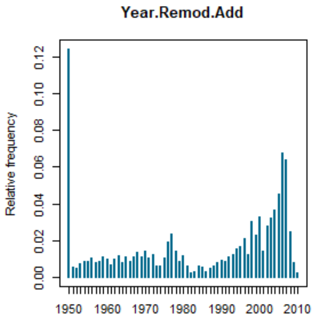
```
set.seed(2822026)
# Define training control
train.control <- trainControl(method = "LOOCV")
# Train the model
model3aa <- train(log(SalePrice) ~ poly(Overall.Qual,5)+
                  poly(Gr.Liv.Area,2)+ Kitchen.Qual + Total.Bsmt.SF +
                  poly(Garage.Area,4)+
                  Year.Built + Fireplaces + Heating.QC +
                  BsmtFin.SF.1 + Bsmt.Exposure  +
                  poly(Lot.Frontage,3) +
                  Sale.Condition_Partial + Wood.Deck.SF  +
                  Lot.Shape, data=testdata, method = "lm", trControl = train.control)
# Summarize the results
print(model3aa)
#RMSE 0.1929567

##########################################
#----------------10-fold cross-validation-----------------
##########################################
set.seed(2822026)
# Define training control
train.control <- trainControl(method = "cv", number = 10)
# Train the model
model3bb <- train(log(SalePrice) ~ poly(Overall.Qual,5)+
                  poly(Gr.Liv.Area,2)+ Kitchen.Qual + Total.Bsmt.SF +
                  poly(Garage.Area,4)+
                  Year.Built + Fireplaces + Heating.QC +
                  BsmtFin.SF.1 + Bsmt.Exposure  +
                  poly(Lot.Frontage,3) +
                  Sale.Condition_Partial + Wood.Deck.SF  +
                  Lot.Shape, data=testdata, method = "lm",trControl = train.control)
# Summarize the results
print(model3bb)
#RMSE 0.1876193
```

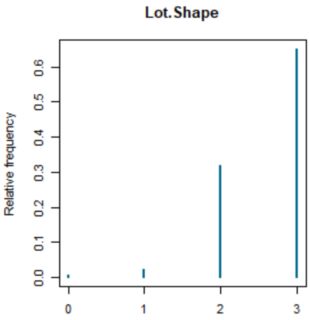
# List of Tables and Figures

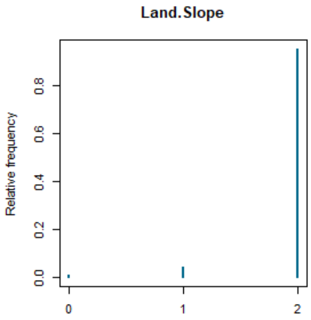
## Univariate analysis

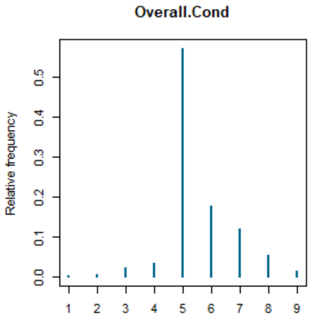
### Discrete Variables

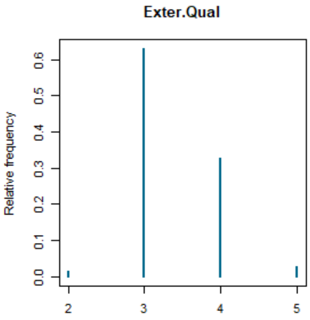


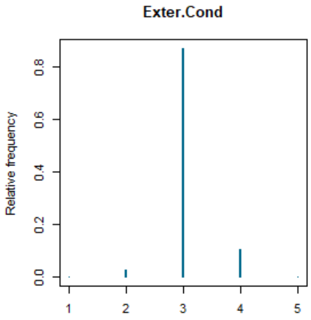

Year.Remod.Add
Relative frequency
0.00
0.02
0.04
0.06
0.08
0.10
0.12
1950
1960
1970
1980
1990
2000
2010

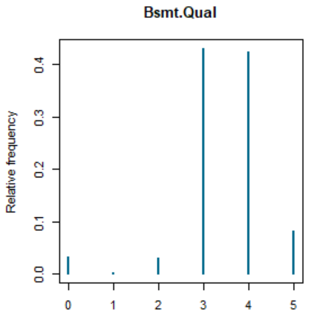

Lot.Shape
Relative frequency
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0
1
2
3

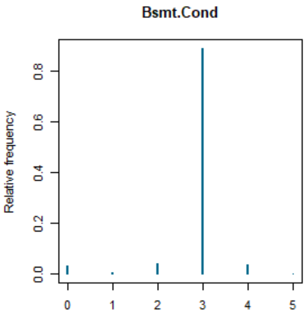

Land.Slope
Relative frequency
0.0
0.2
0.4
0.6
0.8
0
1
2

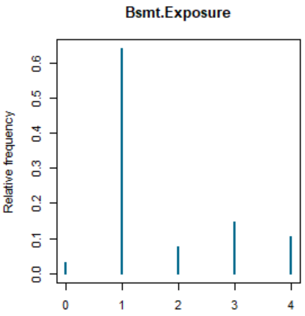

Overall.Cond
Relative frequency
0.0
0.1
0.2
0.3
0.4
0.5
1
2
3
4
5
6
7
8
9

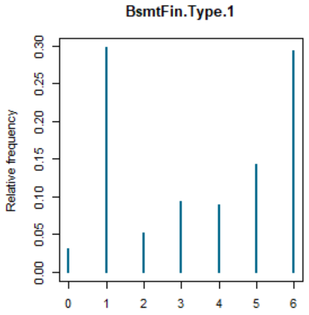

Exter.Qual
Relative frequency
0.0
0.1
0.2
0.3
0.4
0.5
0.6
2
3
4
5

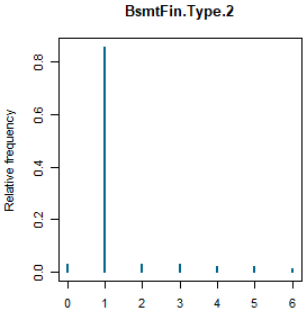

Exter.Cond
Relative frequency
0.0
0.2
0.4
0.6
0.8
1
2
3
4
5

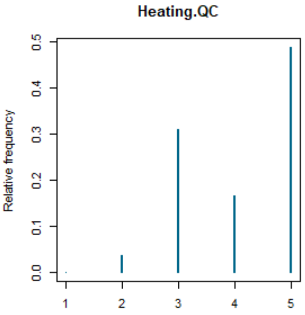

Bsmt.Qual
Relative frequency
0.0
0.1
0.2
0.3
0.4
0
1
2
3
4
5

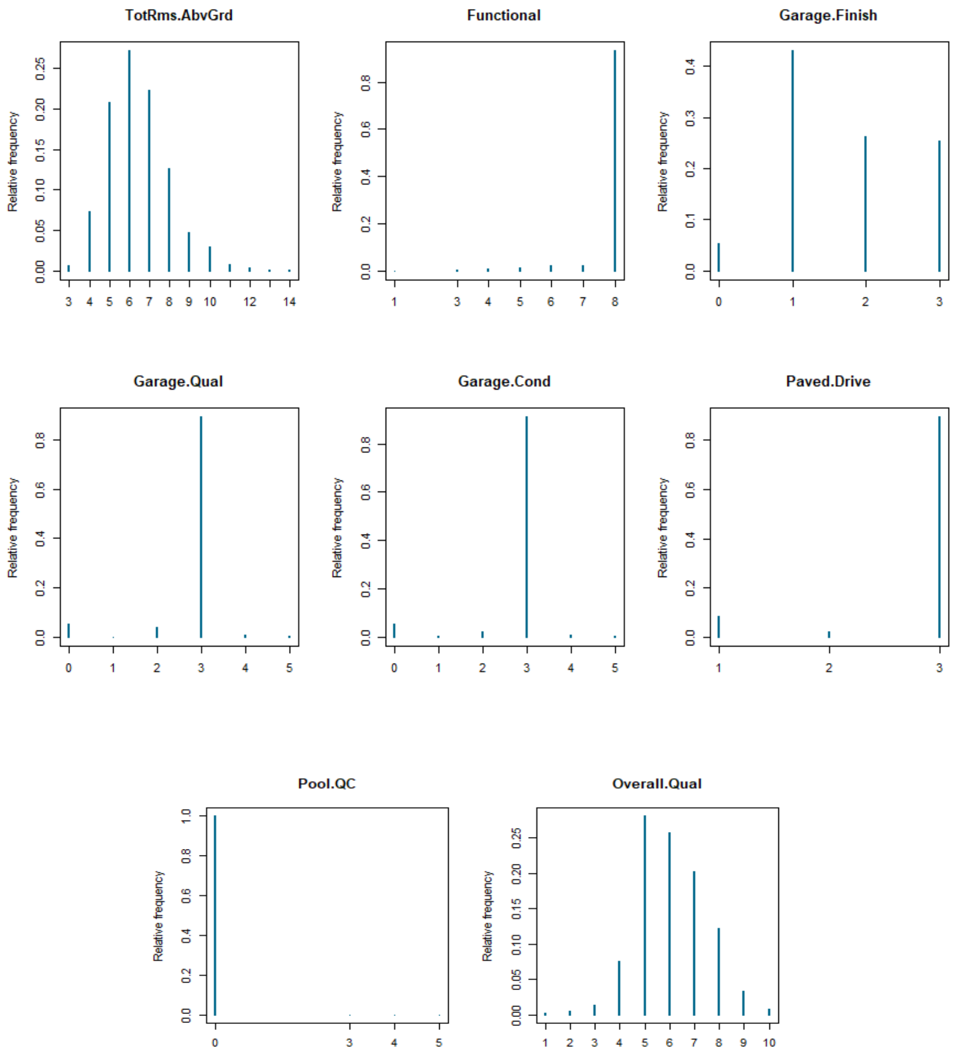

Bsmt.Cond
Relative frequency
0.0
0.2
0.4
0.6
0.8
0
1
2
3
4
5


Bsmt.Exposure
Relative frequency
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0
1
2
3
4

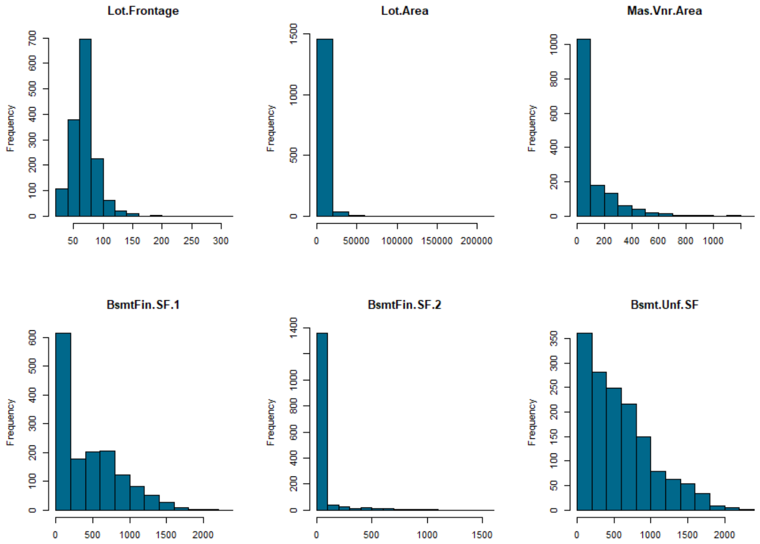

BsmtFin.Type.1
Relative frequency
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0
1
2
3
4
5
6

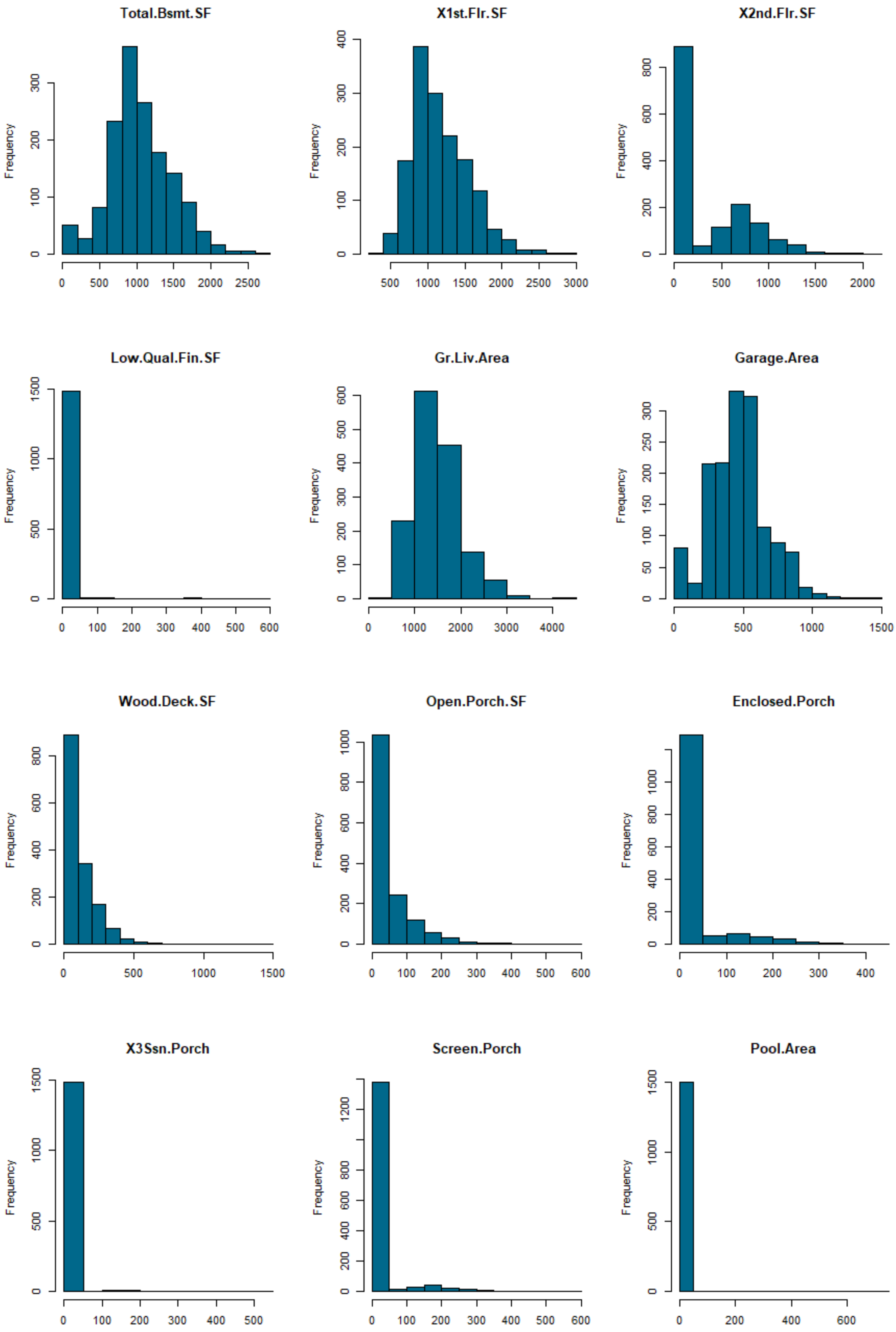

BsmtFin.Type.2
Relative frequency
0.0
0.2
0.4
0.6
0.8
0
1
2
3
4
5
6

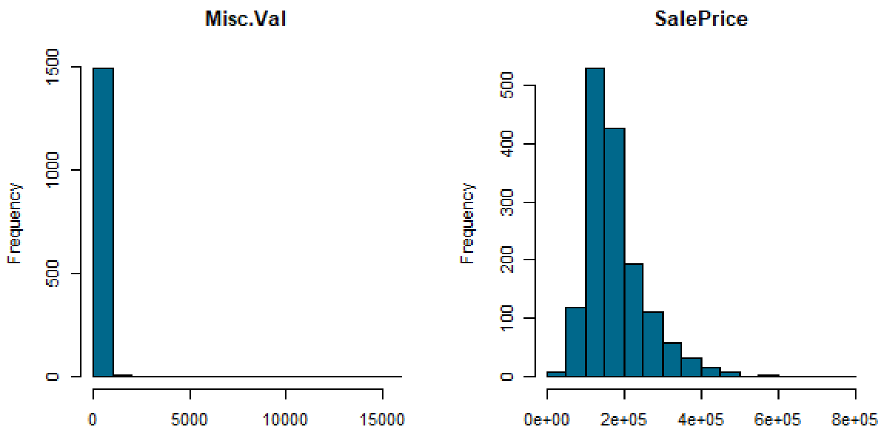

Heating.QC
Relative frequency
0.0
0.1
0.2
0.3
0.4
0.5
1
2
3
4
5

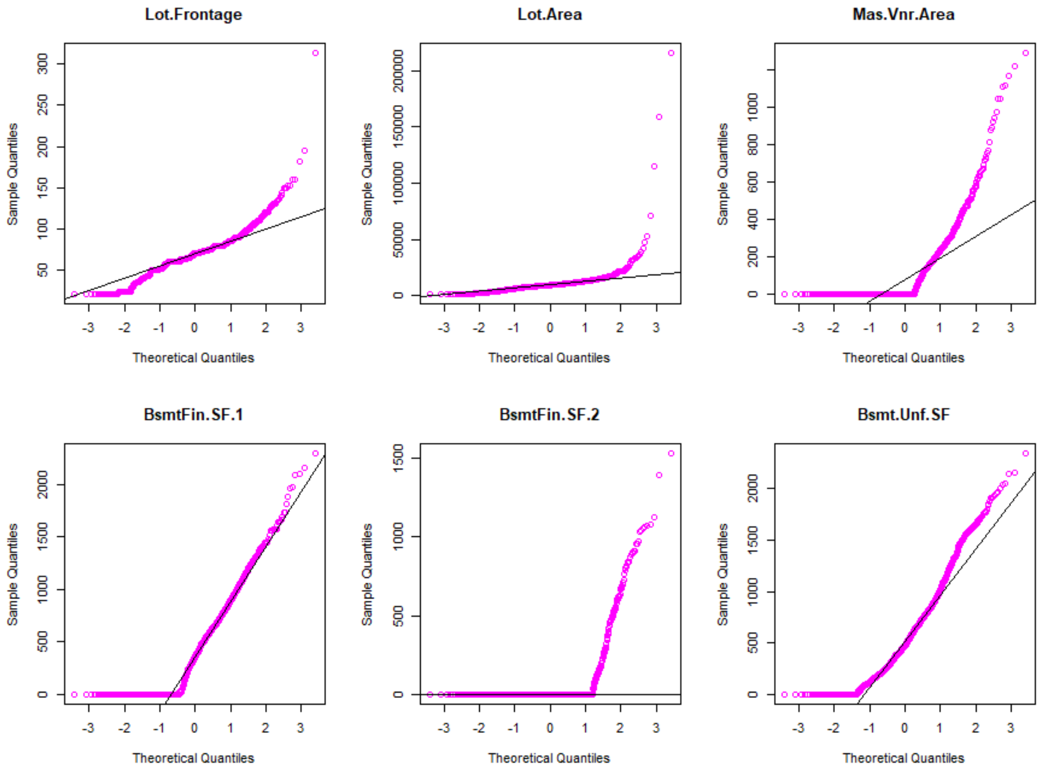


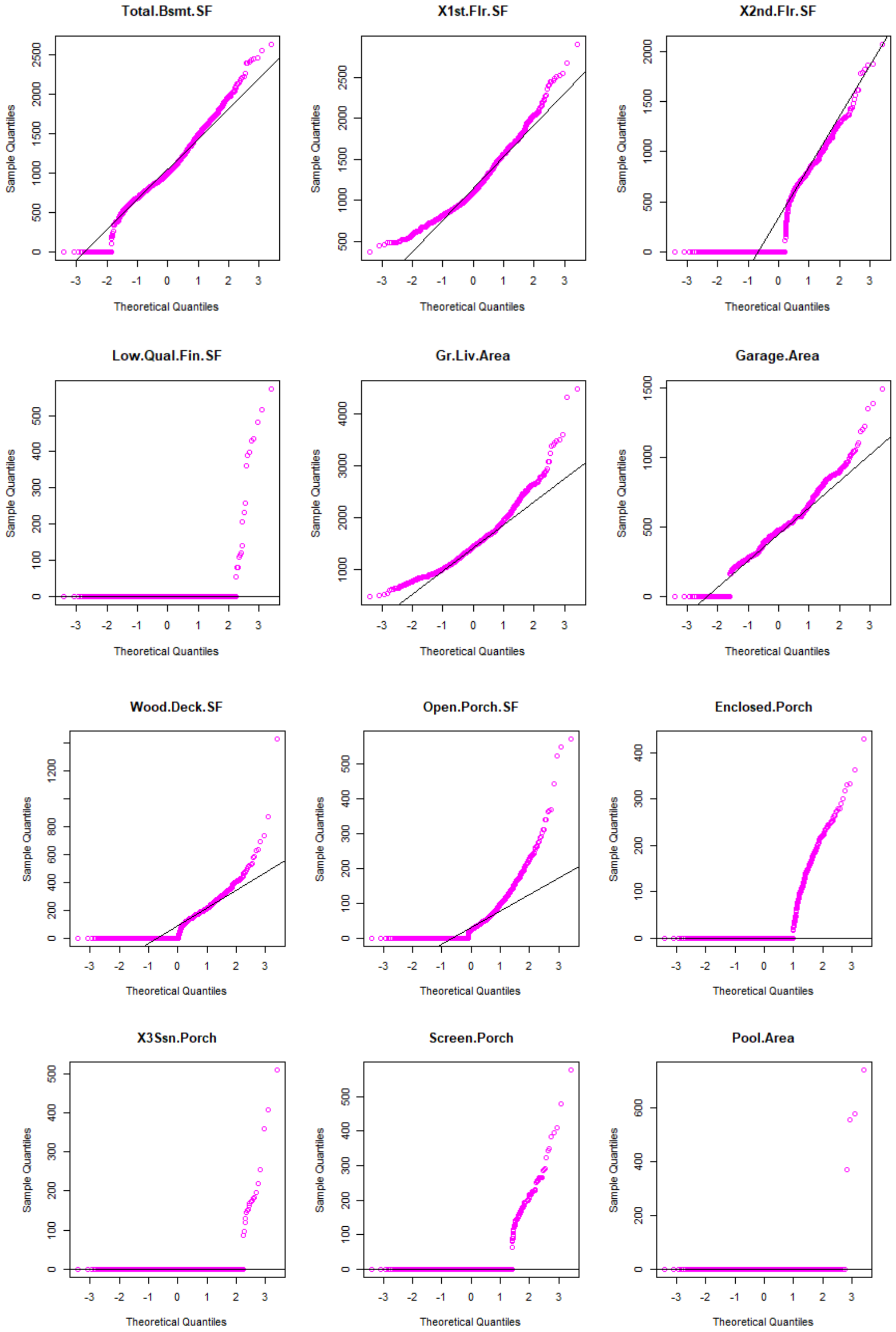
## Continuous Variables

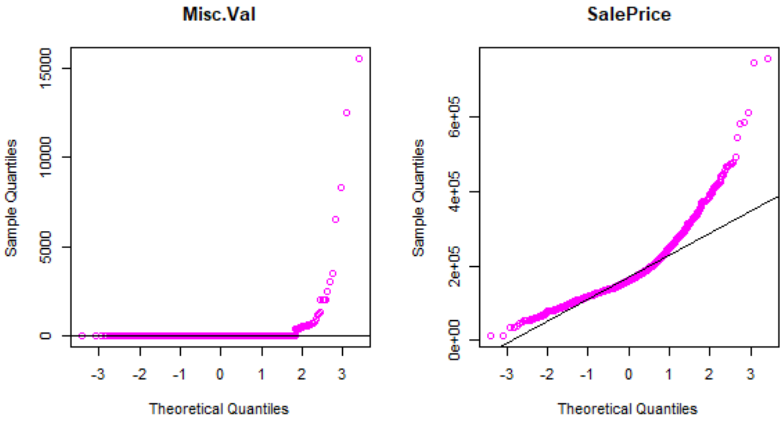



Total.Bsmt.SF
Frequency
0
100
200
300
0
500
1000
1500
2000
2500
X1st.Flr.SF
Frequency
0
100
200
300
400
500
1000
1500
2000
2500
3000
X2nd.Flr.SF
Frequency
0
200
400
600
800
0
500
1000
1500
2000
Low.Qual.Fin.SF
Frequency
0
500
1000
1500
0
100
200
300
400
500
600
Gr.Liv.Area
Frequency
0
100
200
300
400
500
600
0
1000
2000
3000
4000
Garage.Area
Frequency
0
50
100
150
200
250
300
0
500
1000
1500
Wood.Deck.SF
Frequency
0
200
400
600
800
0
500
1000
1500
Open.Porch.SF
Frequency
0
200
400
600
800
1000
0
100
200
300
400
500
600
Enclosed.Porch
Frequency
0
200
400
600
800
1000
0
100
200
300
400
X3Ssn.Porch
Frequency
0
500
1000
1500
0
100
200
300
400
500
Screen.Porch
Frequency
0
200
400
600
800
1200
0
100
200
300
400
500
600
Pool.Area
Frequency
0
500
1000
1500
0
200
400
600

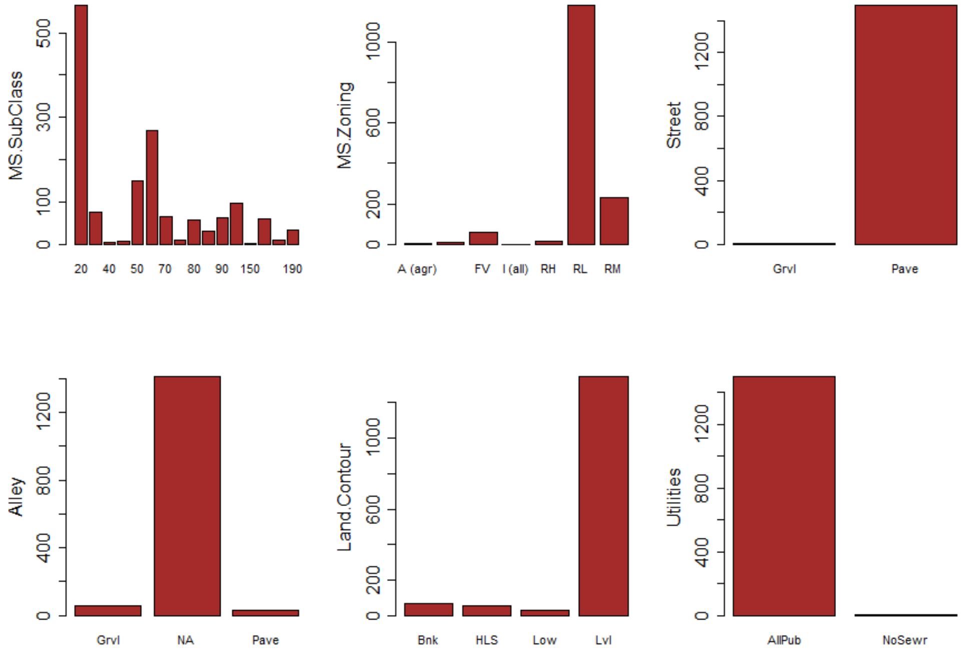

Misc.Val
Frequency
0
500
1000
1500
0
5000
10000
15000
SalePrice
Frequency
0
100
200
300
400
500
0e+00
2e+05
4e+05
6e+05
8e+05

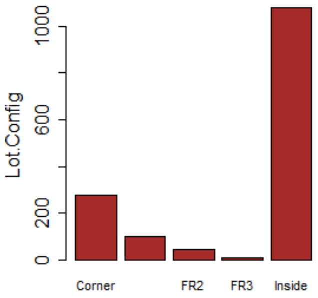

Lot.Frontage
Sample Quantiles
50
100
150
200
250
300
Theoretical Quantiles
-3
-2
-1
0
1
2
3
Lot.Area
Sample Quantiles
0
50000
100000
150000
200000
Theoretical Quantiles
Mas.Vnr.Area
Sample Quantiles
0
200
400
600
800
1000
Theoretical Quantiles
BsmtFin.SF.1
Sample Quantiles
0
500
1000
1500
2000
Theoretical Quantiles
BsmtFin.SF.2
Sample Quantiles
0
500
1000
1500
Theoretical Quantiles
Bsmt.Unf.SF
Sample Quantiles
0
500
1000
1500
2000
Theoretical Quantiles

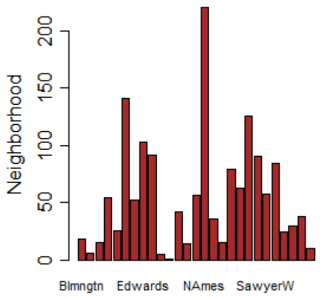

Total.Bsmt.SF
Sample Quantiles
0 500 1000 1500 2000 2500
-3 -2 -1 0 1 2 3
Theoretical Quantiles
X1st.Flr.SF
Sample Quantiles
500 1000 1500 2000 2500
-3 -2 -1 0 1 2 3
Theoretical Quantiles
X2nd.Flr.SF
Sample Quantiles
0 500 1000 1500 2000
-3 -2 -1 0 1 2 3
Theoretical Quantiles
Low.Qual.Fin.SF
Sample Quantiles
0 100 200 300 400 500
-3 -2 -1 0 1 2 3
Theoretical Quantiles
Gr.Liv.Area
Sample Quantiles
1000 2000 3000 4000
-3 -2 -1 0 1 2 3
Theoretical Quantiles
Garage.Area
Sample Quantiles
0 500 1000 1500
-3 -2 -1 0 1 2 3
Theoretical Quantiles
Wood.Deck.SF
Sample Quantiles
0 200 400 600 800 1200
-3 -2 -1 0 1 2 3
Theoretical Quantiles
Open.Porch.SF
Sample Quantiles
0 100 200 300 400 500
-3 -2 -1 0 1 2 3
Theoretical Quantiles
Enclosed.Porch
Sample Quantiles
0 100 200 300 400
-3 -2 -1 0 1 2 3
Theoretical Quantiles
X3Ssn.Porch
Sample Quantiles
0 100 200 300 400 500
-3 -2 -1 0 1 2 3
Theoretical Quantiles
Screen.Porch
Sample Quantiles
0 100 200 300 400 500
-3 -2 -1 0 1 2 3
Theoretical Quantiles
Pool.Area
Sample Quantiles
0 200 400 600
-3 -2 -1 0 1 2 3
Theoretical Quantiles

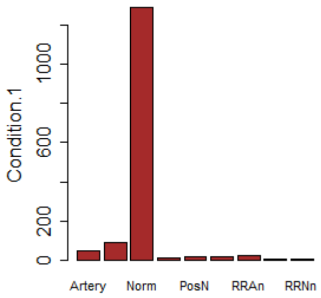


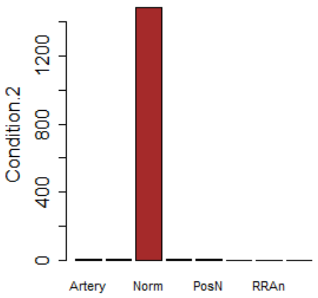
## Categorical variables

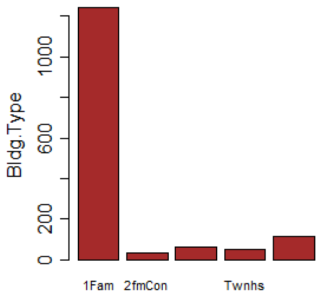


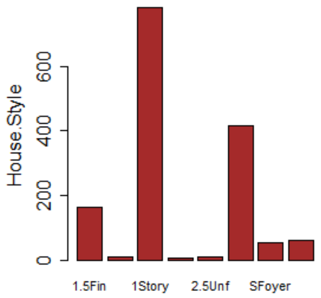

Lot.Config
0
200
600
1000
Corner
FR2
FR3
Inside

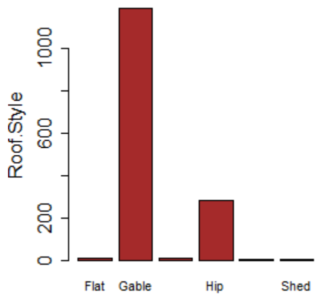

Neighborhood
0
50
100
150
200
Blmngtn
Edwards
NAmes
SawyerW

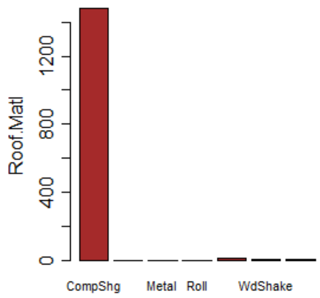

Condition.1
0
200
600
1000
Artery
Norm
PosN
RRAn
RRNn

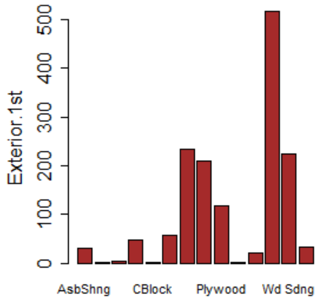

Condition.2
0
400
800
1200
Artery
Norm
PosN
RRAn

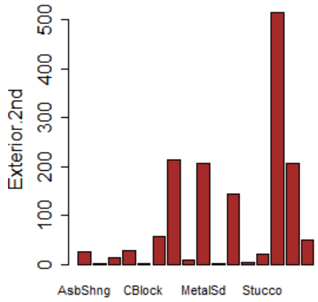

Bldg.Type
0
200
600
1000
1Fam
2fmCon
Twnhs

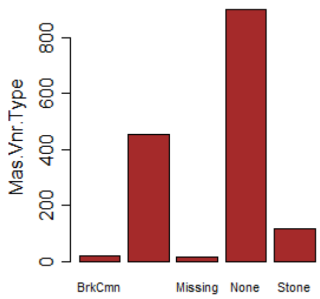

House.Style
0
200
400
600
1.5Fin
1Story
2.5Unf
SFoyer

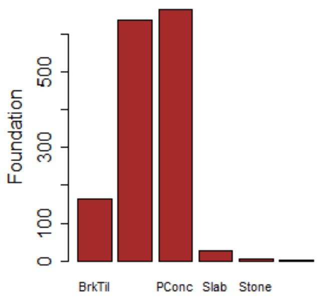

Roof.Style
0
200
600
1000
Flat
Gable
Hip
Shed

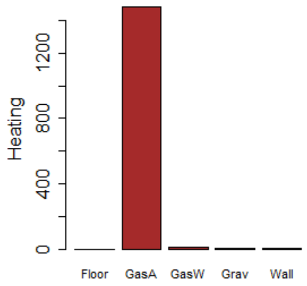

Roof.Matl
0
400
800
1200
CompShg
Metal
Roll
WdShake

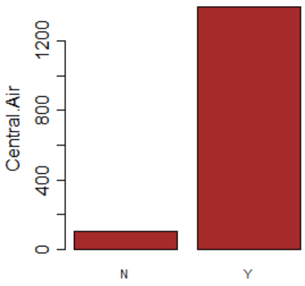

Exterior.1st
0
100
200
300
400
500
AsbShng
CBlock
Plywood
Wd Sdng

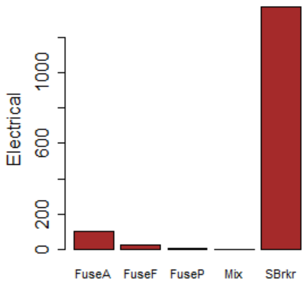

Exterior.2nd
0
100
200
300
400
500
AsbShng
CBlock
MetalSd
Stucco

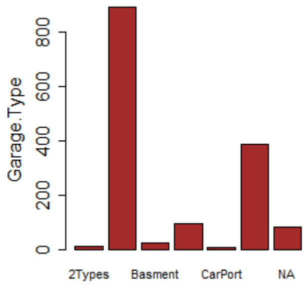

Mas.Vnr.Type
0
200
400
600
800
BrkCmn
Missing
None
Stone

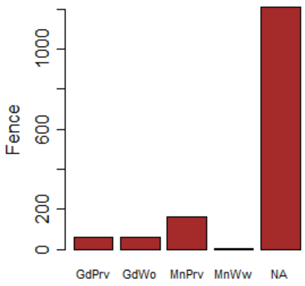

Foundation
0
100
300
500
BrkTil
PConc
Slab
Stone

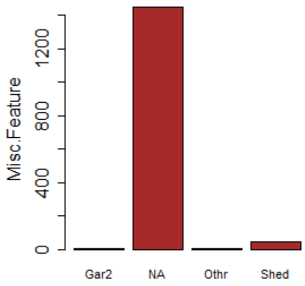

Heating
0
400
800
1200
Floor
GasA
GasW
Grav
Wall

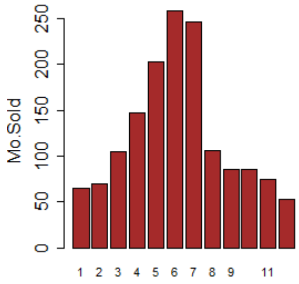

Central.Air
0
400
800
1200
N
Y

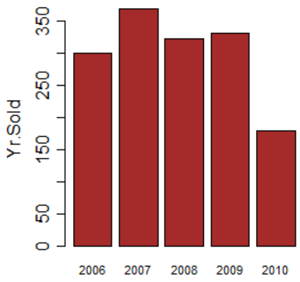

Electrical
0
200
600
1000
FuseA
FuseF
FuseP
Mix
SBrkr

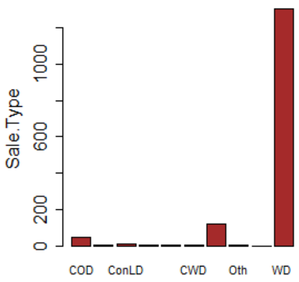

Garage.Type
0
200
400
600
800
2Types
Basment
CarPort
NA

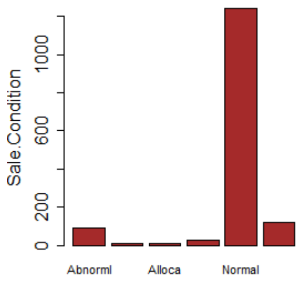

Fence
0
200
600
1000
GdPrv
GdWo
MnPrv
MnWw
NA


Misc.Feature
0
400
800
1200
Gar2
NA
Othr
Shed

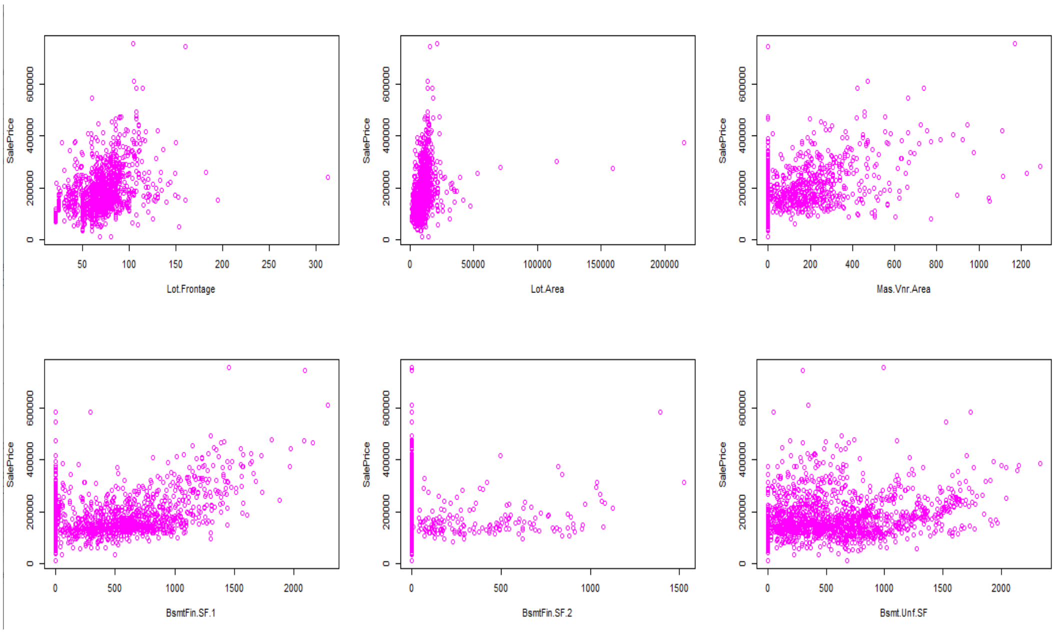

Mo.Sold
0
50
100
150
200
250
1
2
3
4
5
6
7
8
9
11

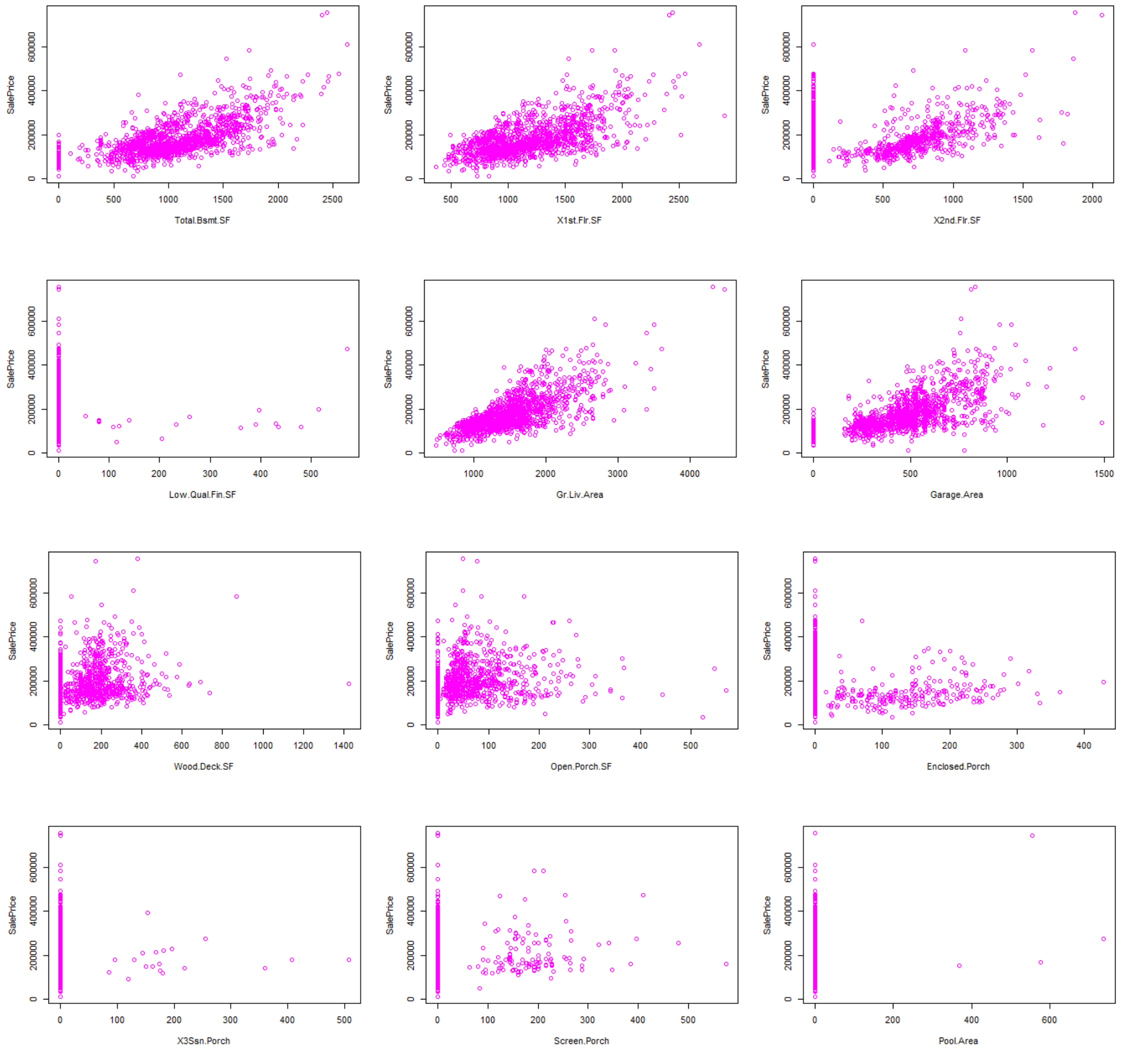

Yr.Sold
0
50
150
250
350
2006
2007
2008
2009
2010

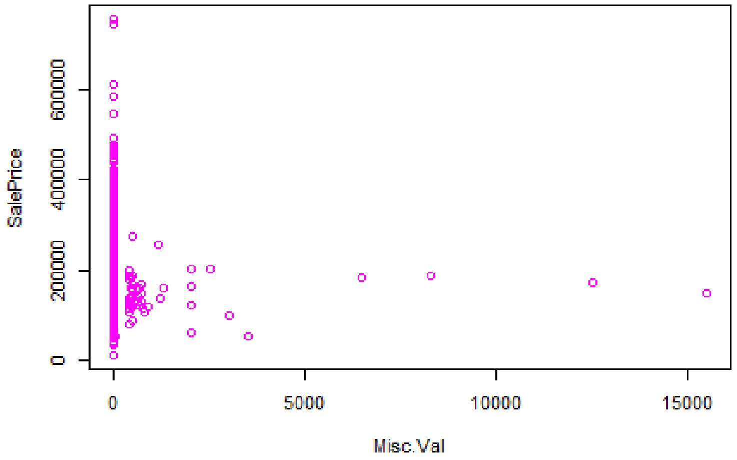

Sale.Type
0
200
600
1000
COD
ConLD
CWD
Oth
WD


Sale.Condition
0
200
600
1000
Abnorml
Alloca
Normal

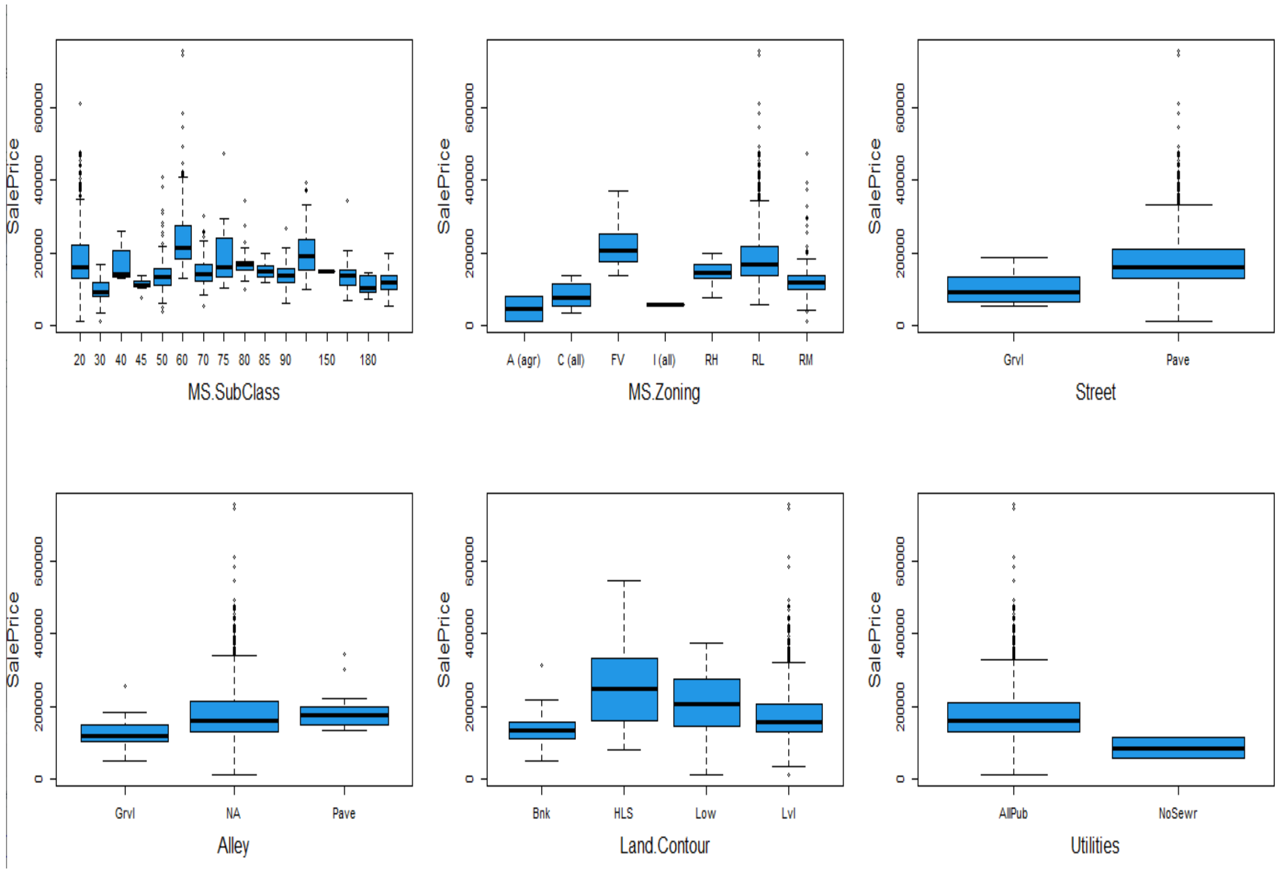
# Bivariate Analysis

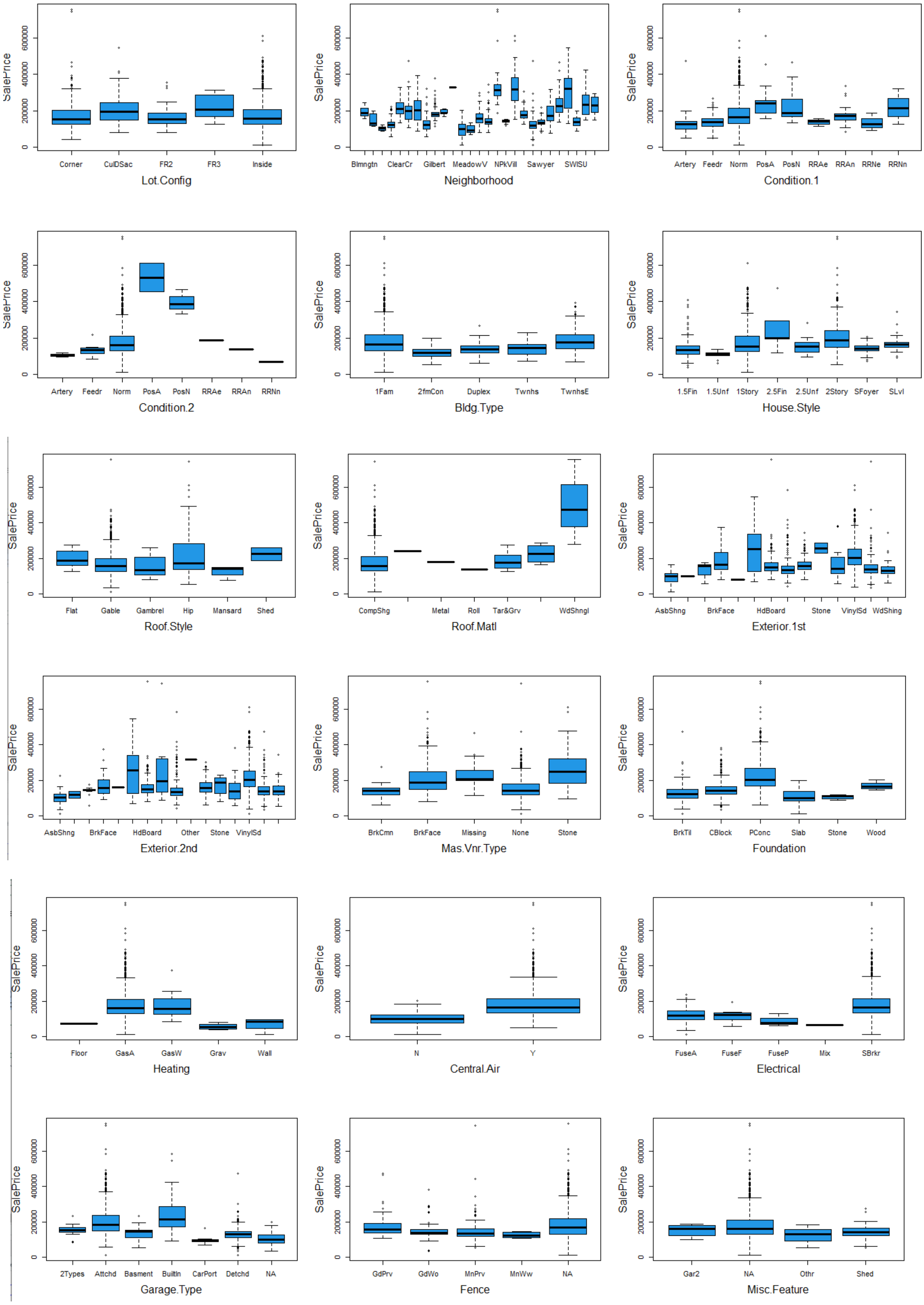




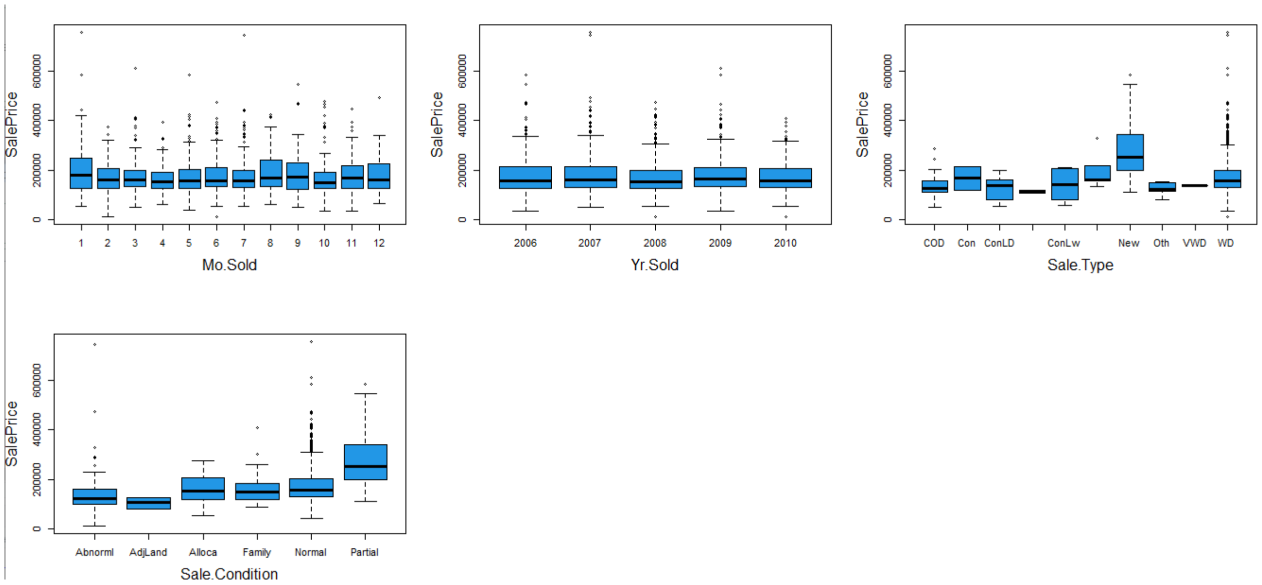


## Correlations of numeric variables and SalePrice





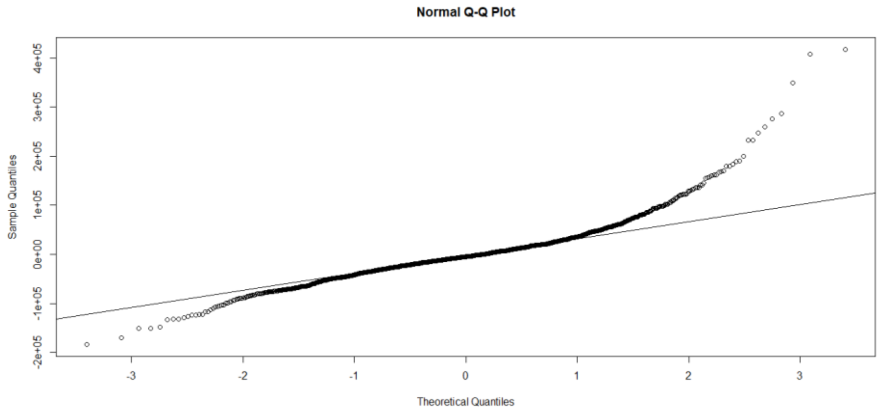
SalePrice
0
200000
400000
600000
Corner CulDSac FR2 FR3 Inside
Lot.Config
Blmngtn ClearCr Gilbert MeadowV NPkVill Sawyer SWISU
Neighborhood
Artery Feedr Norm PosA PosN RRAe RRAn RRNe RRNn
Condition.1
Artery Feedr Norm PosA PosN RRAe RRAn RRNn
Condition.2
1Fam 2fmCon Duplex Twnhs TwnhsE
Bldg.Type
1.5Fin 1.5Unf 1Story 2.5Fin 2.5Unf 2Story SFoyer SLvl
House.Style
Flat Gable Gambrel Hip Mansard Shed
Roof.Style
CompShg Metal Roll Tar&Grv WdShngl
Roof.Matl
AsbShng BrkFace HdBoard Stone VinylSd WdShing
Exterior.1st
AsbShng BrkFace HdBoard Other Stone VinylSd
Exterior.2nd
BrkCmn BrkFace Missing None Stone
Mas.Vnr.Type
BrkTil CBlock PConc Slab Stone Wood
Foundation
Floor GasA GasW Grav Wall
Heating
N Y
Central.Air
FuseA FuseF FuseP Mix SBrkr
Electrical
2Types Attchd Basment BuiltIn CarPort Detchd NA
Garage.Type
GdPrv GdWo MnPrv MnWw NA
Fence
Gar2 NA Othr Shed
Misc.Feature



## SalePrice – Neighborhood

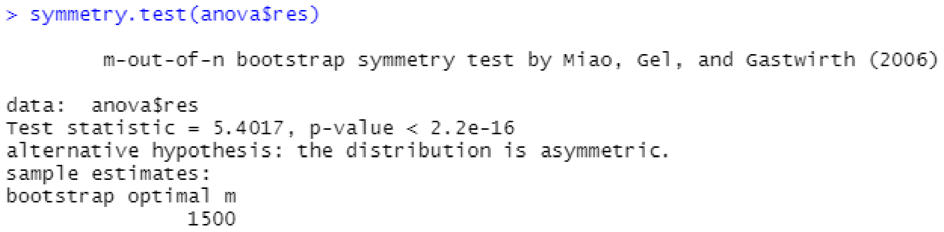
### Tests for k>2 independent samples - 1 quantitative & 1 categorical factor

Can we assume normality?



Do we have large sample? Yes.

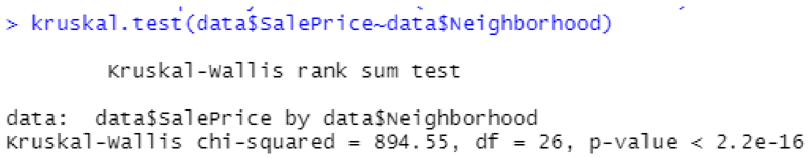
Is the mean a sufficient descriptive measure for central location for all groups?



```
> symmetry.test(anova$res)

        m-out-of-n bootstrap symmetry test by Miao, Gel, and Gastwirth (2006)

data:  anova$res
Test statistic = 5.4017, p-value < 2.2e-16
alternative hypothesis: the distribution is asymmetric.
sample estimates:
bootstrap optimal m
               1500
```

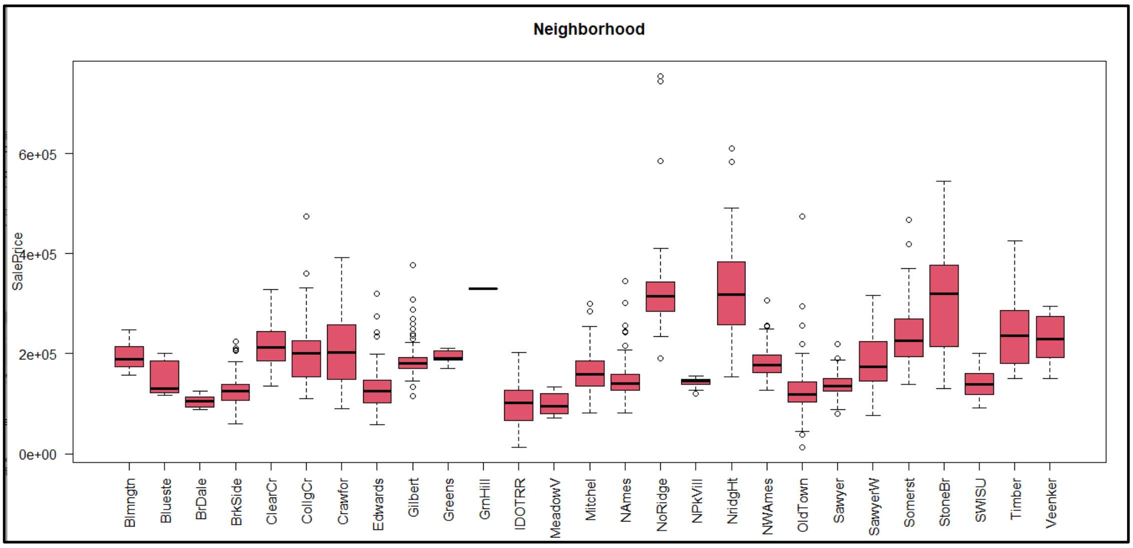
We reject Ho (P-value < 2.2e-16 < 0.05). The mean is not a sufficient descriptive measure for central location for all groups.

Test for the equality of medians (Kruskall-wallis test)



```
> kruskal.test(data$SalePrice~data$Neighborhood)

        Kruskal-Wallis rank sum test

data:  data$SalePrice by data$Neighborhood
Kruskal-Wallis chi-squared = 894.55, df = 26, p-value < 2.2e-16
```

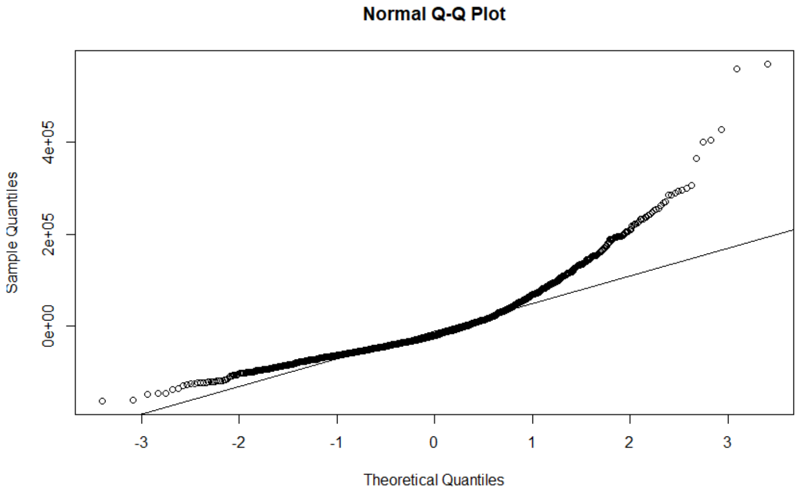
There is a significant difference in the median of the SalePrice between some Neighbourhood groups (Kruskal.wallis test, reject Ho, p.value < 2.2e-16 < 0.05).



## Saleprice – YrSold

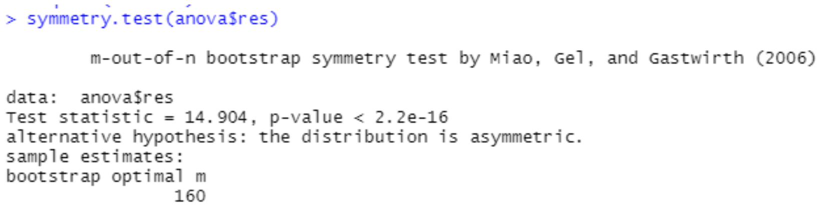
### Tests for k>2 independent samples - 1 quantitative & 1 categorical factor

Can we assume normality?



Do we have large sample? Yes.

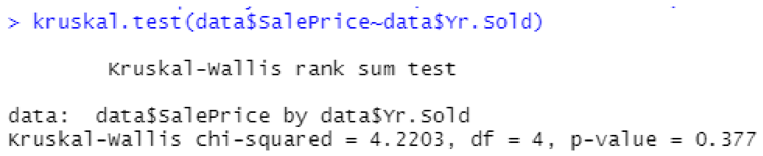
Is the mean a sufficient descriptive measure for central location for all groups?



```
> symmetry.test(anova$res)

        m-out-of-n bootstrap symmetry test by Miao, Gel, and Gastwirth (2006)

data:  anova$res
Test statistic = 14.904, p-value < 2.2e-16
alternative hypothesis: the distribution is asymmetric.
sample estimates:
bootstrap optimal m
                160
```

We reject Ho (P-value < 2.2e-16 < 0.05). The mean is not a sufficient descriptive measure for central location for all groups.

Test for the equality of medians (Kruskall-wallis test)



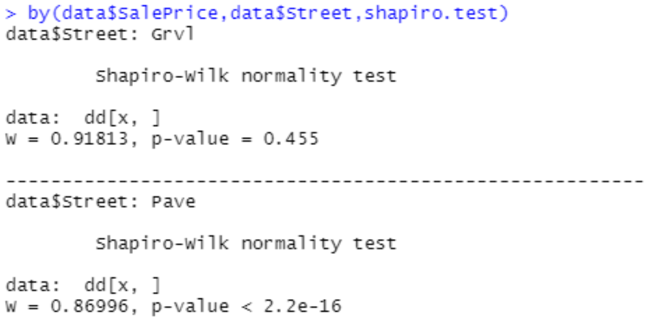
```
> kruskal.test(data$SalePrice~data$Yr.Sold)

        Kruskal-Wallis rank sum test

data:  data$SalePrice by data$Yr.Sold
Kruskal-Wallis chi-squared = 4.2203, df = 4, p-value = 0.377
```

We do not reject Ho (Kruskal.Wallis Test, p.value = 0.377 > 0.05). There is no significant difference in the median of the SalePrice between Yr.Sold groups.

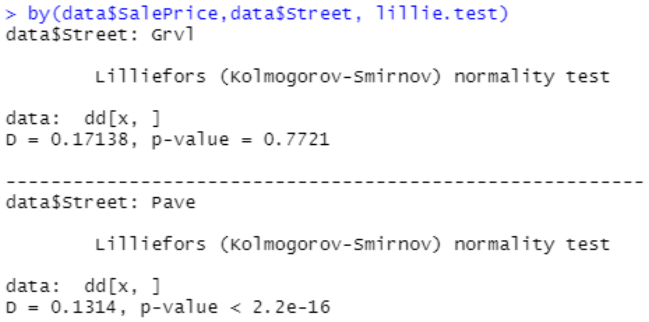
## Sales – Street

### Test for 2 independent samples - 1 quantitative & 1 binary

Can we assume the normality?



```
> by(data$SalePrice,data$Street,shapiro.test)
data$Street: Grvl

        Shapiro-Wilk normality test

data:  dd[x, ]
W = 0.91813, p-value = 0.455

------------------------------------------------------------
data$Street: Pave

        Shapiro-Wilk normality test

data:  dd[x, ]
W = 0.86996, p-value < 2.2e-16
```

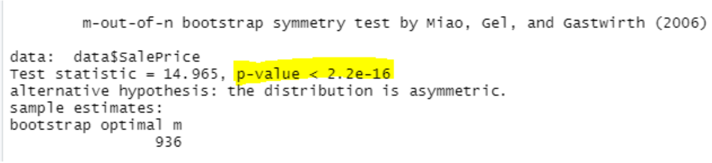


```
> by(data$SalePrice,data$Street, lillie.test)
data$Street: Grvl

        Lilliefors (Kolmogorov-Smirnov) normality test

data:  dd[x, ]
D = 0.17138, p-value = 0.7721

------------------------------------------------------------
data$Street: Pave

        Lilliefors (Kolmogorov-Smirnov) normality test

data:  dd[x, ]
D = 0.1314, p-value < 2.2e-16
```

P-value < 0.05 in at least one category. We reject the normality.

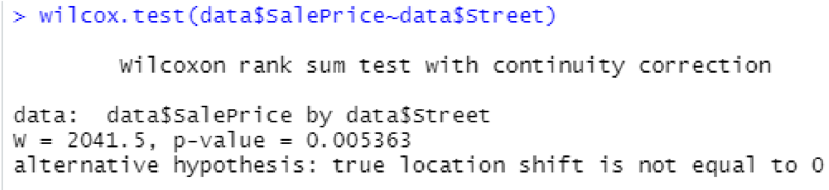
Do we have large sample? Yes.

Is the mean a sufficient descriptive measure for central location for all groups?



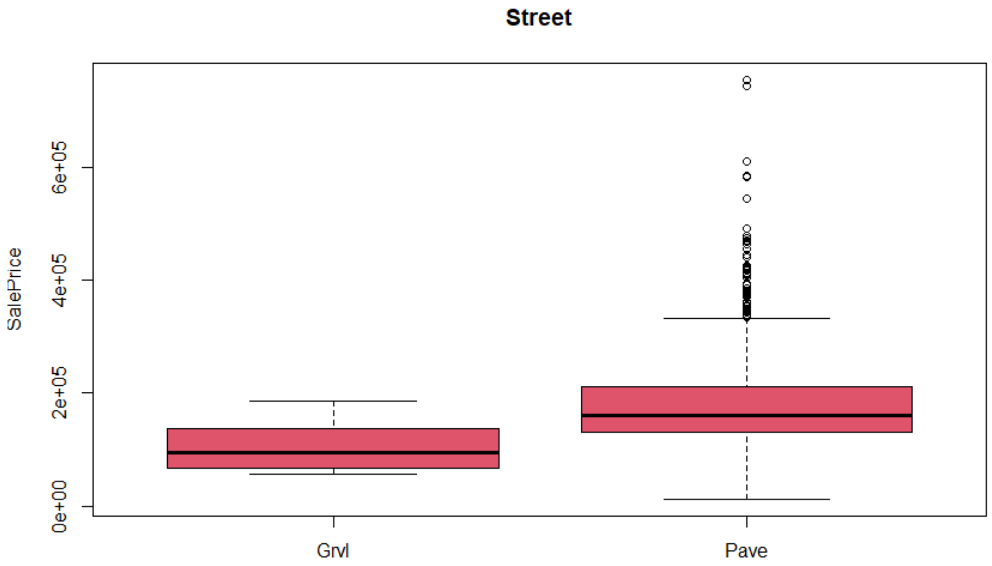
```
        m-out-of-n bootstrap symmetry test by Miao, Gel, and Gastwirth (2006)

data:  data$SalePrice
Test statistic = 14.965, p-value < 2.2e-16
alternative hypothesis: the distribution is asymmetric.
sample estimates:
bootstrap optimal m
                936
```

Test for zero difference between the medians



```
> wilcox.test(data$SalePrice~data$Street)

        Wilcoxon rank sum test with continuity correction

data:  data$SalePrice by data$Street
W = 2041.5, p-value = 0.005363
alternative hypothesis: true location shift is not equal to 0
```

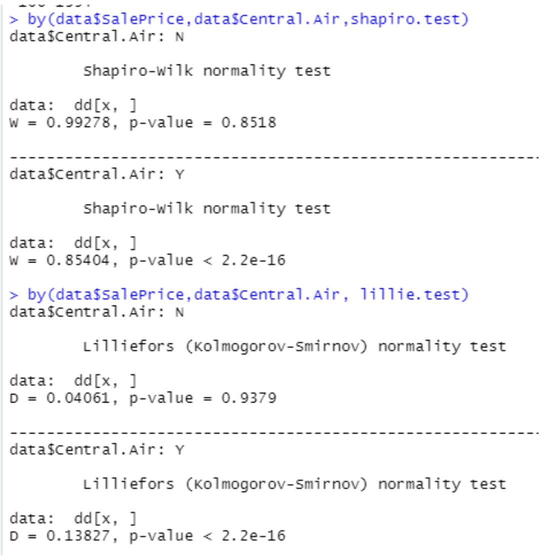
We reject Ho: M1=M2 significant difference is found about the median of the Saleprice between grvl and pave.



# Sales -Central Air

## Test for 2 independent samples - 1 quantitative & 1 binary

Can we assume the normality?

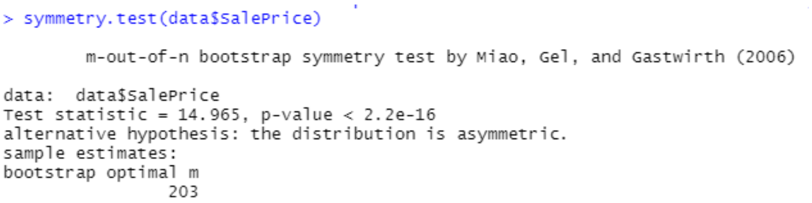


```
> by(data$SalePrice,data$Central.Air,shapiro.test)
data$Central.Air: N

        Shapiro-Wilk normality test

data:  dd[x, ]
W = 0.99278, p-value = 0.8518

---------------------------------------------------------
data$Central.Air: Y

        Shapiro-Wilk normality test

data:  dd[x, ]
W = 0.85404, p-value < 2.2e-16

> by(data$SalePrice,data$Central.Air, lillie.test)
data$Central.Air: N

        Lilliefors (Kolmogorov-Smirnov) normality test

data:  dd[x, ]
D = 0.04061, p-value = 0.9379

---------------------------------------------------------
data$Central.Air: Y

        Lilliefors (Kolmogorov-Smirnov) normality test

data:  dd[x, ]
D = 0.13827, p-value < 2.2e-16
```

P-value < 0.05 in at least one category. We reject the normality.

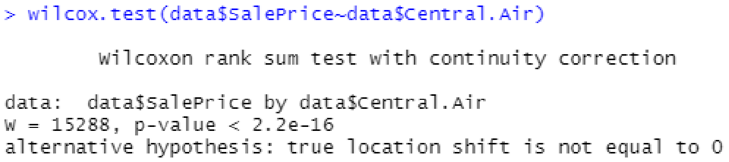
Do we have large sample? Yes.

Is the mean a sufficient descriptive measure for central location for all groups?



```
> symmetry.test(data$SalePrice)

        m-out-of-n bootstrap symmetry test by Miao, Gel, and Gastwirth (2006)

data:  data$SalePrice
Test statistic = 14.965, p-value < 2.2e-16
alternative hypothesis: the distribution is asymmetric.
sample estimates:
bootstrap optimal m
                203
```

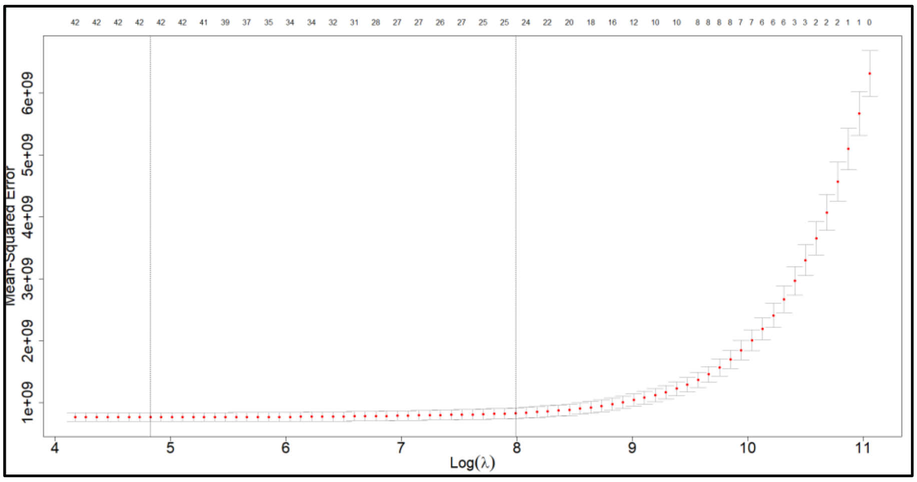
Test for zero difference between the medians



```
> wilcox.test(data$SalePrice~data$Central.Air)

        Wilcoxon rank sum test with continuity correction

data:  data$SalePrice by data$Central.Air
W = 15288, p-value < 2.2e-16
alternative hypothesis: true location shift is not equal to 0
```

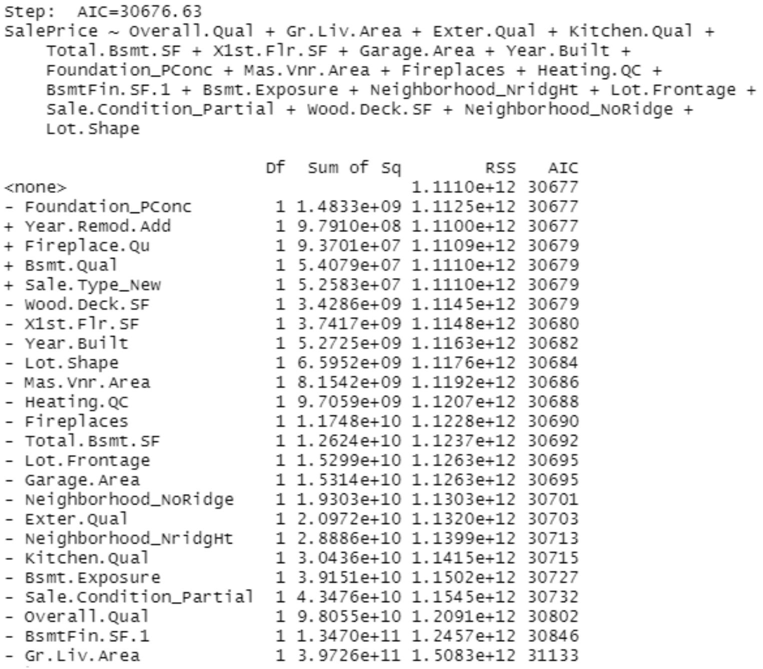
There is a significant difference in the median of the SalePrice between those houses who have Central Air and those who do not have.

## LASSO



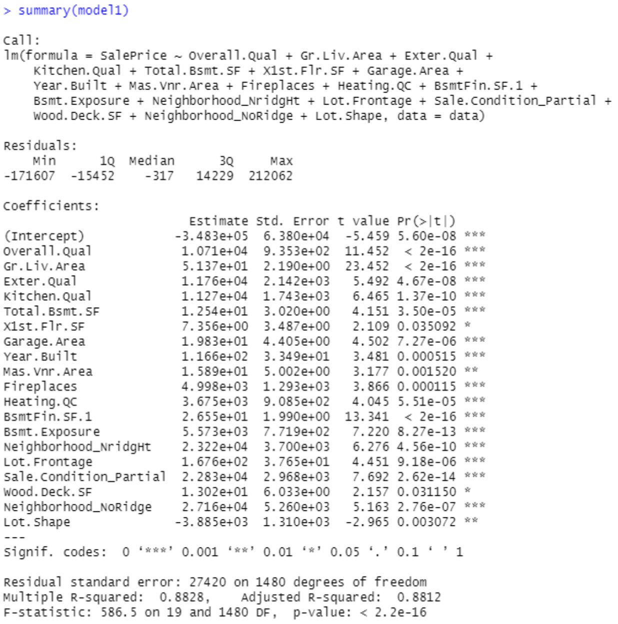
The plot above shows the log lambda value on the x-axis and the mean-squared error on the y-axis. Above the plot, there are numbers that correspond to the number of predictors in the model at each value of log lambda. The first vertical dotted line represents the lowest mean-squared error. In this model, there are 42 variables included in our model. However, we do not want to necessarily build a model with the lowest mean-squared error. Instead, we want to build a model with a low mean-squared error that is parsimonious. As we can see, for only a little increase in the mean-squared error, we get a model that only has 25 predictors.

## Stepwise procedure according to AIC



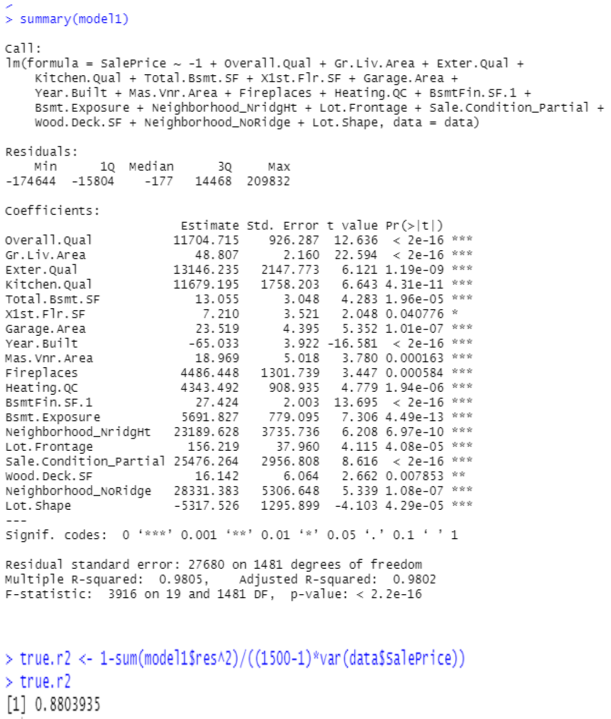
```
Step:  AIC=30676.63
SalePrice ~ Overall.Qual + Gr.Liv.Area + Exter.Qual + Kitchen.Qual + 
    Total.Bsmt.SF + X1st.Flr.SF + Garage.Area + Year.Built + 
    Foundation_PConc + Mas.Vnr.Area + Fireplaces + Heating.QC + 
    BsmtFin.SF.1 + Bsmt.Exposure + Neighborhood_NridgHt + Lot.Frontage + 
    Sale.Condition_Partial + Wood.Deck.SF + Neighborhood_NoRidge + 
    Lot.Shape

                         Df  Sum of Sq        RSS   AIC
<none>                                 1.1110e+12 30677
- Foundation_PConc        1 1.4833e+09 1.1125e+12 30677
+ Year.Remod.Add          1 9.7910e+08 1.1100e+12 30677
+ Fireplace.Qu            1 9.3701e+07 1.1109e+12 30679
+ Bsmt.Qual               1 5.4079e+07 1.1110e+12 30679
+ Sale.Type_New           1 5.2583e+07 1.1110e+12 30679
- Wood.Deck.SF            1 3.4286e+09 1.1145e+12 30679
- X1st.Flr.SF             1 3.7417e+09 1.1148e+12 30680
- Year.Built              1 5.2725e+09 1.1163e+12 30682
- Lot.Shape               1 6.5952e+09 1.1176e+12 30684
- Mas.Vnr.Area            1 8.1542e+09 1.1192e+12 30686
- Heating.QC              1 9.7059e+09 1.1207e+12 30688
- Fireplaces              1 1.1748e+10 1.1228e+12 30690
- Total.Bsmt.SF           1 1.2624e+10 1.1237e+12 30692
- Lot.Frontage            1 1.5299e+10 1.1263e+12 30695
- Garage.Area             1 1.5314e+10 1.1263e+12 30695
- Neighborhood_NoRidge    1 1.9303e+10 1.1303e+12 30701
- Exter.Qual              1 2.0972e+10 1.1320e+12 30703
- Neighborhood_NridgHt    1 2.8886e+10 1.1399e+12 30713
- Kitchen.Qual            1 3.0436e+10 1.1415e+12 30715
- Bsmt.Exposure           1 3.9151e+10 1.1502e+12 30727
- Sale.Condition_Partial  1 4.3476e+10 1.1545e+12 30732
- Overall.Qual            1 9.8055e+10 1.2091e+12 30802
- BsmtFin.SF.1            1 1.3470e+11 1.2457e+12 30846
- Gr.Liv.Area             1 3.9726e+11 1.5083e+12 31133
```

## Estimate of stepwise model – Model1



```
> summary(model1)

Call:
lm(formula = SalePrice ~ Overall.Qual + Gr.Liv.Area + Exter.Qual +
    Kitchen.Qual + Total.Bsmt.SF + X1st.Flr.SF + Garage.Area +
    Year.Built + Mas.Vnr.Area + Fireplaces + Heating.QC + BsmtFin.SF.1 +
    Bsmt.Exposure + Neighborhood_NridgHt + Lot.Frontage + Sale.Condition_Partial +
    Wood.Deck.SF + Neighborhood_NoRidge + Lot.Shape, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-171607  -15452    -317   14229  212062

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)            -3.483e+05  6.380e+04  -5.459 5.60e-08 ***
Overall.Qual            1.071e+04  9.353e+02  11.452  < 2e-16 ***
Gr.Liv.Area             5.137e+01  2.190e+00  23.452  < 2e-16 ***
Exter.Qual              1.176e+04  2.142e+03   5.492 4.67e-08 ***
Kitchen.Qual            1.127e+04  1.743e+03   6.465 1.37e-10 ***
Total.Bsmt.SF           1.254e+01  3.020e+00   4.151 3.50e-05 ***
X1st.Flr.SF             7.356e+00  3.487e+00   2.109 0.035092 *
Garage.Area             1.983e+01  4.405e+00   4.502 7.27e-06 ***
Year.Built              1.166e+02  3.349e+01   3.481 0.000515 ***
Mas.Vnr.Area            1.589e+01  5.002e+00   3.177 0.001520 **
Fireplaces              4.998e+03  1.293e+03   3.866 0.000115 ***
Heating.QC              3.675e+03  9.085e+02   4.045 5.51e-05 ***
BsmtFin.SF.1            2.655e+01  1.990e+00  13.341  < 2e-16 ***
Bsmt.Exposure           5.573e+03  7.719e+02   7.220 8.27e-13 ***
Neighborhood_NridgHt    2.322e+04  3.700e+03   6.276 4.56e-10 ***
Lot.Frontage            1.676e+02  3.765e+01   4.451 9.18e-06 ***
Sale.Condition_Partial  2.283e+04  2.968e+03   7.692 2.62e-14 ***
Wood.Deck.SF            1.302e+01  6.033e+00   2.157 0.031150 *
Neighborhood_NoRidge    2.716e+04  5.260e+03   5.163 2.76e-07 ***
Lot.Shape              -3.885e+03  1.310e+03  -2.965 0.003072 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 27420 on 1480 degrees of freedom
Multiple R-squared:  0.8828,    Adjusted R-squared:  0.8812
F-statistic: 586.5 on 19 and 1480 DF,  p-value: < 2.2e-16
```

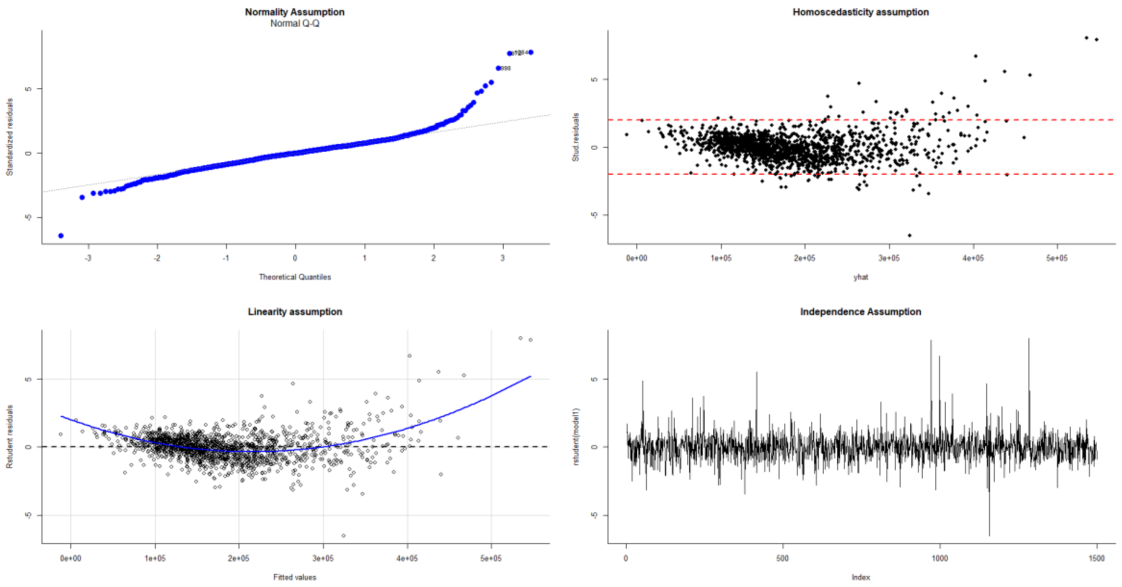
## Model 1 without constant

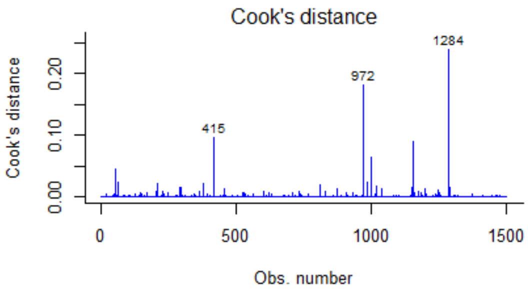


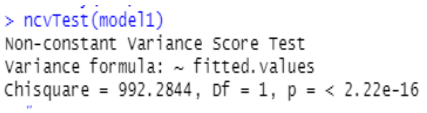
```
> summary(model1)

Call:
lm(formula = SalePrice ~ -1 + Overall.Qual + Gr.Liv.Area + Exter.Qual +
    Kitchen.Qual + Total.Bsmt.SF + X1st.Flr.SF + Garage.Area +
    Year.Built + Mas.Vnr.Area + Fireplaces + Heating.QC + BsmtFin.SF.1 +
    Bsmt.Exposure + Neighborhood_NridgHt + Lot.Frontage + Sale.Condition_Partial +
    Wood.Deck.SF + Neighborhood_NoRidge + Lot.Shape, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-174644  -15804    -177   14468  209832

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
Overall.Qual           11704.715    926.287  12.636  < 2e-16 ***
Gr.Liv.Area               48.807      2.160  22.594  < 2e-16 ***
Exter.Qual             13146.235   2147.773   6.121 1.19e-09 ***
Kitchen.Qual           11679.195   1758.203   6.643 4.31e-11 ***
Total.Bsmt.SF             13.055      3.048   4.283 1.96e-05 ***
X1st.Flr.SF                7.210      3.521   2.048 0.040776 *
Garage.Area               23.519      4.395   5.352 1.01e-07 ***
Year.Built               -65.033      3.922 -16.581  < 2e-16 ***
Mas.Vnr.Area              18.969      5.018   3.780 0.000163 ***
Fireplaces              4486.448   1301.739   3.447 0.000584 ***
Heating.QC              4343.492    908.935   4.779 1.94e-06 ***
BsmtFin.SF.1              27.424      2.003  13.695  < 2e-16 ***
Bsmt.Exposure           5691.827    779.095   7.306 4.49e-13 ***
Neighborhood_NridgHt   23189.628   3735.736   6.208 6.97e-10 ***
Lot.Frontage             156.219     37.960   4.115 4.08e-05 ***
Sale.Condition_Partial 25476.264   2956.808   8.616  < 2e-16 ***
Wood.Deck.SF              16.142      6.064   2.662 0.007853 **
Neighborhood_NoRidge   28331.383   5306.648   5.339 1.08e-07 ***
Lot.Shape              -5317.526   1295.899  -4.103 4.29e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 27680 on 1481 degrees of freedom
Multiple R-squared:  0.9805,    Adjusted R-squared:  0.9802
F-statistic:  3916 on 19 and 1481 DF,  p-value: < 2.2e-16


> true.r2 <- 1-sum(model1$res^2)/((1500-1)*var(data$SalePrice))
> true.r2
[1] 0.8803935
```

We choose the model with constant as Radj^2 is better and constant is statistically significant.

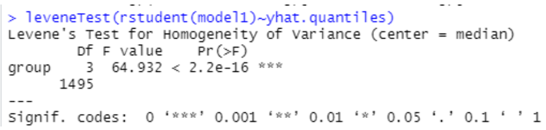
## Residual Assumptions of Model 1



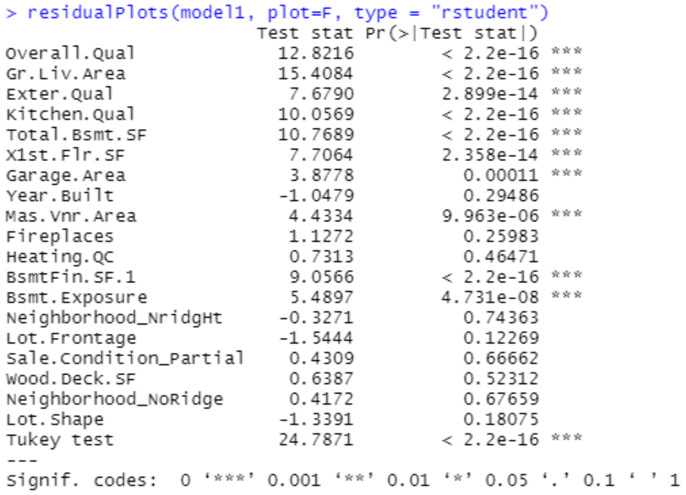




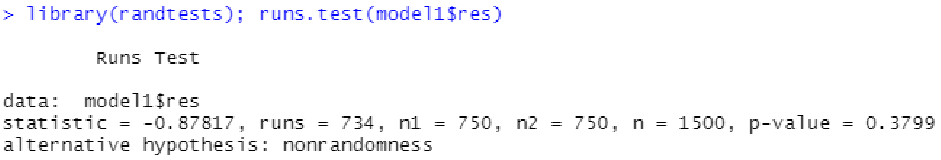
```
> ncvTest(model1)
Non-constant Variance Score Test 
Variance formula: ~ fitted.values 
Chisquare = 992.2844, Df = 1, p = < 2.22e-16
```



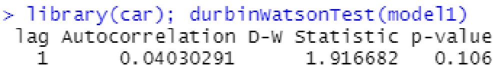
```
> leveneTest(rstudent(model1)~yhat.quantiles)
Levene's Test for Homogeneity of Variance (center = median)
        Df F value    Pr(>F)    
group    3  64.932 < 2.2e-16 ***
      1495                      
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```



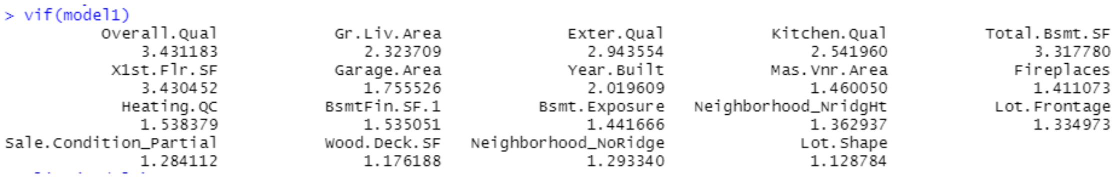
```
> residualPlots(model1, plot=F, type = "rstudent")
                       Test stat Pr(>|Test stat|)    
Overall.Qual             12.8216        < 2.2e-16 ***
Gr.Liv.Area              15.4084        < 2.2e-16 ***
Exter.Qual                7.6790        2.899e-14 ***
Kitchen.Qual             10.0569        < 2.2e-16 ***
Total.Bsmt.SF            10.7689        < 2.2e-16 ***
X1st.Flr.SF               7.7064        2.358e-14 ***
Garage.Area               3.8778          0.00011 ***
Year.Built               -1.0479          0.29486    
Mas.Vnr.Area              4.4334        9.963e-06 ***
Fireplaces                1.1272          0.25983    
Heating.QC                0.7313          0.46471    
BsmtFin.SF.1              9.0566        < 2.2e-16 ***
Bsmt.Exposure             5.4897        4.731e-08 ***
Neighborhood_NridgHt     -0.3271          0.74363    
Lot.Frontage             -1.5444          0.12269    
Sale.Condition_Partial    0.4309          0.66662    
Wood.Deck.SF              0.6387          0.52312    
Neighborhood_NoRidge      0.4172          0.67659    
Lot.Shape                -1.3391          0.18075    
Tukey test               24.7871        < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```



```
> library(randtests); runs.test(model1$res)

        Runs Test

data:  model1$res
statistic = -0.87817, runs = 734, n1 = 750, n2 = 750, n = 1500, p-value = 0.3799
alternative hypothesis: nonrandomness
```



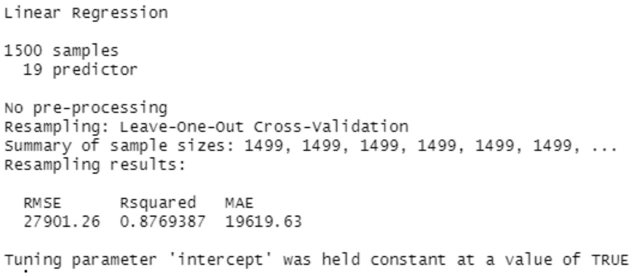
```
> library(car); durbinWatsonTest(model1)
 lag Autocorrelation D-W Statistic p-value
   1      0.04030291      1.916682   0.106
```



```
> vif(model1)
          Overall.Qual            Gr.Liv.Area             Exter.Qual           Kitchen.Qual          Total.Bsmt.SF 
              3.431183               2.323709               2.943554               2.541960               3.317780 
           X1st.Flr.SF            Garage.Area             Year.Built           Mas.Vnr.Area             Fireplaces 
              3.430452               1.755526               2.019609               1.460050               1.411073 
            Heating.QC           BsmtFin.SF.1          Bsmt.Exposure   Neighborhood_NridgHt           Lot.Frontage 
              1.538379               1.535051               1.441666               1.362937               1.334973 
Sale.Condition_Partial           Wood.Deck.SF   Neighborhood_NoRidge              Lot.Shape 
              1.284112               1.176188               1.293340               1.128784 
```

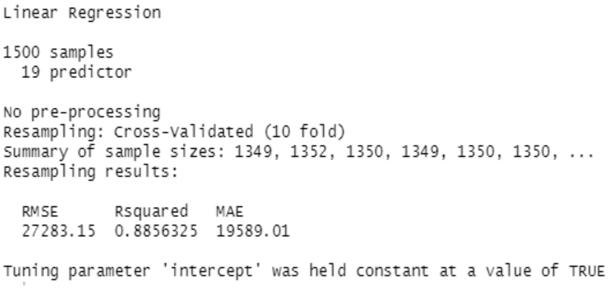
## Evaluation of Model 1

### Leave one Out



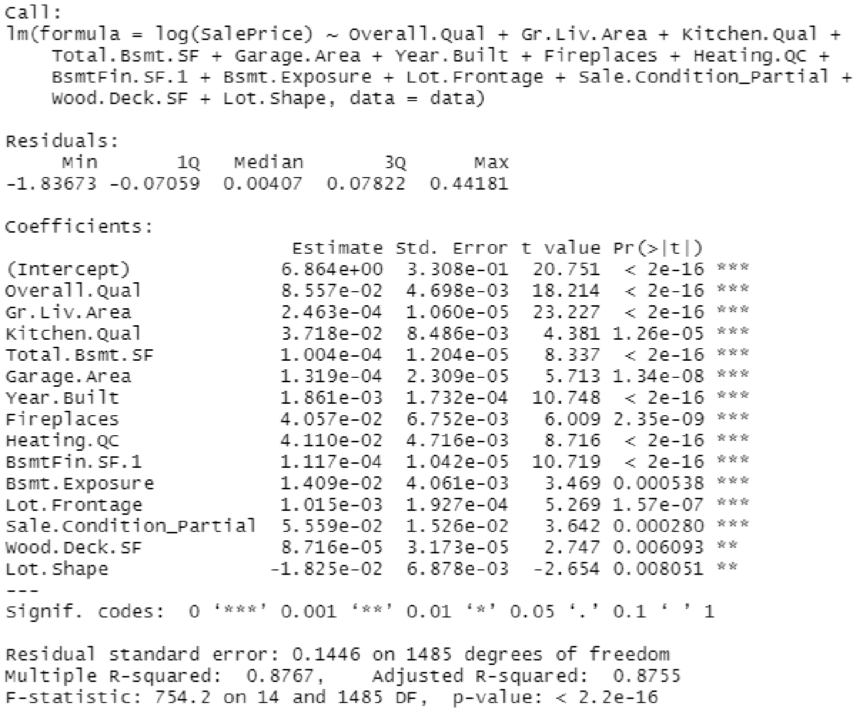
```
Linear Regression

1500 samples
  19 predictor

No pre-processing
Resampling: Leave-One-Out Cross-Validation
Summary of sample sizes: 1499, 1499, 1499, 1499, 1499, 1499, ...
Resampling results:

  RMSE      Rsquared   MAE
  27901.26  0.8769387  19619.63

Tuning parameter 'intercept' was held constant at a value of TRUE
```

### 10 – fold CV



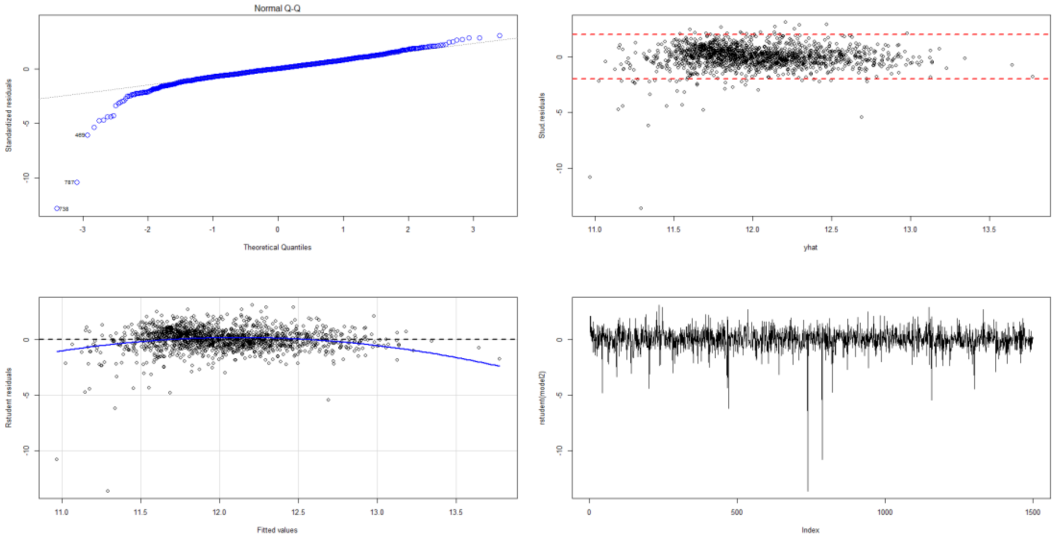
```
Linear Regression

1500 samples
  19 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 1349, 1352, 1350, 1349, 1350, 1350, ...
Resampling results:

  RMSE      Rsquared   MAE
  27283.15  0.8856325  19589.01

Tuning parameter 'intercept' was held constant at a value of TRUE
```

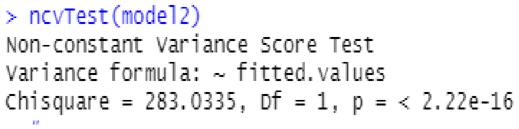
## Model 2 – Log Tranformation of SalePrice



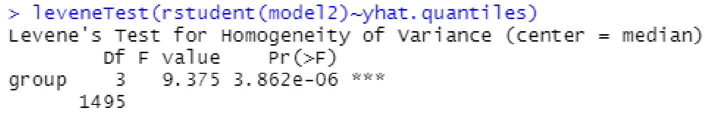
```
Call:
lm(formula = log(SalePrice) ~ Overall.Qual + Gr.Liv.Area + Kitchen.Qual +
    Total.Bsmt.SF + Garage.Area + Year.Built + Fireplaces + Heating.QC +
    BsmtFin.SF.1 + Bsmt.Exposure + Lot.Frontage + Sale.Condition_Partial +
    Wood.Deck.SF + Lot.Shape, data = data)

Residuals:
     Min       1Q   Median       3Q      Max
-1.83673 -0.07059  0.00407  0.07822  0.44181

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)            6.864e+00  3.308e-01  20.751  < 2e-16 ***
Overall.Qual           8.557e-02  4.698e-03  18.214  < 2e-16 ***
Gr.Liv.Area            2.463e-04  1.060e-05  23.227  < 2e-16 ***
Kitchen.Qual           3.718e-02  8.486e-03   4.381 1.26e-05 ***
Total.Bsmt.SF          1.004e-04  1.204e-05   8.337  < 2e-16 ***
Garage.Area            1.319e-04  2.309e-05   5.713 1.34e-08 ***
Year.Built             1.861e-03  1.732e-04  10.748  < 2e-16 ***
Fireplaces             4.057e-02  6.752e-03   6.009 2.35e-09 ***
Heating.QC             4.110e-02  4.716e-03   8.716  < 2e-16 ***
BsmtFin.SF.1           1.117e-04  1.042e-05  10.719  < 2e-16 ***
Bsmt.Exposure          1.409e-02  4.061e-03   3.469 0.000538 ***
Lot.Frontage           1.015e-03  1.927e-04   5.269 1.57e-07 ***
Sale.Condition_Partial 5.559e-02  1.526e-02   3.642 0.000280 ***
Wood.Deck.SF           8.716e-05  3.173e-05   2.747 0.006093 **
Lot.Shape             -1.825e-02  6.878e-03  -2.654 0.008051 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1446 on 1485 degrees of freedom
Multiple R-squared:  0.8767,    Adjusted R-squared:  0.8755
F-statistic: 754.2 on 14 and 1485 DF,  p-value: < 2.2e-16
```

# Residual Assumptions of Model 2

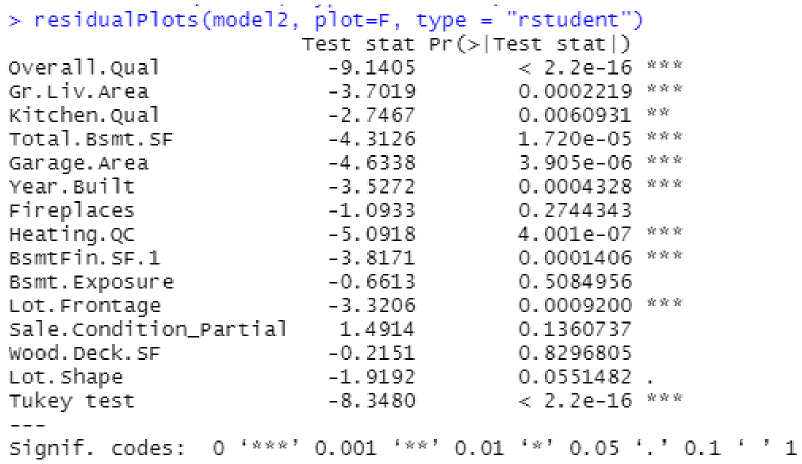




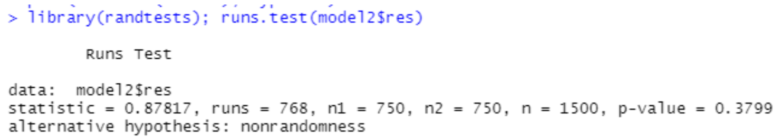
```
> ncvTest(model2)
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 283.0335, Df = 1, p = < 2.22e-16
```



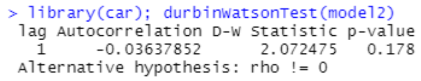
```
> leveneTest(rstudent(model2)~yhat.quantiles)
Levene's Test for Homogeneity of Variance (center = median)
        Df F value    Pr(>F)
group    3   9.375 3.862e-06 ***
      1495
```



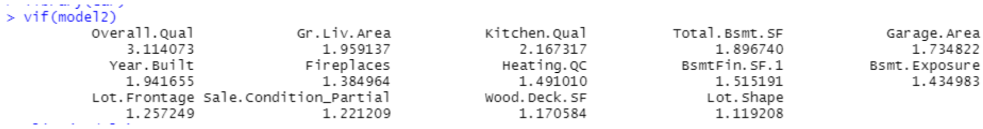
```
> residualPlots(model2, plot=F, type = "rstudent")
                         Test stat Pr(>|Test stat|)
Overall.Qual                -9.1405        < 2.2e-16 ***
Gr.Liv.Area                 -3.7019        0.0002219 ***
Kitchen.Qual                -2.7467        0.0060931 **
Total.Bsmt.SF               -4.3126        1.720e-05 ***
Garage.Area                 -4.6338        3.905e-06 ***
Year.Built                  -3.5272        0.0004328 ***
Fireplaces                  -1.0933        0.2744343
Heating.QC                  -5.0918        4.001e-07 ***
BsmtFin.SF.1                -3.8171        0.0001406 ***
Bsmt.Exposure               -0.6613        0.5084956
Lot.Frontage                -3.3206        0.0009200 ***
Sale.Condition_Partial       1.4914        0.1360737
Wood.Deck.SF                -0.2151        0.8296805
Lot.Shape                   -1.9192        0.0551482 .
Tukey test                  -8.3480        < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```



```
> library(randtests); runs.test(model2$res)

	Runs Test

data:  model2$res
statistic = 0.87817, runs = 768, n1 = 750, n2 = 750, n = 1500, p-value = 0.3799
alternative hypothesis: nonrandomness
```



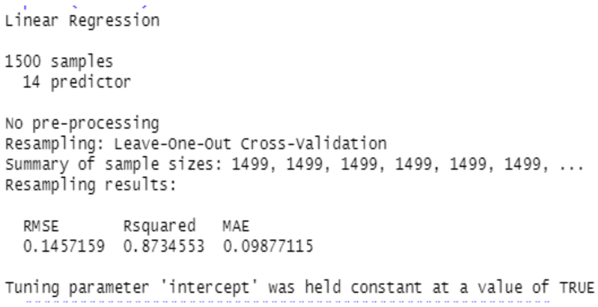
```
> library(car); durbinWatsonTest(model2)
 lag Autocorrelation D-W Statistic p-value
   1     -0.03637852      2.072475   0.178
 Alternative hypothesis: rho != 0
```



```
> vif(model2)
          Overall.Qual            Gr.Liv.Area           Kitchen.Qual          Total.Bsmt.SF            Garage.Area
              3.114073               1.959137               2.167317               1.896740               1.734822
            Year.Built             Fireplaces             Heating.QC           BsmtFin.SF.1          Bsmt.Exposure
              1.941655               1.384964               1.491010               1.515191               1.434983
          Lot.Frontage Sale.Condition_Partial           Wood.Deck.SF              Lot.Shape
              1.257249               1.221209               1.170584               1.119208
```

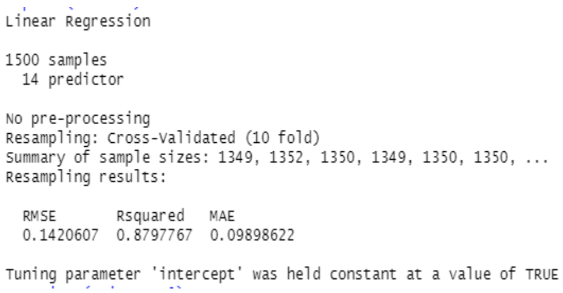
## Evaluation of Model 2

### Leave one Out



```
Linear Regression

1500 samples
  14 predictor

No pre-processing
Resampling: Leave-One-Out Cross-Validation
Summary of sample sizes: 1499, 1499, 1499, 1499, 1499, 1499, ...
Resampling results:

  RMSE       Rsquared   MAE
  0.1457159  0.8734553  0.09877115

Tuning parameter 'intercept' was held constant at a value of TRUE
```

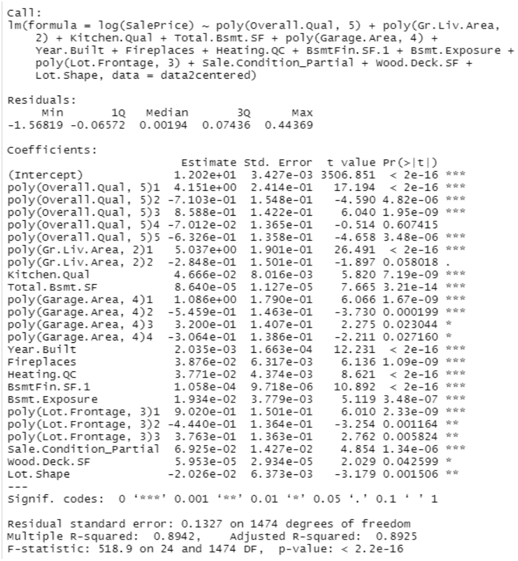
### 10 – fold CV



```
Linear Regression

1500 samples
  14 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 1349, 1352, 1350, 1349, 1350, 1350, ...
Resampling results:

  RMSE       Rsquared   MAE
  0.1420607  0.8797767  0.09898622

Tuning parameter 'intercept' was held constant at a value of TRUE
```

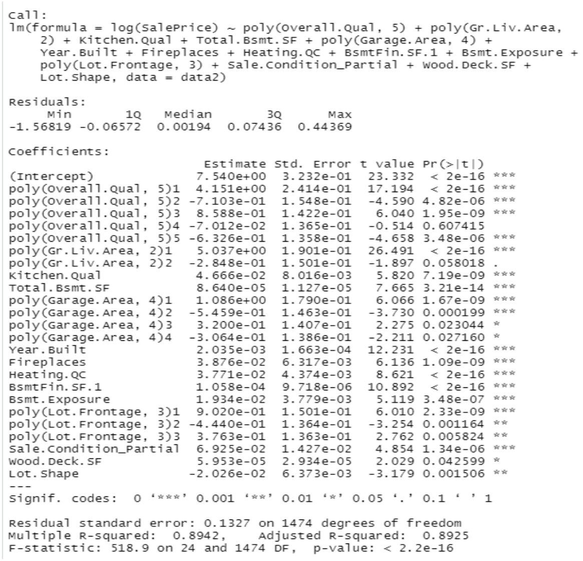
### Estimation of Model 3

### Log and Polynomial Transformation



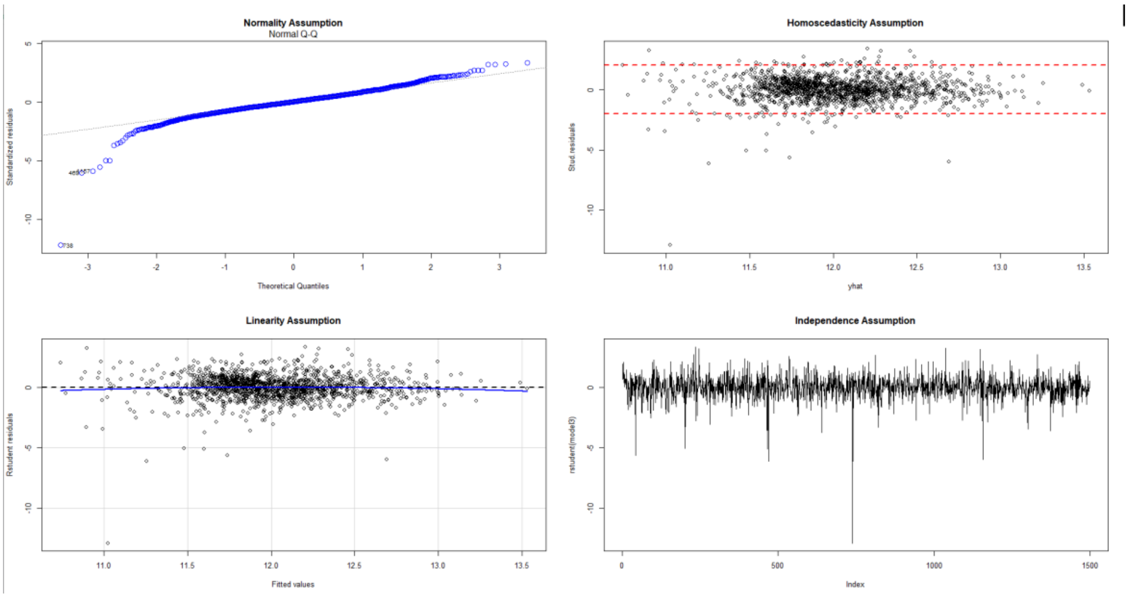
```
Call:
lm(formula = log(SalePrice) ~ poly(Overall.Qual, 5) + poly(Gr.Liv.Area,
    2) + Kitchen.Qual + Total.Bsmt.SF + poly(Garage.Area, 4) +
    Year.Built + Fireplaces + Heating.QC + BsmtFin.SF.1 + Bsmt.Exposure +
    poly(Lot.Frontage, 3) + Sale.Condition_Partial + Wood.Deck.SF +
    Lot.Shape, data = data2centered)

Residuals:
     Min       1Q   Median       3Q      Max
-1.56819 -0.06572  0.00194  0.07436  0.44369

Coefficients:
                        Estimate Std. Error  t value Pr(>|t|)
(Intercept)            1.202e+01  3.427e-03 3506.851  < 2e-16 ***
poly(Overall.Qual, 5)1  4.151e+00  2.414e-01   17.194  < 2e-16 ***
poly(Overall.Qual, 5)2 -7.103e-01  1.548e-01   -4.590 4.82e-06 ***
poly(Overall.Qual, 5)3  8.588e-01  1.422e-01    6.040 1.95e-09 ***
poly(Overall.Qual, 5)4 -7.012e-02  1.365e-01   -0.514 0.607415
poly(Overall.Qual, 5)5 -6.326e-01  1.358e-01   -4.658 3.48e-06 ***
poly(Gr.Liv.Area, 2)1   5.037e+00  1.901e-01   26.491  < 2e-16 ***
poly(Gr.Liv.Area, 2)2  -2.848e-01  1.501e-01   -1.897 0.058018 .
Kitchen.Qual            4.666e-02  8.016e-03    5.820 7.19e-09 ***
Total.Bsmt.SF           8.640e-05  1.127e-05    7.665 3.21e-14 ***
poly(Garage.Area, 4)1   1.086e+00  1.790e-01    6.066 1.67e-09 ***
poly(Garage.Area, 4)2  -5.459e-01  1.463e-01   -3.730 0.000199 ***
poly(Garage.Area, 4)3   3.200e-01  1.407e-01    2.275 0.023044 *
poly(Garage.Area, 4)4  -3.064e-01  1.386e-01   -2.211 0.027160 *
Year.Built              2.035e-03  1.663e-04   12.231  < 2e-16 ***
Fireplaces              3.876e-02  6.317e-03    6.136 1.09e-09 ***
Heating.QC              3.771e-02  4.374e-03    8.621  < 2e-16 ***
BsmtFin.SF.1            1.058e-04  9.718e-06   10.892  < 2e-16 ***
Bsmt.Exposure           1.934e-02  3.779e-03    5.119 3.48e-07 ***
poly(Lot.Frontage, 3)1  9.020e-01  1.501e-01    6.010 2.33e-09 ***
poly(Lot.Frontage, 3)2 -4.440e-01  1.364e-01   -3.254 0.001164 **
poly(Lot.Frontage, 3)3  3.763e-01  1.363e-01    2.762 0.005824 **
Sale.Condition_Partial  6.925e-02  1.427e-02    4.854 1.34e-06 ***
Wood.Deck.SF            5.953e-05  2.934e-05    2.029 0.042599 *
Lot.Shape              -2.026e-02  6.373e-03   -3.179 0.001506 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1327 on 1474 degrees of freedom
Multiple R-squared:  0.8942,	Adjusted R-squared:  0.8925
F-statistic: 518.9 on 24 and 1474 DF,  p-value: < 2.2e-16
```

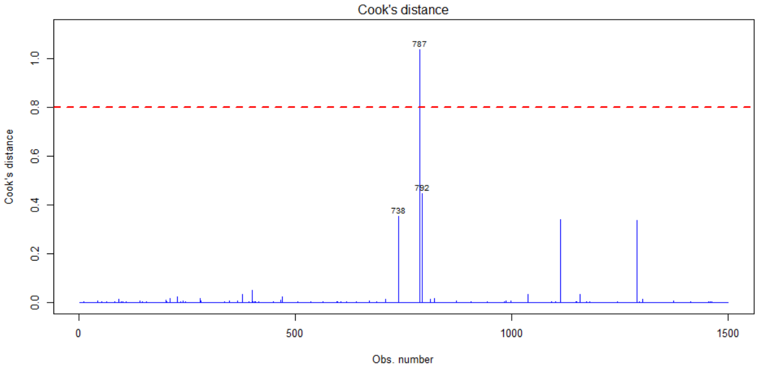
### Model3 with centred covariates



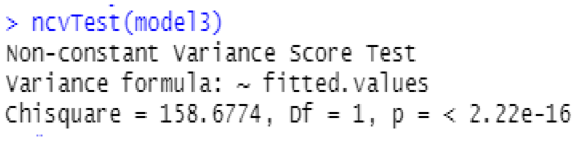
```
Call:
lm(formula = log(SalePrice) ~ poly(Overall.Qual, 5) + poly(Gr.Liv.Area,
    2) + Kitchen.Qual + Total.Bsmt.SF + poly(Garage.Area, 4) +
    Year.Built + Fireplaces + Heating.QC + BsmtFin.SF.1 + Bsmt.Exposure +
    poly(Lot.Frontage, 3) + Sale.Condition_Partial + Wood.Deck.SF +
    Lot.Shape, data = data2)

Residuals:
     Min       1Q   Median       3Q      Max
-1.56819 -0.06572  0.00194  0.07436  0.44369

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)             7.540e+00  3.232e-01  23.332  < 2e-16 ***
poly(Overall.Qual, 5)1  4.151e+00  2.414e-01  17.194  < 2e-16 ***
poly(Overall.Qual, 5)2 -7.103e-01  1.548e-01  -4.590 4.82e-06 ***
poly(Overall.Qual, 5)3  8.588e-01  1.422e-01   6.040 1.95e-09 ***
poly(Overall.Qual, 5)4 -7.012e-02  1.365e-01  -0.514 0.607415
poly(Overall.Qual, 5)5 -6.326e-01  1.358e-01  -4.658 3.48e-06 ***
poly(Gr.Liv.Area, 2)1   5.037e+00  1.901e-01  26.491  < 2e-16 ***
poly(Gr.Liv.Area, 2)2  -2.848e-01  1.501e-01  -1.897 0.058018 .
Kitchen.Qual            4.666e-02  8.016e-03   5.820 7.19e-09 ***
Total.Bsmt.SF           8.640e-05  1.127e-05   7.665 3.21e-14 ***
poly(Garage.Area, 4)1   1.086e+00  1.790e-01   6.066 1.67e-09 ***
poly(Garage.Area, 4)2  -5.459e-01  1.463e-01  -3.730 0.000199 ***
poly(Garage.Area, 4)3   3.200e-01  1.407e-01   2.275 0.023044 *
poly(Garage.Area, 4)4  -3.064e-01  1.386e-01  -2.211 0.027160 *
Year.Built              2.035e-03  1.663e-04  12.231  < 2e-16 ***
Fireplaces              3.876e-02  6.317e-03   6.136 1.09e-09 ***
Heating.QC              3.771e-02  4.374e-03   8.621  < 2e-16 ***
BsmtFin.SF.1            1.058e-04  9.718e-06  10.892  < 2e-16 ***
Bsmt.Exposure           1.934e-02  3.779e-03   5.119 3.48e-07 ***
poly(Lot.Frontage, 3)1  9.020e-01  1.501e-01   6.010 2.33e-09 ***
poly(Lot.Frontage, 3)2 -4.440e-01  1.364e-01  -3.254 0.001164 **
poly(Lot.Frontage, 3)3  3.763e-01  1.363e-01   2.762 0.005824 **
Sale.Condition_Partial  6.925e-02  1.427e-02   4.854 1.34e-06 ***
Wood.Deck.SF            5.953e-05  2.934e-05   2.029 0.042599 *
Lot.Shape              -2.026e-02  6.373e-03  -3.179 0.001506 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1327 on 1474 degrees of freedom
Multiple R-squared:  0.8942,	Adjusted R-squared:  0.8925
F-statistic: 518.9 on 24 and 1474 DF,  p-value: < 2.2e-16
```

# Residual Assumptions of Model 3

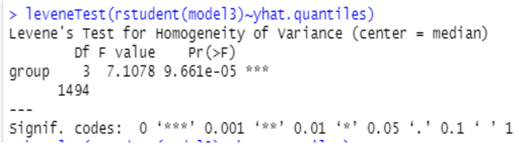




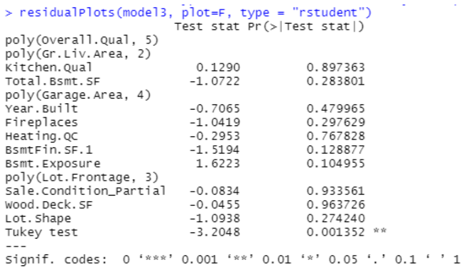
Cook’s distance plot indicates an outlier (787) as influential point which must be removed.



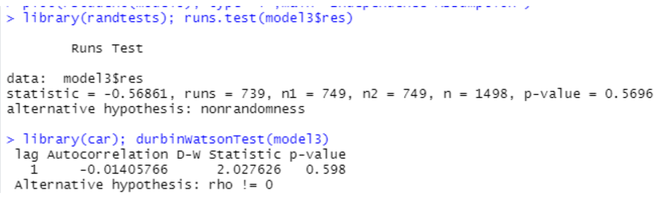
```
> ncvTest(model3)
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 158.6774, Df = 1, p = < 2.22e-16
```



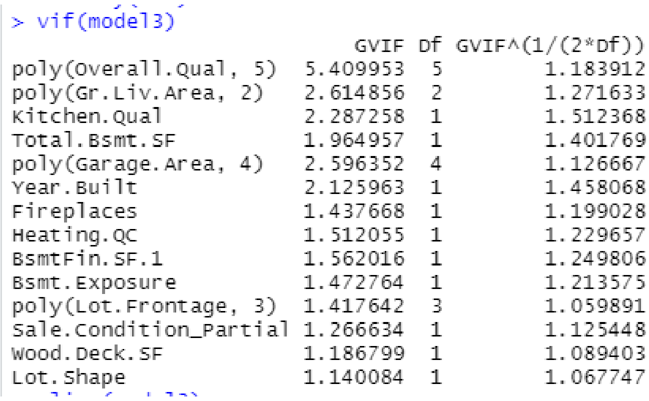
```
> leveneTest(rstudent(model3)~yhat.quantiles)
Levene's Test for Homogeneity of Variance (center = median)
        Df F value    Pr(>F)
group    3  7.1078 9.661e-05 ***
      1494
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```



```
> residualPlots(model3, plot=F, type = "rstudent")
                        Test stat Pr(>|Test stat|)
poly(Overall.Qual, 5)
poly(Gr.Liv.Area, 2)
Kitchen.Qual               0.1290         0.897363
Total.Bsmt.SF             -1.0722         0.283801
poly(Garage.Area, 4)
Year.Built                -0.7065         0.479965
Fireplaces                -1.0419         0.297629
Heating.QC                -0.2953         0.767828
BsmtFin.SF.1              -1.5194         0.128877
Bsmt.Exposure              1.6223         0.104955
poly(Lot.Frontage, 3)
Sale.Condition_Partial    -0.0834         0.933561
Wood.Deck.SF              -0.0455         0.963726
Lot.Shape                 -1.0938         0.274240
Tukey test                -3.2048         0.001352 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```



```
> library(randtests); runs.test(model3$res)

        Runs Test

data:  model3$res
statistic = -0.56861, runs = 739, n1 = 749, n2 = 749, n = 1498, p-value = 0.5696
alternative hypothesis: nonrandomness

> library(car); durbinWatsonTest(model3)
 lag Autocorrelation D-W Statistic p-value
   1     -0.01405766      2.027626   0.598
 Alternative hypothesis: rho != 0
```

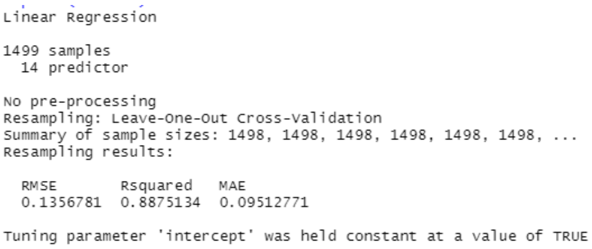


```
> vif(model3)
                           GVIF Df GVIF^(1/(2*Df))
poly(Overall.Qual, 5)  5.409953  5        1.183912
poly(Gr.Liv.Area, 2)   2.614856  2        1.271633
Kitchen.Qual           2.287258  1        1.512368
Total.Bsmt.SF          1.964957  1        1.401769
poly(Garage.Area, 4)   2.596352  4        1.126667
Year.Built             2.125963  1        1.458068
Fireplaces             1.437668  1        1.199028
Heating.QC             1.512055  1        1.229657
BsmtFin.SF.1           1.562016  1        1.249806
Bsmt.Exposure          1.472764  1        1.213575
poly(Lot.Frontage, 3)  1.417642  3        1.059891
Sale.Condition_Partial 1.266634  1        1.125448
Wood.Deck.SF           1.186799  1        1.089403
Lot.Shape              1.140084  1        1.067747
```

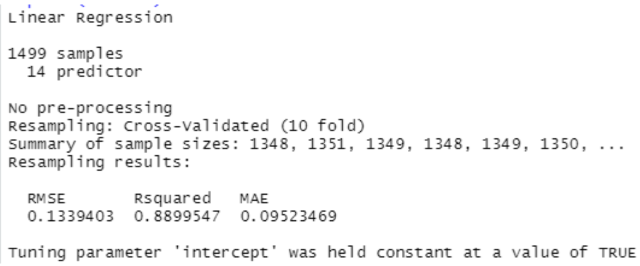
## Evaluation of Model 3

### Leave one Out

### 10 – fold CV



```
Linear Regression

1499 samples
  14 predictor

No pre-processing
Resampling: Leave-One-Out Cross-Validation
Summary of sample sizes: 1498, 1498, 1498, 1498, 1498, 1498, ...
Resampling results:

  RMSE       Rsquared   MAE
  0.1356781  0.8875134  0.09512771

Tuning parameter 'intercept' was held constant at a value of TRUE
```



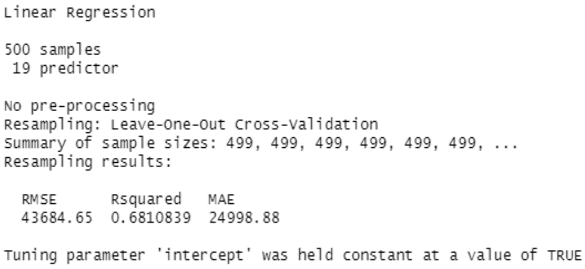
```
Linear Regression

1499 samples
  14 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 1348, 1351, 1349, 1348, 1349, 1350, ...
Resampling results:

  RMSE       Rsquared   MAE
  0.1339403  0.8899547  0.09523469

Tuning parameter 'intercept' was held constant at a value of TRUE
```

## Evaluation of all Models with the test dataset

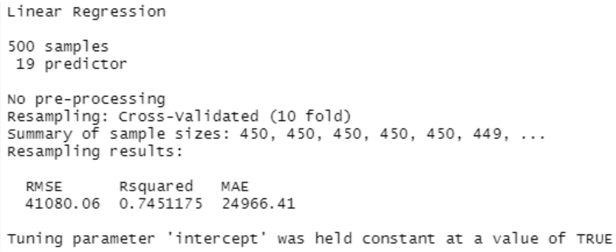
## Model 1

### Leave one Out

### 10 – fold CV



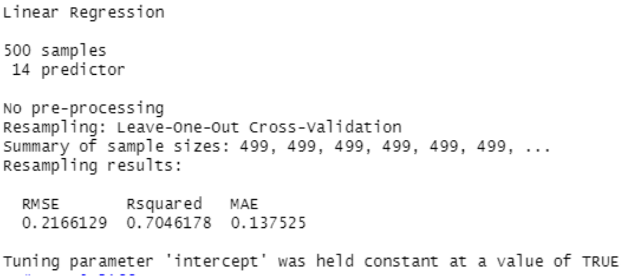
```
Linear Regression

500 samples
 19 predictor

No pre-processing
Resampling: Leave-One-Out Cross-Validation
Summary of sample sizes: 499, 499, 499, 499, 499, 499, ...
Resampling results:

  RMSE      Rsquared   MAE
  43684.65  0.6810839  24998.88

Tuning parameter 'intercept' was held constant at a value of TRUE
```



```
Linear Regression

500 samples
 19 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 450, 450, 450, 450, 450, 449, ...
Resampling results:

  RMSE      Rsquared   MAE
  41080.06  0.7451175  24966.41

Tuning parameter 'intercept' was held constant at a value of TRUE
```

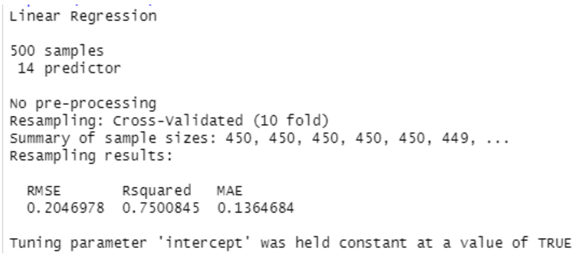
# Model 2

## Leave one Out



```
Linear Regression

500 samples
 14 predictor

No pre-processing
Resampling: Leave-One-Out Cross-Validation
Summary of sample sizes: 499, 499, 499, 499, 499, 499, ...
Resampling results:

  RMSE       Rsquared   MAE
  0.2166129  0.7046178  0.137525

Tuning parameter 'intercept' was held constant at a value of TRUE
```

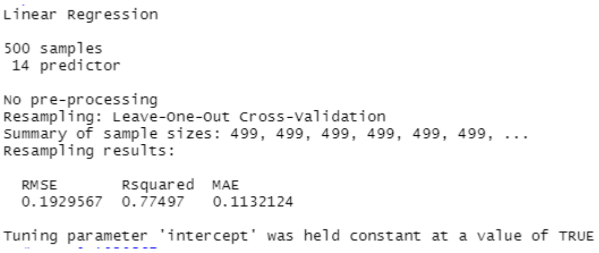
## 10 – fold CV



```
Linear Regression

500 samples
 14 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 450, 450, 450, 450, 450, 449, ...
Resampling results:

  RMSE       Rsquared   MAE
  0.2046978  0.7500845  0.1364684

Tuning parameter 'intercept' was held constant at a value of TRUE
```

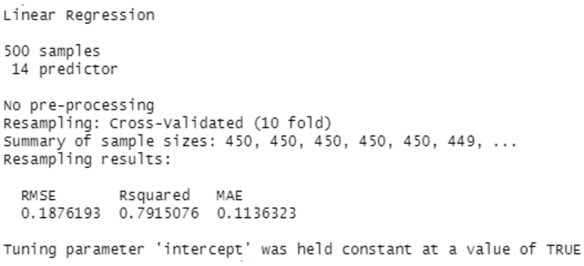
# Model 3

## Leave one Out



```
Linear Regression

500 samples
 14 predictor

No pre-processing
Resampling: Leave-One-Out Cross-Validation
Summary of sample sizes: 499, 499, 499, 499, 499, 499, ...
Resampling results:

  RMSE       Rsquared  MAE
  0.1929567  0.77497   0.1132124

Tuning parameter 'intercept' was held constant at a value of TRUE
```

## 10 – fold CV



```
Linear Regression

500 samples
 14 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 450, 450, 450, 450, 450, 449, ...
Resampling results:

  RMSE       Rsquared   MAE
  0.1876193  0.7915076  0.1136323

Tuning parameter 'intercept' was held constant at a value of TRUE
```