STATISTICS FOR BUSINESS ANALYTICS I
Main Assignment 2020-2021

Vretteas Stylianos
Student Id: p2822003
Number of Dataset: 60
Professor: Ntzoufras Ioannis

Athens University of Economic and Business
MSc In Business Analytics

<u>Contents:</u>

Contents of figures and tables:

Section 1 - Introduction

Ames Iowa Housing dataset – Description

The data of this assignment is a product of the Ames Iowa Housing dataset created from Dean DeCock for use in statistics education. A mixed of 82 nominal, ordinal, continuous and discrete variables are being used in order to describe every aspect of a house sale in Ames Iowa. Most of them are exactly the type of information a buyer would like to know before the purchase of a home (e.g. When the house was built, how big is the lot? How many cars the garage can fit? How many bathrooms the house has?).With this information, we will try to build a predictive model in order to predict future house prices in Ames Iowa.

First of all, we are going to have a look at the structure of the dataset, understand the variables, and make some transformations in order to continue in the model building. A test dataset was given also. We are going to make the same transformations in the test dataset in order to check our model at the end.Since, the train dataset consists of character and integer data type variables, we are going to convert them into numeric and factors respectively. More specifically for the ordinal variables which are in character data type, there is going to be a revaluation of them into ordinal numeric vectors where there is a clear ranking. For the missing values, I will explain later the imputation methods since there were a lot of missing data in the train and the test datasets.

Furthermore, there is going to be a presentation of the most important variables and their association with the variable in question Sales Price and the various predictor variables. Also, a correlation matrix will be presented in order to see the most correlated variables with price alongside the correlation between them.

After this part of descriptive statistics, I created a data frame with both numeric and categorical factors (dummies) in order to insert it into the variable selection algorithms.
There will be a very short briefing of these algorithms (LASSO and AIC) and then I will describe the model building procedures the transformations training -testing and the methods that were used for this.

Finally, in conclusion there will be a short summary of the final model and its accuracy of predictions alongside some suggestions for future engineering and model improving.

Section 2 – Data Cleansing

First of all, we have to check the structure of the train and test dataset that was given to us. It seems there a lot of missing values in both sets. For saving space, the below visualizations will focus only on the train dataset. Nevertheless, there were strong similarities in the missing data and every change in the training set was made simultaneously on the test dataset too. Below are images that describe the missing data on the train dataset. ***(See appendix figure2)***

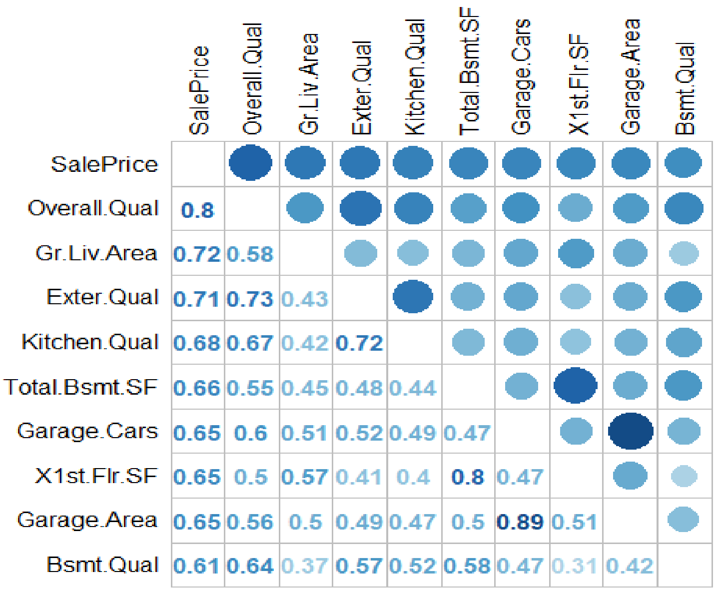
Figure 1 – Missing data Totals

```
> sort(colSums(sapply(train_60[na_values], is.na)), decreasing = TRUE)#
       Pool.QC   Misc.Feature          Alley          Fence   Fireplace.Qu    Lot.Frontage    Garage.Type
          1492           1450           1411           1225            724             235             79
 Garage.Yr.Blt  Garage.Finish    Garage.Qual    Garage.Cond  Bsmt.Exposure       Bsmt.Qual       Bsmt.Cond
            79             79             79             79             43              41              41
BsmtFin.Type.1 BsmtFin.Type.2  Mas.Vnr.Type   Mas.Vnr.Area     Electrical
            41             41             14             14              1
> |
```

From the above graphs, we see the missing data percentage and the totals. NA values for the majority of the variables do not represent a missing observation but the absence of the feature in question, that is why the value “None” was inserted in these observations. Regarding the Lot.Frontage variable, I inserted the median per neighborhood in the missing data as it seemed the better imputation against the alternative of the mean. For Garage.Yr.Blt, the value 0 was inserted and incicates that there is no garage at all in these observations. For variable Electrical, there was only one missing value and the mode was selected for filling the missing point. Specifically, for the Garage variables there were 79 observations without value and for the basement variables there were a total of 41.

Section 3 – Descriptive and Exploratory Data Analysis

Figure 3 – Correlation matrix with Price

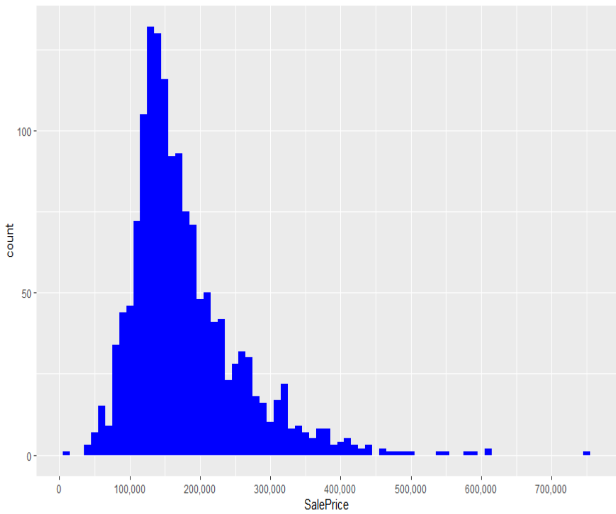


This correlation matrix indicates the numeric variables that there are most correlated with the response variable Sales Price, thus they are the most “important” variables. The overall quality is the most correlated with price as it stands with a 0.8 Pearson correlation rate, following is the above ground live area and the Exterior Quality with 0.72 and 0.71 Pearson correlation rates respectively. This means a positive relationship with the variable price.

See appendix figure 6

The below graph is a histogram for our response variable Sale Price. There is a clear skewness, but this is quite logical due to the fact that people’s income is not sufficient for everyone to buy the expensive houses.

Fig. 4 – Sale Price



From the first look, the distribution of the Sale Price variable seems not to following a normal distribution. This is examined further with QQ plot and normality tests.

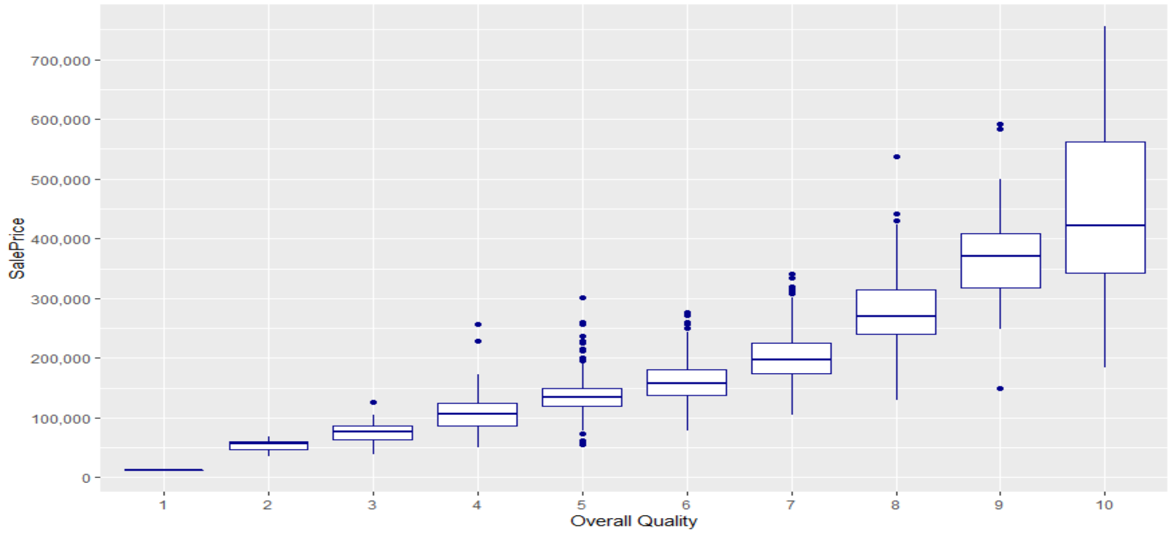
After the tests, we can confirm that there is no normality in the Sales Price variable.
For qq-plots and normality tests see appendix ***Figure5 and Table 1***

Summary of Sale Price

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  13100  128863  158950  179277  213000  755000
```

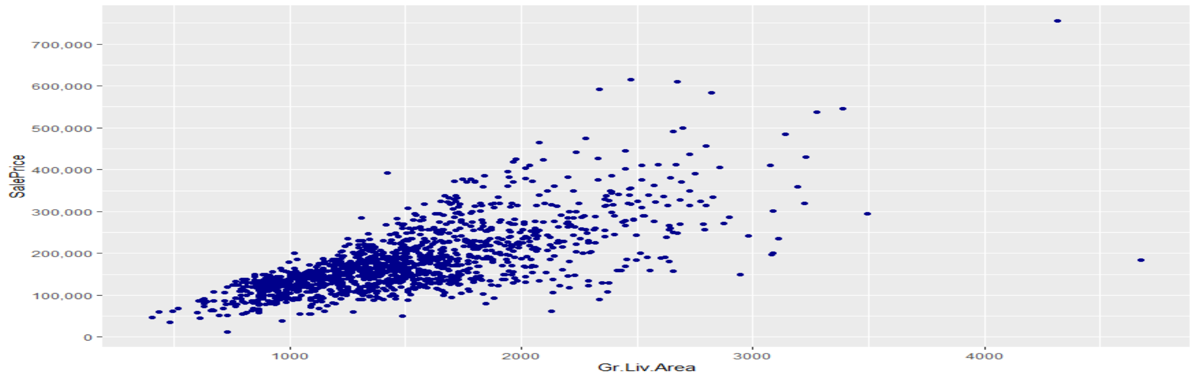
There is a strong correlation with Sales Price (above 0.70) for the variables, Overall. Qual, Gr.Liv.Area, Exter.Qual . The below graphs support this statement.

Fig. 6- Overall Quality



Overall Quality is an ordinal variable, it rates the overall material and finish of the house (variable description) thus as the rating is higher so does the price of the house.

Fig. 7 - Gr.Liv.Area

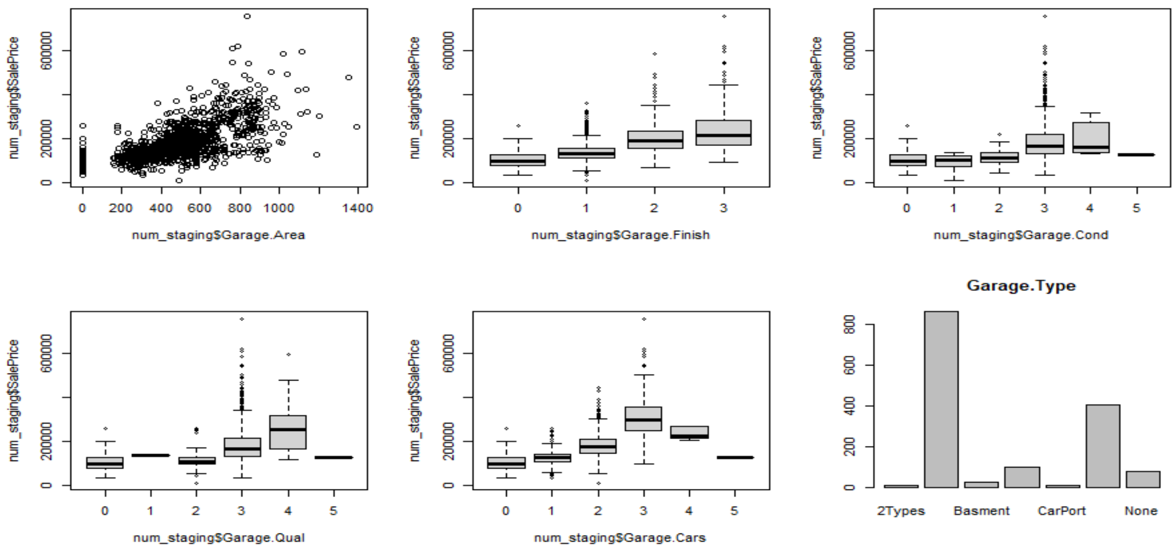


The second most correlated variable is the Above grade (ground) living area in square feet. Above there is a scatterplot to see the dispersion of the data. The bigger area a home has the price goes up and the dispersion is thinner especially for houses above 2500 sq.feet. There are also two houses with extremely high Gr.Liv.Area and we probably deal with them later. Normality tests for the above variables are failing and we cannot say that the data come from a normal distribution. For QQ plots and tests, ***see appendix figure 5 and table 1***

For more descriptive statistics of numeric variables, in order to understand better their distribution (histograms), ***see appendix figures 8 – 9***

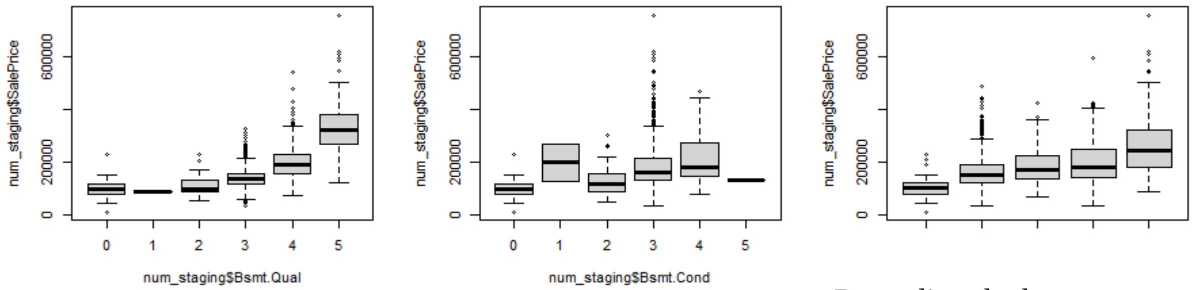
About the **garage** and **basement** features, there are a lot of variables that describe every aspect of them. Below, there are some presentations for garage and basement in order to understand better these features. In order to deal with the multicollinearity effect early on before the modeling, we are going to drop off some of these variables and keep the most correlated with price.

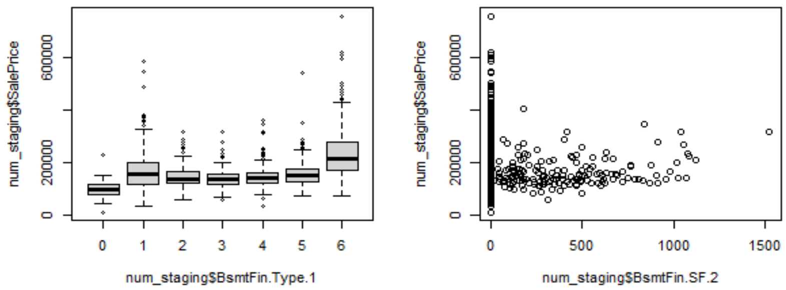
Fig. 8- Garage Variables



From the above, we can quite understand homes with garages which can fit 3 cars are more expensive than the others alongside homes with garages of quality ranking 4, also the garage type with the “Attached “type is the most frequent.

Fig. 9 - Basement variables





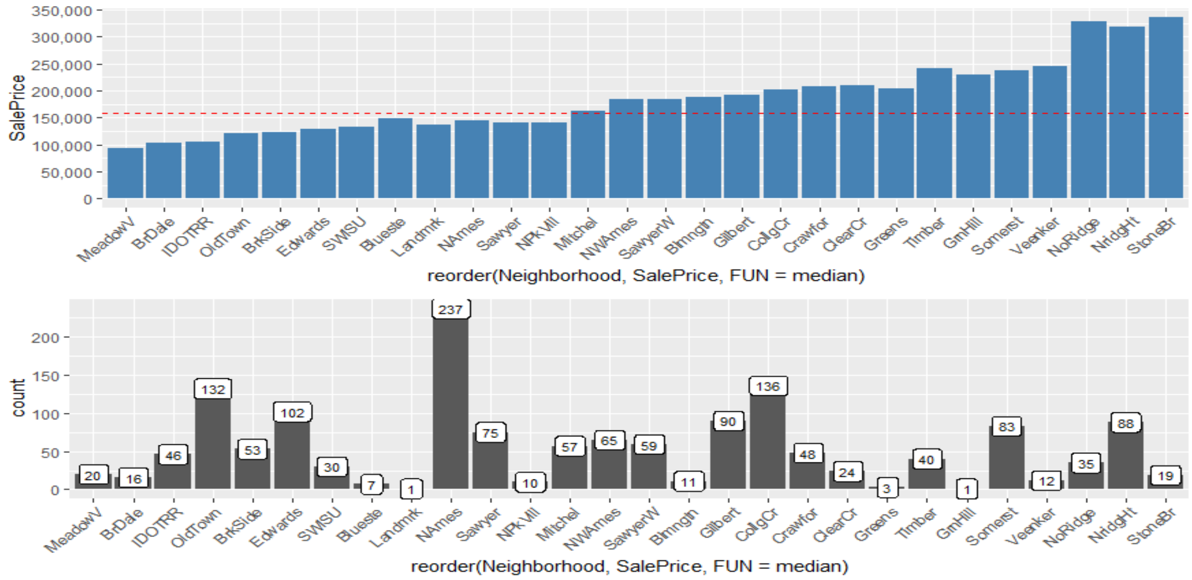
Regarding the basement variables, we can understand from these plots that homes with basement condition level 1 are the most expensive and the with exposure, quality level 4 and 5 respectively.

For both features, there is a quality and condition variables. Especially for the basement, the quality attribute evaluates the height of the basement. There is also the basement exposure which refers to walkout on garden level walls. The rest are quite clear. Number of cars a garage can feet, year the garage was built and space variables in square feet for both features.

Regarding the categorical variables.

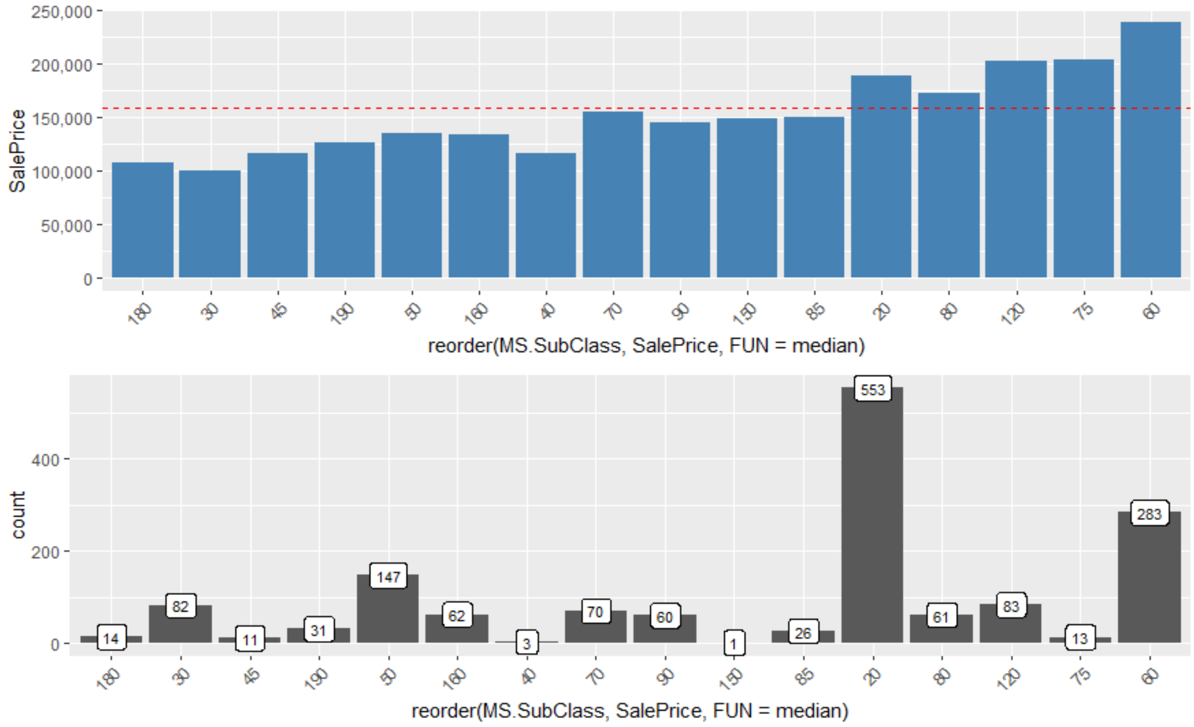
I will present first the two variables with the most factors. These are the Neighborhood variable which refers to physical locations within Ames City limits and the other is the MS SubClass variable which identifies the type of dwelling involved in the sale. Later on, I decide to bin further the Neighborhood and split it to Poor – Typical – Rich in order to reduce the number of factors. I did not do this with the MsSubClass because there was not a clear pattern of the attributes with the Sales Price.

Fig. 10 - Neighborhood



The above graphs provide information about the Sale Price per neighborhood. It is clear that we have three “rich” neighborhoods, No Ridge, Northern Heights and Stone Brook and three “poor” neighborhoods Meadow Village, Briardale and Iowa DOT and Rail Road. The red line indicates the median of the Sales Price. Most of the neighborhoods are above the median Sales Price. The second graphs show the frequencies of observations per Neighborhood thus we understand that the majority of the homes are located on the North Ames neighborhood (237), while Old Town (132) and College Creek (136) are following.

Fig. 11- MS Sub Class



Regarding Categorical Variable - MS Sub-Class, the variable appears as numeric first but there are clear factors. Mapping for the factors is presented below.

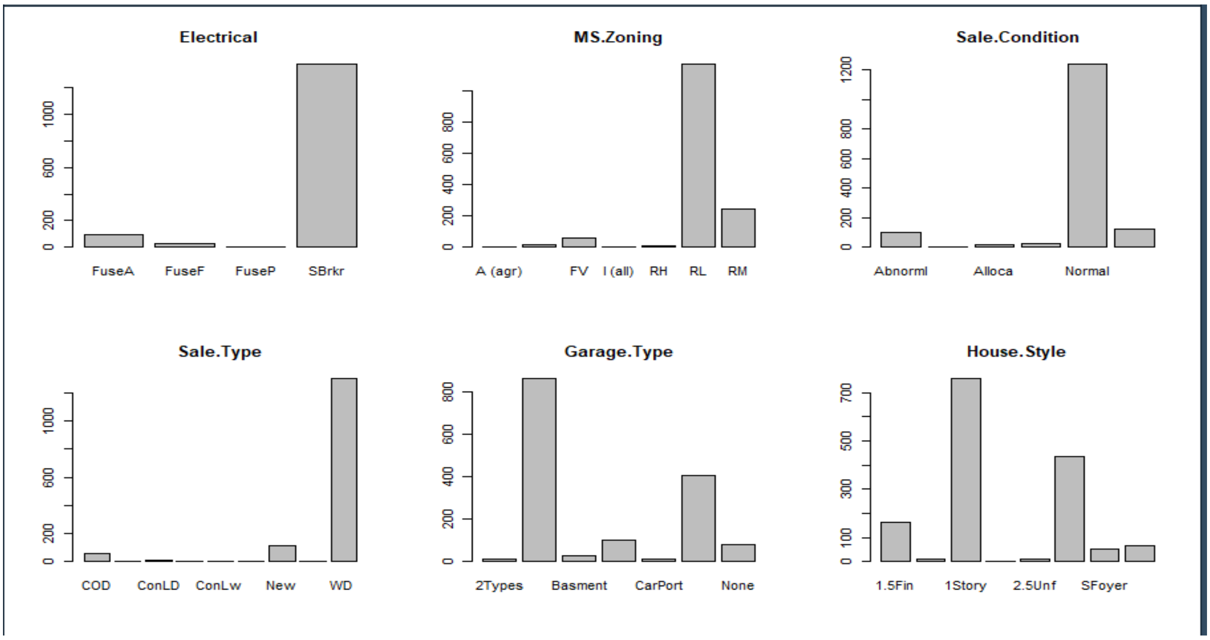
20	1-STORY 1946 & NEWER ALL STYLES	80	SPLIT OR MULTI-LEVEL
30	1-STORY 1945 & OLDER	85	SPLIT FOYER
40	1-STORY W/FINISHED ATTIC ALL AGES	90	DUPLEX - ALL STYLES AND AGES
45	37257 STORY - UNFINISHED ALL AGES	120	1-STORY PUD (Planned Unit Development) -
50	37257 STORY FINISHED ALL AGES	150	37257 STORY PUD - ALL AGES
60	2-STORY 1946 & NEWER	160	2-STORY PUD - 1946 & NEWER
70	2-STORY 1945 & OLDER	180	PUD - MULTILEVEL - INCL SPLIT
75	37258 STORY ALL AGES	190	2 FAMILY CONVERSION - ALL STYLES

It seems that the most “expensive Sub-Class is the “SPLIT OR MULTI-LEVEL” and the “37258 STORY ALL AGES “while the cheaper ones are the PUD – MULTILEVEL – INCL SPLIT and the 1-STORY 1945 & OLDER.
Even though there is a classification, 1-story houses – 2 story houses and other, I decided not to bin further this variable because there is not clear pattern regarding the Sales Price. Regarding the frequencies table, it is clear that the “SPLIT OR MULTI-LEVEL “andthe “1-STORY 1946 & NEWER ALL STYLES” are the most frequent.

Before moving into variable selection algorithms, I decided to drop a significant number of variables due to the fact that their frequency was above 99%. These were the following variables, Street, Utilities, Condition.2, Roof.Matl, Heating. ***See appendix table 3***

The rest of the categorical variables provide information regarding various aspects of property such as MS-Zoning that identifies the general zoning classification, Electrical which describes the Electrical system of the property, Sale type which describes the method of purchase (warranty, contract, home just constructed and sold) Sale condition. Below bar plots for the above-mentioned categorical variables.

Fig. 12- Barplots for categorical



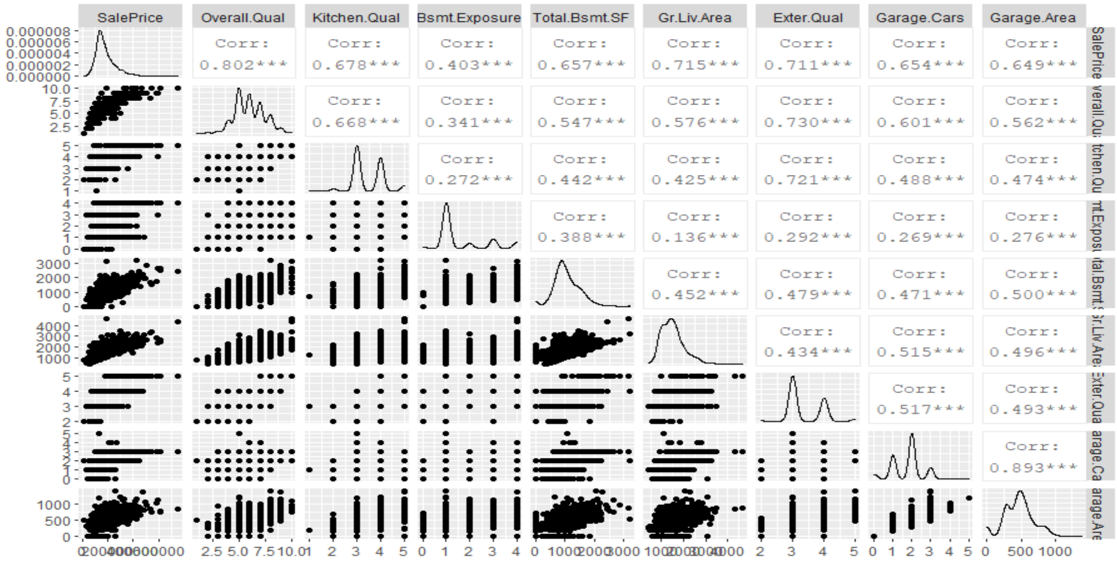
Finally, above there are some simple barplots in order to understand typical characteristics of a property purchase and the property. The frequencies of the above indicated that almost the majority of the homes has Electrical system of type “SBrkr”, the MsZoning area is labeled as RL which stands for Residential Low density. The sale condition is of type normal and the Sale Type is WD (warranty) which indicate a typical transaction and at last the majority of the houses is labeld as 1 or 2 story houses with garage.

Section 4 – Pairwise Comparisons

This section will focus on pairwise comparisons between some variables.

Below is some comparisons between the Sale Price, Overall Quality, Kitchen Quality, Basement Exposure, Gr.Livving Area, Exterior Quality, Garage cars and Garage area scatter plots and the correlations between them.

table 2 – correlations and scatterplots



Furthermore, I will examine further the pair correlations between all the numeric variables as seen in the below table.

Table 3– top pair correlations

row	column	cor	p
Garage.Yr.Blt	Garage.Qual	0.9463992	0
Garage.Yr.Blt	Garage.Cond	0.9455517	0
Garage.Qual	Garage.Cond	0.9367572	0
Garage.Cars	Garage.Area	0.8934135	0
Pool.Area	Pool.QC	0.8719930	0
Fireplaces	Fireplace.Qu	0.8589486	0
Gr.Liv.Area	TotRms.AbvGrd	0.8177860	0
Total.Bsmt.SF	X1st.Flr.SF	0.8049797	0
Overall.Qual	SalePrice	0.8023188	0
BsmtFin.Type.2	BsmtFin.SF.2	0.7885772	0
BsmtFin.Type.1	BsmtFin.SF.1	0.7334622	0
Overall.Qual	Exter.Qual	0.7300369	0
Exter.Qual	Kitchen.Qual	0.7212810	0
Gr.Liv.Area	SalePrice	0.7151540	0
Exter.Qual	SalePrice	0.7112551	0

After filtering the correlations table for all the pairs of variables, these are the most correlated. In order to deal with the multicollinearity effect early on I decide to drop some of them. Especially for the garage and the basement features I keep in my dataset the most correlated variable with the response Sale Price. There is also a strong correlation between some quality variables such as the Exterior Quality and the Kitchen Quality, but they are both correlated strongly with the Sale Price so I decide to keep them.

Section 5 – Predictive Models

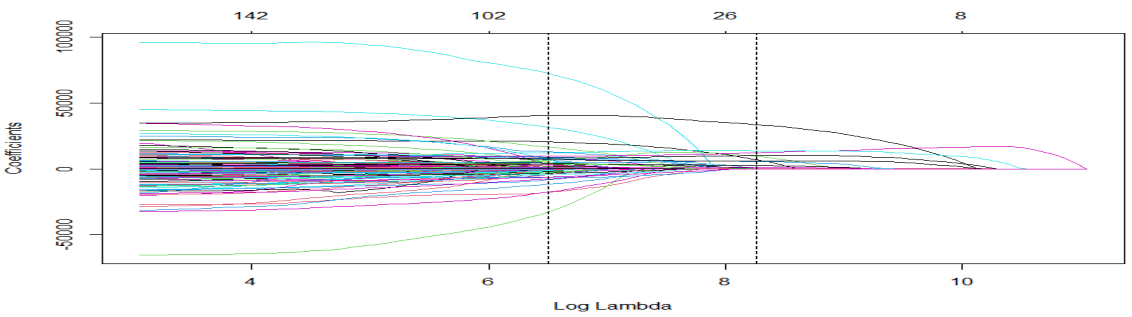
Before stepping into Model building, we are going to prepare the data for the selection variable algorithms (AIC – LASSO). This means that we will create ranking vectors for the ordinal variables (already done this for data presentation) and dummy variables for the categorical. After this modification, we have a vast dataset of 186 variables (staging2 df).

We will use first the LASSO method in order to select the most appropriate variables for our model. Then we will further filter these variables with the AIC algorithm because we want the best predictive model possible. Lasso is a regression analysis method that performs both variable selection and regularization.
What LASSO does is to force the sum of the absolute value of the coefficients to be less than a fixed value which results certain coefficients set to zero, removing them from the model,

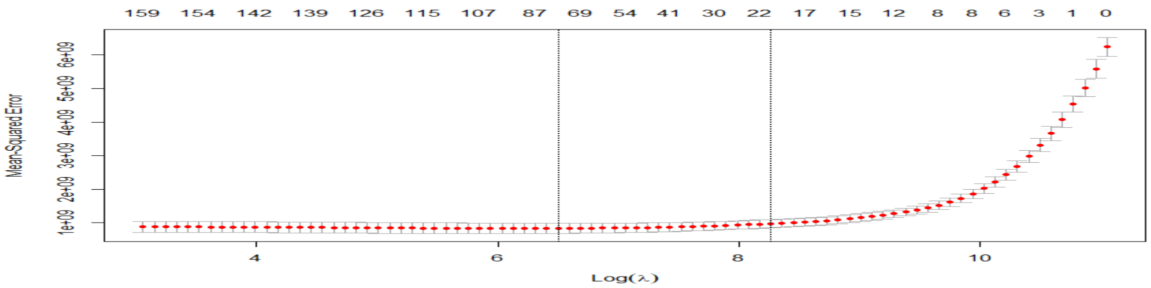
this makes the model simpler and more interpretable. The regularization parameter is λ and measures the degree where the coefficients are penalized.

Below is the outcomes of the LASSO algorithm.

figure 13 – LASSO variables shrinkage



. figure 14– LASSO variables



The above graphs indicate the LASSO selected 19 variables from the 186 of the input data. This is quite acceptable since it dealt further with multicollinearity as most variables droppedoff. We choose the lambda.1se value (right vertical line)instead of lambda. In (left vertical line) because we have a simpler model for a lower value of MSE and less shrinkage penalty.

For working further, I create the final data, dataframe in order to keep the LASSO selected variables and work separately only with them into the model building section.

Now that we have these 19 variables, we are going to build and test some models in order to find the better one. As above mentioned, we use further the AIC algorithm in order to filter further these variables thus we finally have our first model with the following variables (lowest AIC rate 30870.93). ***see appendix table 5***

model1 <- lm(SalePrice ~ Land.Contour.HLS + Lot.Config.CulDSac + Bldg.Type.1Fam + Binhood.2 + Lot.Area + Overall.Qual + Year.Built + Year.Remod.Add + Mas.Vnr.Area + Exter.Qual + Bsmt.Qual + Total.Bsmt.SF + X1st.Flr.SF + Gr.Liv.Area + Kitchen.Qual + Fireplaces + Garage.Cars + Wood.Deck.SF)

Summary output of this model indicates statistically significant p-values (below 0.05) so we cannot drop other variables and the R2 adj is 0.86 – ***See appendix table6***

The model1 fails the regression assumptions of normality of residuals, homoscedasticity and non-linearity – but it satisfies the assumption of independence of errors.
For plotand tests of model1. ***See appendix figure 10 and table 7.***

For the second model, in an attempt to fix the assumptions, I am going to implement a log function in the price response variable thus our model2 is the below.

model2 <- lm(log(SalePrice) ~ Land.Contour.HLS + Lot.Config.CulDSac + Bldg.Type.1Fam + Binhood.2 + Lot.Area + Overall.Qual + Year.Built + Year.Remod.Add +
Mas.Vnr.Area + Exter.Qual + Bsmt.Qual + Total.Bsmt.SF + X1st.Flr.SF +
Gr.Liv.Area + Kitchen.Qual + Fireplaces + Garage.Cars + Wood.Deck.SF

Summary output of this model indicatesnot statistically significant p-values(above the level of 0.05) for these variables Lot.Config.CulDSac,Mas.Vnr.Area,Exter.Qual so I decide to drop them.
See appendix table 8

The model 2 with the log price is the following.

model2_log<- lm(log(SalePrice) ~ Land.Contour.HLS + Bldg.Type.1Fam +
Binhood.2 + Lot.Area + Overall.Qual + Year.Built + Year.Remod.Add
+ Bsmt.Qual + Total.Bsmt.SF + X1st.Flr.SF +
Gr.Liv.Area + Kitchen.Qual + Fireplaces + Garage.Cars + Wood.Deck.SF

Summary output for the revised model2 indicates ok p-values and a good R2 adj at 87%.
This model fails too in the regression assumptions of normality of residuals and Homoscedasticity. For plot and tests of *model2_log*. ***See appendix figure 14 and table9***

Further to the model building, I will try to use polynomials in order to fix the assumptions and have a model that fits the data better from the previous two.

Finally, we will use polynomials in order to construct a third model. We first do residual plots and we implement polynomials in the statistically significant attributes
(***see appendix table 9)***and the final model is presented below.

model3_poly <- lm(log (SalePrice) ~ + Bldg.Type.1Fam +
Binhood.2 + poly (Lot.Area,2) + poly (Overall.Qual,2) + poly (Year.Built,2)+
Bsmt.Qual + poly (Total.Bsmt.SF,2) + poly (X1st.Flr.SF,2) + poly (Gr.Liv.Area,2) +
Kitchen.Qual + Fireplaces + poly (Garage.Cars,2) + Wood.Deck.SF

Unfortunately, the above model does not satisfy again the regression assumptions of Normality of Residuals and Homoscedasticity but the linearity is excellent and the independence of error is assumption is not violated. For plot and tests of *model3_poly*
See appendix figure 15 and table 10

TRAINNING MODEL WITH LOOCV & 10-FOLD CROSS VALIDATION METHODS

Now that we have 3 models, we will implement algorithms in order to "train" the models to the data we already have and will make the selection of the model with the lowest RMSE (root-mean-square error)and the R2 adj metric. Below a short description of these two metrics.

- Root Mean Squared Error (RMSE): As the name suggests it is the square root of the averaged squared difference between the actual value and the predicted value of the target variable. It gives the average prediction error made by the model, thus decrease the RMSE value to increase the accuracy of the model.
- R2 adj: The value of R-squared metric gives an idea about how much percentage of variance in the dependent variable is explained collectively by the independent variables. In other words, it reflects the relationship strength between the target variable and the model on a scale of 0 – 100%. So, a better model should have a high value of R-squared adjusted.

table 11

```
> train_results
               intercept         RMSE  Rsquared          MAE
model1_cv10         TRUE 29594.5362544 0.8604729 19945.9535776
model1_LOOCV        TRUE 29783.5875477 0.8583836 19913.9273965
model2_cv10         TRUE     0.1478678 0.8681322     0.1027233
model2_LOOCV        TRUE     0.1494884 0.8666075     0.1023552
model3_cv10         TRUE     0.1487003 0.8671567     0.1043091
model3_LOOCV        TRUE     0.1489949 0.8675145     0.1037479
>
```

In the above table, we see the results of the training algorithms. Model1 has a great RMSE so I choose not to adopt it. Regarding the other two models there are no great differences both in R Squared and in RMSE metrics, so I choose to adopt the model3 because it was the model with the most satisfying Regression assumptions.

<u>Mathematical formula of the model3</u>

*Log (SalePrice) = 11.520 + 0.085*Bldg.Type.1Fam + 0.087*Binhood.2 + 0.875*poly(Lot.Area, 2)1 - -0.477*poly(Lot.Area, 2)2 + 4.575* poly(Overall.Qual, 2)1-0.899*poly(Overall.Qual, 2)2 + 2.047*poly(Year.Built, 2)1-0.346*poly(Year.Built, 2)2+0.042*Bsmt.Qual+1.293*poly(Total.Bsmt.SF, 2)1+0.472*poly(Total.Bsmt.SF, 2)2+0.842*poly(X1st.Flr.SF, 2)1 -0.976*poly(X1st.Flr.SF, 2)2+3.924*poly(Gr.Liv.Area, 2)1-0.778*poly(Gr.Liv.Area, 2)2+0.065*Kitchen.Qual+0.041*Fireplaces+ 1.216*poly(Garage.Cars, 2)1 -0.142*poly(Garage.Cars, 2)2 + 0.00009*Wood.Deck.SF*

Interpretation of the model and output:

- Intercept or constant is the expected value of our response when all other variables are zero. Graphically it is the point when the regression lies crosses the y-axis.
  If the predictors never equal zero then the intercept has no practical meaning and does not tell anything about the relationship between the response and the predictors. When this happens, in order to fix it and give our intercept a meaning we can rescale the predictors to the center of their distribution.
  If we do this then the intercept now has a meaning and It is the mean value of Y at the chosen value of X it gives us information about the typical observation.
- Since the model is having a log transformed outcome variable, the most natural way to do this is to interpret the exponentiated regression coefficients, since exponentiation is the inverse of logarithm function. Coefficients indicate the mean change of our response variable for one-unit change in the predictor variable while holding the other predictors in the model stable.
- R2 adjusted is very good at 86%, this means that the model fits the observed data and explains the variation at this level.
- P-values are ok and all variables can be considered statistically significant.

See appendix table 11

Finally, we are going to test model3 further to see its accuracy on unseen data. For this we are going to use the test dataset that was given to us at the beginning of this project. As I have previously said the same methods of data transformations that were implement in the training dataset were implemented into the test dataset too. In order to see how the model3 fits the unseen data we are going to create a simple table of the actual values, the predicted values and their differences. Then we are going to see the variance and standard deviation of the absolute differences in order to see how well the model fit the data. At the end we are going to visualize the results, especially the actual values vs the predicted values the regression line and the confidence interval. For prediction tables and visualization of predicted vs actual values. ***See appendix see figure 16***

Conclusion- Model Discussion

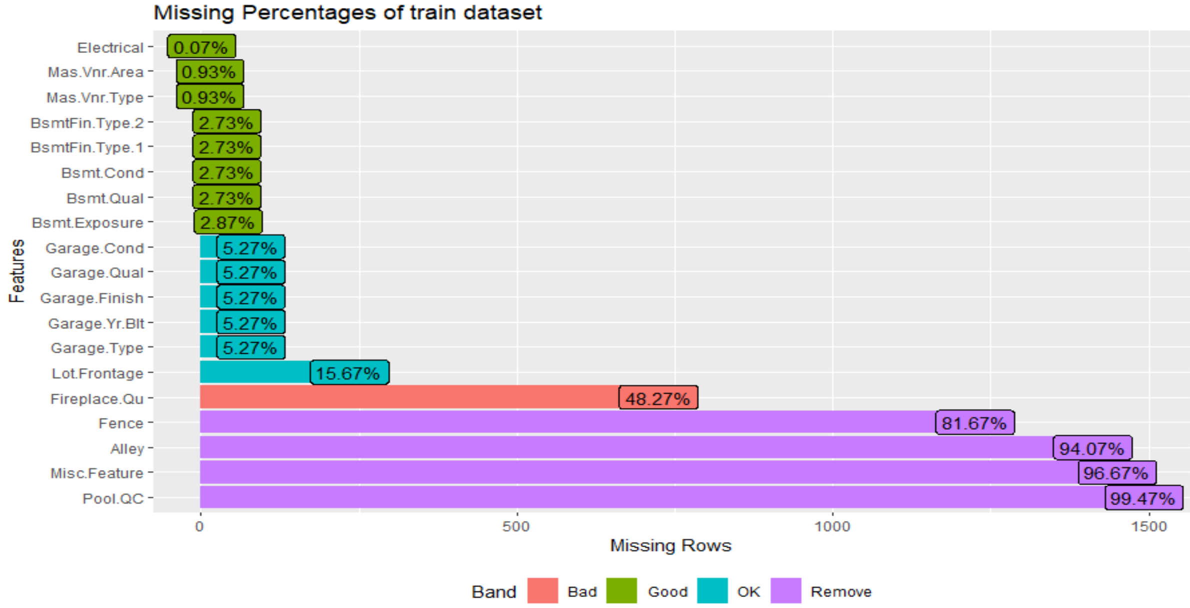
In conclusion, from the data observed we can understand that a typical purchase of a home has the below characteristics. A typical property will be purchased approximately at the price of 158.950 $, the transaction will be of "normal" condition and method will be with Warranty Deed – Conventional. About the property, a typical house in Ames Iowa will be located at the residential low density MS zoning label and the lot. Area will be about 10329 square feet. The majority of the homes has basement, detached type of garage that fits 1-2 cars, number of bathrooms will be 1 or 2 and finally most of the homes has at least 1 fireplace. We can understand also that there is no variety of classification in Ames Iowa neighborhoods except the three “rich” and the “three” poor neighborhoods that I separated earlier on.

Regarding the model, since two out of 4 regression assumption are violated, there are plenty of transformation which can be made except polynomials and log of the response variable price. The rescaling of the numeric data in order to be distributed normally could solve the above problems and have a model that satisfies the normality of residuals and homoscedasticity assumptions in order to trust more its predictive ability. Furthermore, the removal of some outliers could improve the model too. In this part, I strongly believe that a merging of some variables would have good effects (e.g., merging of the bathroom, porch, total area of lot, variables) Regarding the categorical variables I could not find a solution to further binning except the neighborhood and drop these with the top frequencies.

Finally, about the prediction ability of the model I am not satisfied enough because as it seems from the actual vs predicted visualizations many observations are not fit the regression line well. Regarding the minimum difference of the actual data and the predicted data, was 37.4 opposite to the maximum value of the 424658.9 and a median value of 13408.02, regarding the standard deviation of the differences is 25905.01 which I think it is great enough and the model can be further improved.

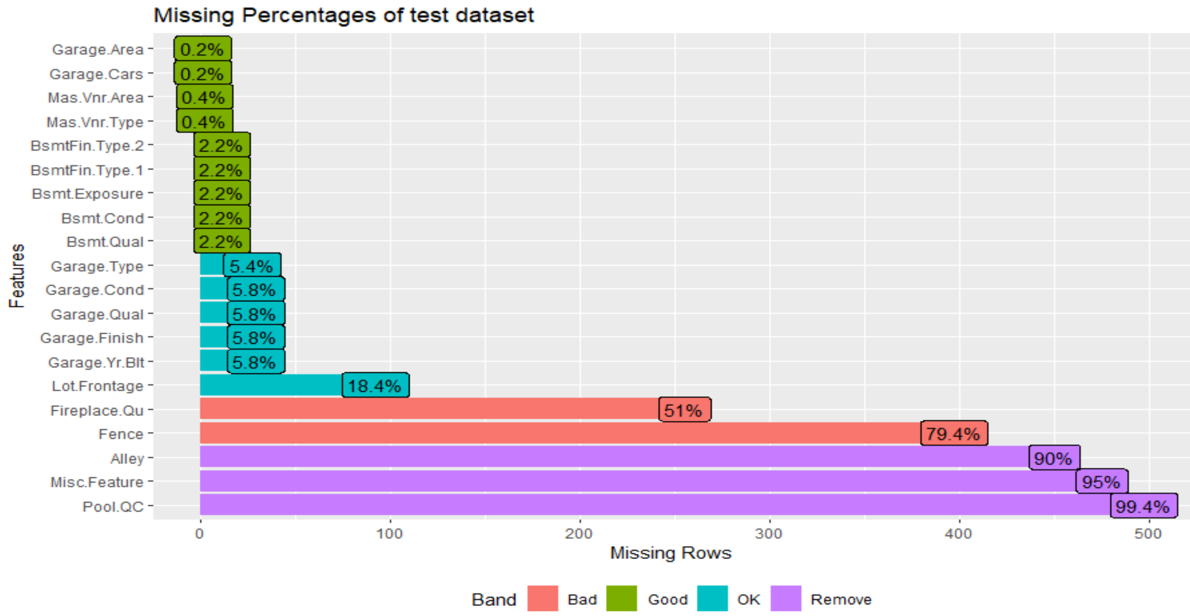
APPENDIX

Fig.1 - Missing data percentages and totals on train dataset




```
      Pool.QC  Misc.Feature         Alley         Fence  Fireplace.Qu  Lot.Frontage   Garage.Type
         1492          1450          1411          1225           724           235            79
 Garage.Yr.Blt Garage.Finish   Garage.Qual   Garage.Cond Bsmt.Exposure     Bsmt.Qual     Bsmt.Cond
           79            79            79            79            43            41            41
BsmtFin.Type.1 BsmtFin.Type.2  Mas.Vnr.Type  Mas.Vnr.Area    Electrical
           41            41            14            14             1
> |
```

Fig.2 – Missing data percentages and totals on test dataset




```
      Pool.QC  Misc.Feature         Alley         Fence  Fireplace.Qu  Lot.Frontage Garage.Yr.Blt
          497           475           450           397           255            92            29
Garage.Finish   Garage.Qual   Garage.Cond   Garage.Type     Bsmt.Qual     Bsmt.Cond Bsmt.Exposure
           29            29            29            27            11            11            11
BsmtFin.Type.1 BsmtFin.Type.2  Mas.Vnr.Type  Mas.Vnr.Area   Garage.Cars   Garage.Area
           11            11             2             2             1             1
> |
```

Figure 5

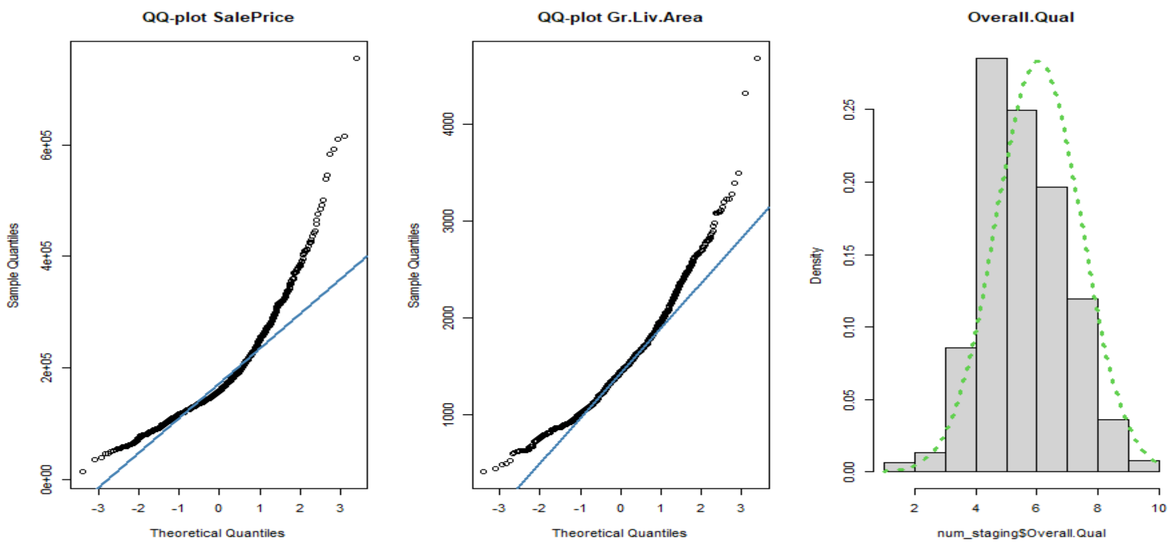


Table 1 – Normality tests

```
> lillie.test(num_staging$SalePrice)

        Lilliefors (Kolmogorov-Smirnov) normality test

data:  num_staging$SalePrice
D = 0.12413, p-value < 2.2e-16

> shapiro.test(num_staging$SalePrice)

        Shapiro-Wilk normality test

data:  num_staging$SalePrice
W = 0.88079, p-value < 2.2e-16

> lillie.test(num_staging$Gr.Liv.Area)

        Lilliefors (Kolmogorov-Smirnov) normality test

data:  num_staging$Gr.Liv.Area
D = 0.065242, p-value < 2.2e-16

> shapiro.test(num_staging$Gr.Liv.Area)

        Shapiro-Wilk normality test

data:  num_staging$Gr.Liv.Area
W = 0.94898, p-value < 2.2e-16

> lillie.test(num_staging$Overall.Qual)

        Lilliefors (Kolmogorov-Smirnov) normality test

data:  num_staging$Overall.Qual
D = 0.1631, p-value < 2.2e-16

> shapiro.test(num_staging$Overall.Qual)

        Shapiro-Wilk normality test

data:  num_staging$Overall.Qual
W = 0.94842, p-value < 2.2e-16

>
```

Fig.6 – Top correlated with price

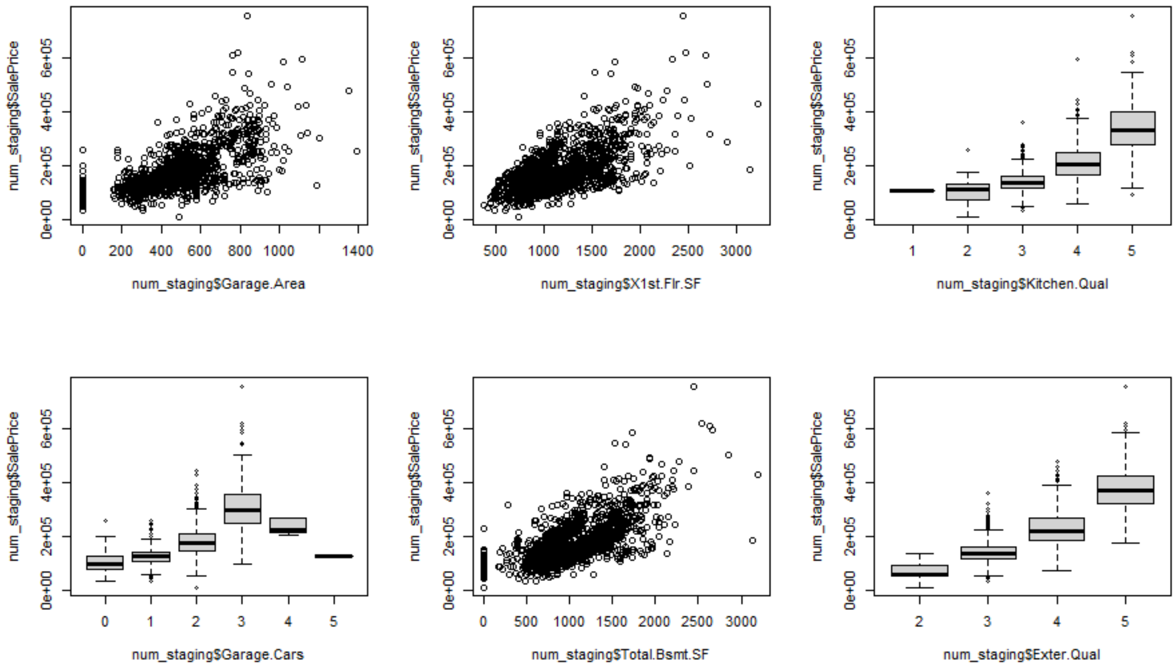


Fig.8 – Various histograms

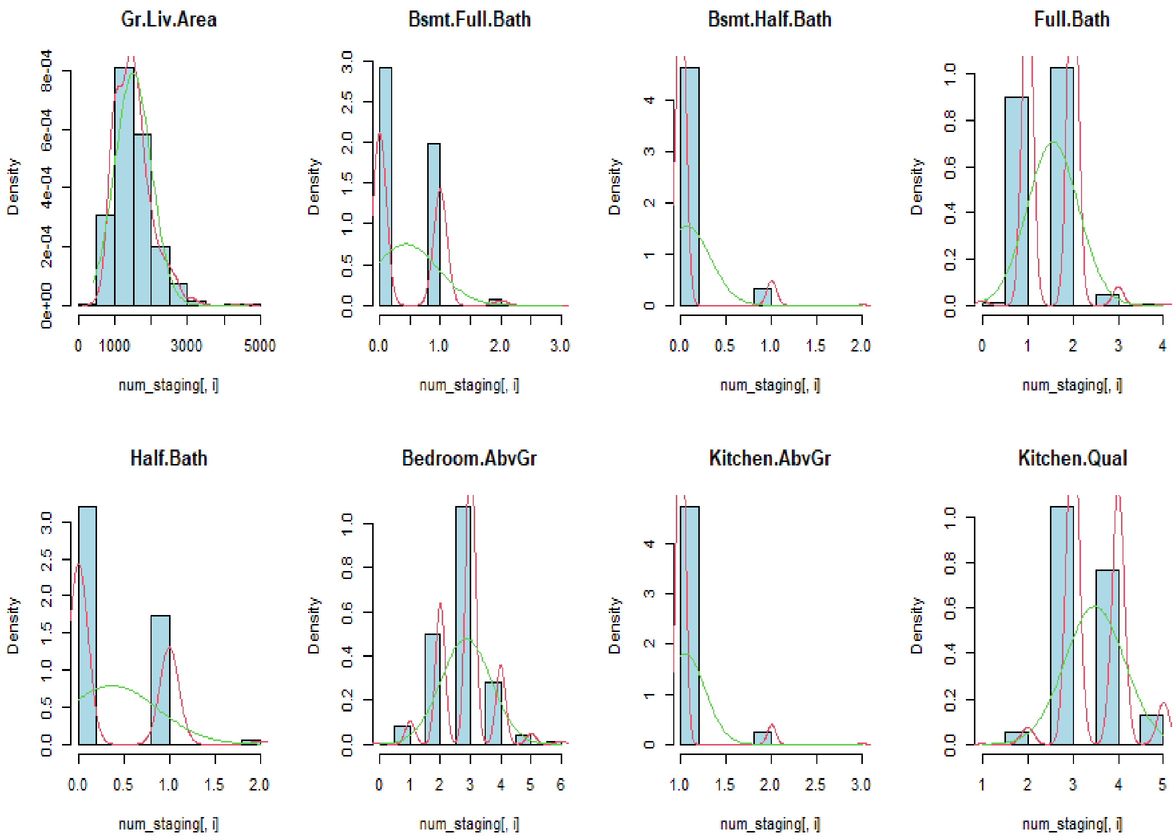


Fig.9 - Various histograms

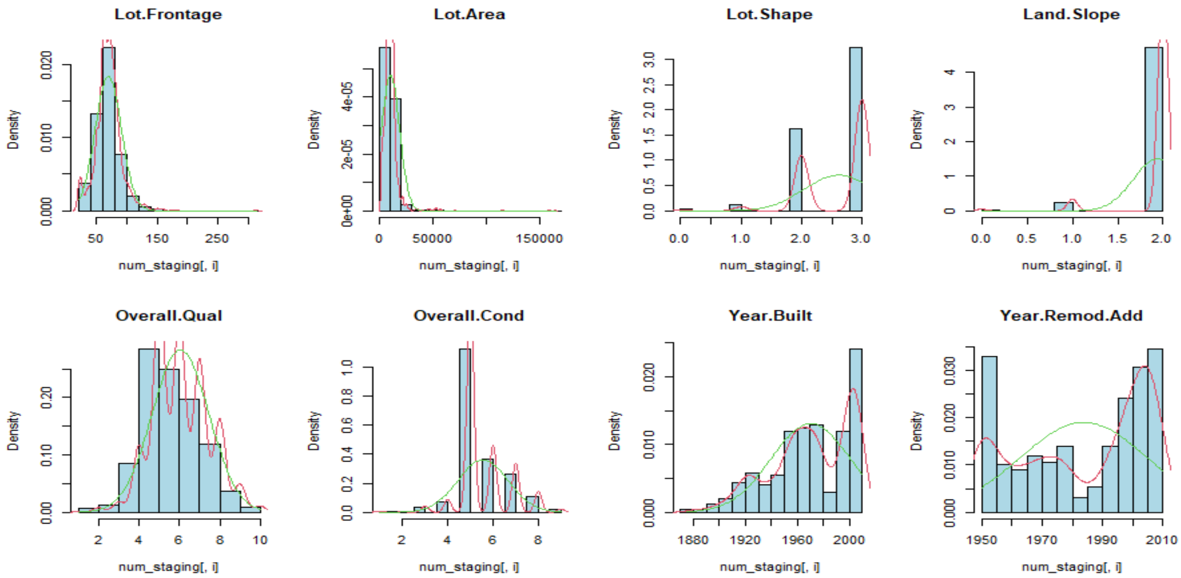


Table 3 – drop list variables (Frequencies above 98%)

```
Frequencies
cat_staging$Street
Type: Factor

              Freq   % Valid   % Valid Cum.   % Total   % Total Cum.
----------- ------ --------- -------------- --------- --------------
       Pave   1493     99.53          99.53     99.53          99.53
       Grvl      7      0.47         100.00      0.47         100.00
       <NA>      0                               0.00         100.00
      Total   1500    100.00         100.00    100.00         100.00

cat_staging$Utilities
Type: Factor

               Freq   % Valid   % Valid Cum.   % Total   % Total Cum.
------------ ------ --------- -------------- --------- --------------
      AllPub   1498    99.867         99.867    99.867         99.867
      NoSeWa      1     0.067         99.933     0.067         99.933
      NoSewr      1     0.067        100.000     0.067        100.000
        <NA>      0                              0.000        100.000
       Total   1500   100.000        100.000   100.000        100.000

cat_staging$Condition.2
Type: Factor

               Freq   % Valid   % Valid Cum.   % Total   % Total Cum.
------------ ------ --------- -------------- --------- --------------
        Norm   1484    98.933         98.933    98.933         98.933
       Feedr      7     0.467         99.400     0.467         99.400
      Artery      3     0.200         99.600     0.200         99.600
        PosA      2     0.133         99.733     0.133         99.733
        PosN      1     0.067         99.800     0.067         99.800
        RRAe      1     0.067         99.867     0.067         99.867
        RRAn      1     0.067         99.933     0.067         99.933
        RRNn      1     0.067        100.000     0.067        100.000
        <NA>      0                              0.000        100.000
       Total   1500   100.000        100.000   100.000        100.000
```

Table 4 – drop list variables (Frequencies above 98%)

cat_staging$Condition.2
Type: Factor

	Freq	% Valid	% Valid Cum.	% Total	% Total Cum.
Norm	1484	98.933	98.933	98.933	98.933
Feedr	7	0.467	99.400	0.467	99.400
Artery	3	0.200	99.600	0.200	99.600
PosA	2	0.133	99.733	0.133	99.733
PosN	1	0.067	99.800	0.067	99.800
RRAe	1	0.067	99.867	0.067	99.867
RRAn	1	0.067	99.933	0.067	99.933
RRNn	1	0.067	100.000	0.067	100.000
<NA>	0			0.000	100.000
Total	1500	100.000	100.000	100.000	100.000

cat_staging$Roof.Matl
Type: Factor

	Freq	% Valid	% Valid Cum.	% Total	% Total Cum.
CompShg	1479	98.600	98.600	98.600	98.600
Tar&Grv	12	0.800	99.400	0.800	99.400
WdShake	4	0.267	99.667	0.267	99.667
WdShngl	4	0.267	99.933	0.267	99.933
Roll	1	0.067	100.000	0.067	100.000
<NA>	0			0.000	100.000
Total	1500	100.000	100.000	100.000	100.000

cat_staging$Heating
Type: Factor

	Freq	% Valid	% Valid Cum.	% Total	% Total Cum.
GasA	1476	98.40	98.40	98.40	98.40
GasW	17	1.13	99.53	1.13	99.53
Grav	5	0.33	99.87	0.33	99.87
Wall	2	0.13	100.00	0.13	100.00
<NA>	0			0.00	100.00
Total	1500	100.00	100.00	100.00	100.00

Table 5 – AIC results

```
Step:  AIC=30870.93
SalePrice ~ Land.Contour.HLS + Lot.Config.CulDSac + Bldg.Type.1Fam +
    Binhood.2 + Lot.Area + Overall.Qual + Year.Built + Year.Remod.Add +
    Mas.Vnr.Area + Exter.Qual + Bsmt.Qual + Total.Bsmt.SF + X1st.Flr.SF +
    Gr.Liv.Area + Kitchen.Qual + Fireplaces + Garage.Cars + Wood.Deck.SF

                     Df  Sum of Sq        RSS   AIC
<none>                             1.2681e+12 30871
- Bsmt.Qual           1 4.5609e+09 1.2726e+12 30874
- Wood.Deck.SF        1 6.2911e+09 1.2743e+12 30876
- Year.Remod.Add      1 6.3422e+09 1.2744e+12 30876
- Lot.Config.CulDSac  1 9.5434e+09 1.2776e+12 30880
- X1st.Flr.SF         1 1.1444e+10 1.2795e+12 30882
- Mas.Vnr.Area        1 1.2003e+10 1.2801e+12 30883
- Total.Bsmt.SF       1 1.5085e+10 1.2831e+12 30887
- Lot.Area            1 1.7081e+10 1.2851e+12 30889
- Garage.Cars         1 1.8138e+10 1.2862e+12 30890
- Year.Built          1 2.3173e+10 1.2912e+12 30896
- Fireplaces          1 2.4431e+10 1.2925e+12 30898
- Kitchen.Qual        1 2.5402e+10 1.2935e+12 30899
- Exter.Qual          1 3.0767e+10 1.2988e+12 30905
- Land.Contour.HLS    1 3.1698e+10 1.2998e+12 30906
- Bldg.Type.1Fam      1 7.1117e+10 1.3392e+12 30951
- Overall.Qual        1 9.4923e+10 1.3630e+12 30977
- Binhood.2           1 1.3328e+11 1.4013e+12 31019
- Gr.Liv.Area         1 2.3646e+11 1.5045e+12 31125
```

Table 6 – model1

```
> summary(model1) # R2 0.8634 - pvalues ok

Call:
lm(formula = SalePrice ~ Land.Contour.HLS + Lot.Config.CulDSac +
    Bldg.Type.1Fam + Binhood.2 + Lot.Area + Overall.Qual + Year.Built +
    Year.Remod.Add + Mas.Vnr.Area + Exter.Qual + Bsmt.Qual +
    Total.Bsmt.SF + X1st.Flr.SF + Gr.Liv.Area + Kitchen.Qual +
    Fireplaces + Garage.Cars + Wood.Deck.SF, data = finaldata)

Residuals:
    Min      1Q  Median      3Q     Max
-306220  -15311    -657   14082  253944

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)        -7.965e+05  1.063e+05  -7.491 1.17e-13 ***
Land.Contour.HLS    2.392e+04  3.931e+03   6.085 1.48e-09 ***
Lot.Config.CulDSac  1.111e+04  3.329e+03   3.339 0.000863 ***
Bldg.Type.1Fam      1.942e+04  2.131e+03   9.114  < 2e-16 ***
Binhood.2           3.992e+04  3.199e+03  12.476  < 2e-16 ***
Lot.Area            4.581e-01  1.026e-01   4.466 8.56e-06 ***
Overall.Qual        1.067e+04  1.013e+03  10.529  < 2e-16 ***
Year.Built          2.036e+02  3.913e+01   5.202 2.24e-07 ***
Year.Remod.Add      1.425e+02  5.238e+01   2.722 0.006572 **
Mas.Vnr.Area        1.843e+01  4.922e+00   3.744 0.000188 ***
Exter.Qual          1.382e+04  2.305e+03   5.994 2.56e-09 ***
Bsmt.Qual           3.286e+03  1.424e+03   2.308 0.021136 *
Total.Bsmt.SF       1.578e+01  3.760e+00   4.197 2.86e-05 ***
X1st.Flr.SF         1.490e+01  4.075e+00   3.656 0.000265 ***
Gr.Liv.Area         3.699e+01  2.226e+00  16.618  < 2e-16 ***
Kitchen.Qual        9.935e+03  1.824e+03   5.447 6.00e-08 ***
Fireplaces          7.331e+03  1.372e+03   5.342 1.06e-07 ***
Garage.Cars         6.296e+03  1.368e+03   4.603 4.53e-06 ***
Wood.Deck.SF        1.682e+01  6.206e+00   2.711 0.006793 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 29260 on 1481 degrees of freedom
Multiple R-squared:  0.865,	Adjusted R-squared:  0.8634
F-statistic: 527.3 on 18 and 1481 DF,  p-value: < 2.2e-16

>
```

Figure 10 – model1 plots

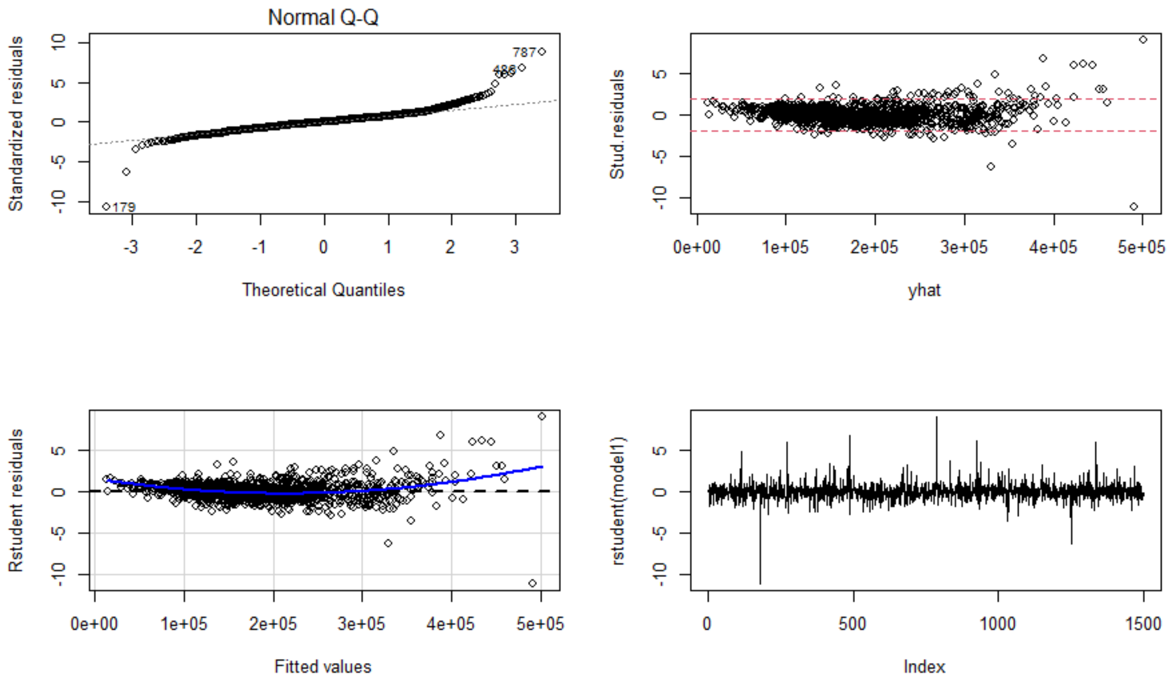


Table 7 – model1 tests

```
> lillie.test(residuals(model1))

        Lilliefors (Kolmogorov-Smirnov) normality test

data:  residuals(model1)
D = 0.085068, p-value < 2.2e-16

> shapiro.test(residuals(model1))

        Shapiro-Wilk normality test

data:  residuals(model1)
W = 0.87898, p-value < 2.2e-16

> ncvTest(model1)
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 1442.05, 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  59.678 < 2.2e-16 ***
      1495
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> durbinWatsonTest(model1)
 lag Autocorrelation D-W Statistic p-value
   1     -0.03760724      2.075178   0.142
 Alternative hypothesis: rho != 0
>
```

Table 8 – model2 coefficients with not statistically significant values

```
Coefficients:
                      Estimate Std. Error t value Pr(>|t|)    
(Intercept)          3.476e+00  5.368e-01   6.476 1.27e-10 ***
Land.Contour.HLS     5.031e-02  1.985e-02   2.535  0.01134 *  
Lot.Config.CulDSac   2.750e-02  1.681e-02   1.636  0.10204    
Bldg.Type.1Fam       9.385e-02  1.076e-02   8.723  < 2e-16 ***
Binhood.2            5.162e-02  1.615e-02   3.196  0.00142 ** 
Lot.Area             2.262e-06  5.178e-07   4.370 1.33e-05 ***
Overall.Qual         8.296e-02  5.114e-03  16.222  < 2e-16 ***
Year.Built           1.756e-03  1.975e-04   8.890  < 2e-16 ***
Year.Remod.Add       1.846e-03  2.644e-04   6.983 4.36e-12 ***
Mas.Vnr.Area        -2.649e-05  2.485e-05  -1.066  0.28664    
Exter.Qual           1.366e-02  1.164e-02   1.174  0.24070    
Bsmt.Qual            2.192e-02  7.187e-03   3.050  0.00233 ** 
Total.Bsmt.SF        9.771e-05  1.898e-05   5.148 2.99e-07 ***
X1st.Flr.SF          5.877e-05  2.057e-05   2.857  0.00434 ** 
Gr.Liv.Area          1.923e-04  1.124e-05  17.112  < 2e-16 ***
Kitchen.Qual         2.856e-02  9.208e-03   3.102  0.00196 ** 
Fireplaces           5.624e-02  6.929e-03   8.118 9.90e-16 ***
Garage.Cars          4.610e-02  6.906e-03   6.675 3.49e-11 ***
Wood.Deck.SF         8.186e-05  3.133e-05   2.613  0.00907 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1477 on 1481 degrees of freedom
Multiple R-squared:  0.8714,	Adjusted R-squared:  0.8698 
F-statistic: 557.4 on 18 and 1481 DF,  p-value: < 2.2e-16
```

Table 9 – model2_logoutput

```
> summary(model2_log) # p-values ok - R2 adj 0.87

Call:
lm(formula = log(SalePrice) ~ Land.Contour.HLS + Bldg.Type.1Fam + 
    Binhood.2 + Lot.Area + Overall.Qual + Year.Built + Year.Remod.Add + 
    Bsmt.Qual + Total.Bsmt.SF + X1st.Flr.SF + Gr.Liv.Area + Kitchen.Qual + 
    Fireplaces + Garage.Cars + Wood.Deck.SF, data = finaldata)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.60244 -0.06951  0.00793  0.08352  0.58462 

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)      3.312e+00  5.275e-01   6.278 4.50e-10 ***
Land.Contour.HLS 5.222e-02  1.982e-02   2.634 0.008522 ** 
Bldg.Type.1Fam   9.464e-02  1.076e-02   8.796  < 2e-16 ***
Binhood.2        4.963e-02  1.551e-02   3.200 0.001405 ** 
Lot.Area         2.417e-06  5.092e-07   4.747 2.27e-06 ***
Overall.Qual     8.428e-02  4.902e-03  17.192  < 2e-16 ***
Year.Built       1.781e-03  1.951e-04   9.131  < 2e-16 ***
Year.Remod.Add   1.917e-03  2.603e-04   7.363 2.96e-13 ***
Bsmt.Qual        2.307e-02  7.172e-03   3.216 0.001327 ** 
Total.Bsmt.SF    9.621e-05  1.893e-05   5.081 4.23e-07 ***
X1st.Flr.SF      5.977e-05  2.052e-05   2.912 0.003640 ** 
Gr.Liv.Area      1.903e-04  1.115e-05  17.068  < 2e-16 ***
Kitchen.Qual     3.142e-02  8.630e-03   3.641 0.000281 ***
Fireplaces       5.556e-02  6.910e-03   8.040 1.81e-15 ***
Garage.Cars      4.641e-02  6.907e-03   6.720 2.59e-11 ***
Wood.Deck.SF     8.277e-05  3.133e-05   2.642 0.008323 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1478 on 1484 degrees of freedom
Multiple R-squared:  0.8709,	Adjusted R-squared:  0.8696 
F-statistic: 667.6 on 15 and 1484 DF,  p-value: < 2.2e-16
```

Figure 14 – model2_log plots

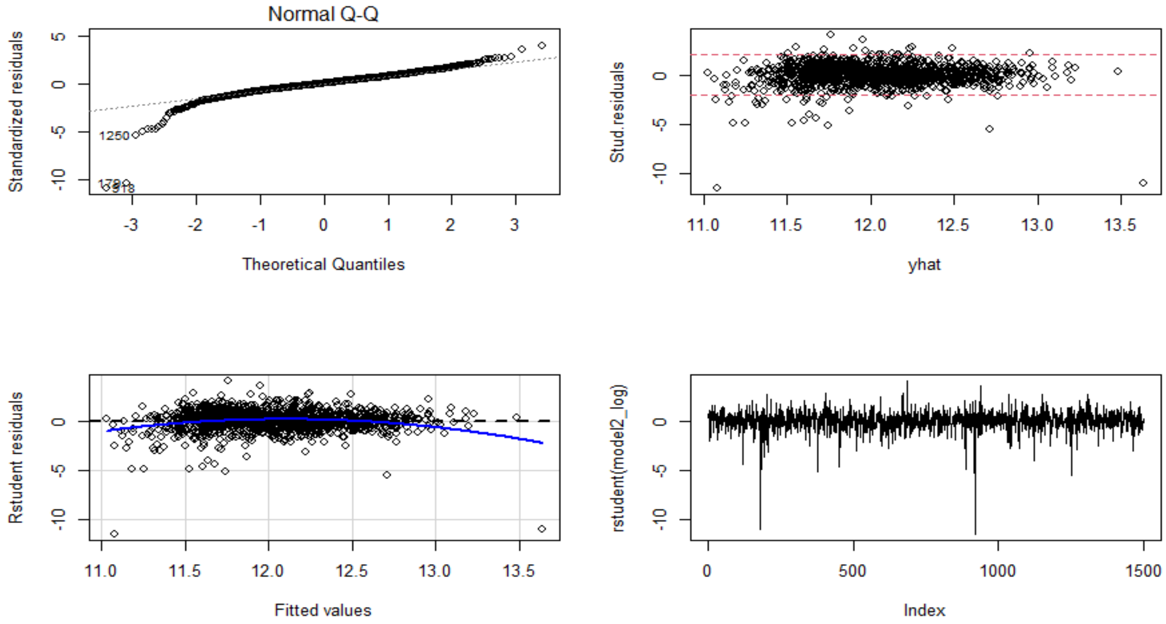


Table 10 – residuals p-values in order to choose where to implement polynomials

	Test stat	Pr(>|Test stat|)	
Land.Contour.HLS	1.0222	0.306862	
Bldg.Type.1Fam	0.6468	0.517876	
Binhood.2	2.0070	0.044933	*
Lot.Area	-3.0326	0.002466	**
Overall.Qual	-8.0383	1.844e-15	***
Year.Built	-4.8128	1.640e-06	***
Year.Remod.Add	-1.7941	0.072993	.
Bsmt.Qual	-1.3708	0.170630	
Total.Bsmt.SF	-6.2286	6.118e-10	***
X1st.Flr.SF	-7.8755	6.510e-15	***
Gr.Liv.Area	-8.0573	1.589e-15	***
Kitchen.Qual	-0.8385	0.401908	
Fireplaces	-0.2460	0.805710	
Garage.Cars	-2.8910	0.003896	**
Wood.Deck.SF	0.0834	0.933559	
Tukey test	-9.0235	< 2.2e-16	***

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Table 11 – model3_poly output

```
> summary(model3_poly)

Call:
lm(formula = log(SalePrice) ~ +Bldg.Type.1Fam + Binhood.2 + poly(Lot.Area,
    2) + poly(Overall.Qual, 2) + poly(Year.Built, 2) + Bsmt.Qual +
    poly(Total.Bsmt.SF, 2) + poly(X1st.Flr.SF, 2) + poly(Gr.Liv.Area,
    2) + Kitchen.Qual + Fireplaces + poly(Garage.Cars, 2) + Wood.Deck.SF,
    data = finaldata)

Residuals:
     Min       1Q   Median       3Q      Max
-1.36343 -0.07178  0.00441  0.08080  0.59591

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)              1.152e+01  4.050e-02 284.441  < 2e-16 ***
Bldg.Type.1Fam           8.525e-02  1.124e-02   7.582 5.96e-14 ***
Binhood.2                8.735e-02  1.617e-02   5.403 7.61e-08 ***
poly(Lot.Area, 2)1       8.756e-01  1.650e-01   5.307 1.28e-07 ***
poly(Lot.Area, 2)2      -4.779e-01  1.681e-01  -2.843  0.00453 **
poly(Overall.Qual, 2)1   4.575e+00  2.733e-01  16.739  < 2e-16 ***
poly(Overall.Qual, 2)2  -8.991e-01  1.795e-01  -5.008 6.17e-07 ***
poly(Year.Built, 2)1     2.047e+00  2.437e-01   8.401  < 2e-16 ***
poly(Year.Built, 2)2    -3.464e-01  1.869e-01  -1.853  0.06407 .
Bsmt.Qual                4.295e-02  8.094e-03   5.306 1.29e-07 ***
poly(Total.Bsmt.SF, 2)1  1.294e+00  3.228e-01   4.009 6.41e-05 ***
poly(Total.Bsmt.SF, 2)2  4.725e-01  2.744e-01   1.722  0.08534 .
poly(X1st.Flr.SF, 2)1    8.427e-01  3.417e-01   2.466  0.01378 *
poly(X1st.Flr.SF, 2)2   -9.765e-01  2.295e-01  -4.256 2.21e-05 ***
poly(Gr.Liv.Area, 2)1    3.925e+00  2.201e-01  17.834  < 2e-16 ***
poly(Gr.Liv.Area, 2)2   -7.785e-01  1.688e-01  -4.613 4.31e-06 ***
Kitchen.Qual             6.562e-02  8.339e-03   7.869 6.86e-15 ***
Fireplaces               4.126e-02  6.912e-03   5.970 2.96e-09 ***
poly(Garage.Cars, 2)1    1.217e+00  2.066e-01   5.890 4.77e-09 ***
poly(Garage.Cars, 2)2   -1.424e-01  1.562e-01  -0.911  0.36220
Wood.Deck.SF             9.858e-05  3.062e-05   3.219  0.00131 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1443 on 1479 degrees of freedom
Multiple R-squared:  0.8775,    Adjusted R-squared:  0.8758
F-statistic: 529.6 on 20 and 1479 DF,  p-value: < 2.2e-16
```

Figure 15 – model3_poly plots

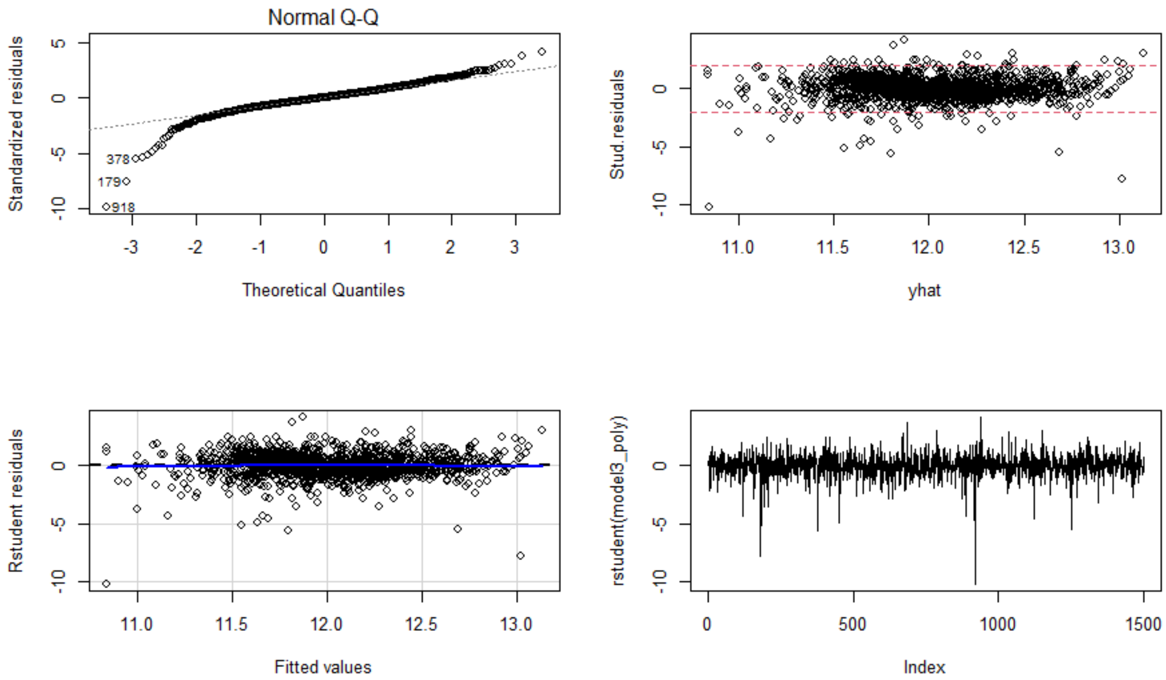
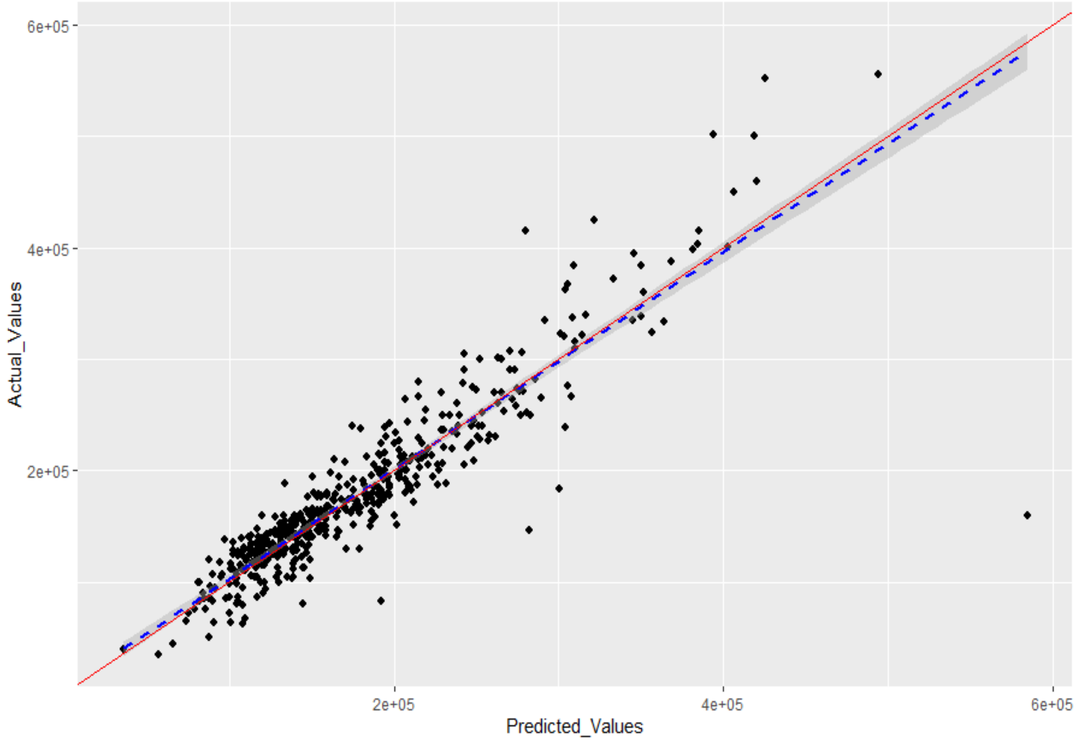


Table 12 – model3_poly tests

```
> lillie.test(residuals(model3_poly))

        Lilliefors (Kolmogorov-Smirnov) normality test

data:  residuals(model3_poly)
D = 0.065027, p-value = 2.646e-16

> shapiro.test(residuals(model3_poly))

        Shapiro-Wilk normality test

data:  residuals(model3_poly)
W = 0.92566, p-value < 2.2e-16

> ncvTest(model3_poly)
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 53.18312, Df = 1, p = 3.0386e-13
> leveneTest(rstudent(model3_poly)~yhat.quantiles)
Levene's Test for Homogeneity of Variance (center = median)
        Df F value    Pr(>F)
group    3  5.9496 0.0004955 ***
      1495
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> durbinWatsonTest(model3_poly)
 lag Autocorrelation D-W Statistic p-value
   1     -0.02442781      2.048656   0.322
 Alternative hypothesis: rho != 0
> 
```

Figure 16: Regression – actual vs predicted values



SOURCES

- **OpenIntro Statistics -** David M Diez - Christopher D Barr - Mine Cetinkaya-Rundel
- **Εισαγωγή στον Προγραμματισμό και στη Στατιστική Ανάλυση με R (Καρλής Δ, Τζούφρας Ι)**
- https://statsandr.com/blog/correlogram-in-r-how-to-highlight-the-most-correlated-variables-in-a-dataset/
- https://www.kaggle.com/erikbruin/house-prices-lasso-xgboost-and-a-detailed-eda
- https://www.geeksforgeeks.org/cross-validation-in-r-programming/
- https://www.geeksforgeeks.org/simple-linear-regression-using-r/

R CODE – REFERENCE

```
# Main Assignment - Statistics I - BA PT - Vretteas Stylianos -
p2822003
setwd("D:/Documents_2/01 Business Analytics - AUEB/02 Statistics 1/big
project/data_60")
train_60<-read.csv(file.choose(),sep = ";")# load the
ames_iowa_housing_60
test_60<-read.csv(file.choose(),sep = ";") #load the
ames_iowa_housing_test
#
str(train_60)
summary(train_60)
str(test_60)
summary(test_60)
#
library(DataExplorer)
introduce(train_60) # 7168 missing values in the train_60
introduce(test_60)  #2370 missing values in the test_60
# find NA values in train_60
na_values<- which(colSums(is.na(train_60)) >0)
sort(colSums(sapply(train_60[na_values], is.na)), decreasing = TRUE)
# find NA values in test_60
na_values1 <- which(colSums(is.na(test_60)) >0)
sort(colSums(sapply(test_60[na_values1], is.na)), decreasing = TRUE)
# visualize missing values percentages
plot_missing(train_60, missing_only = TRUE, title = "Missing
Percentages of train dataset")
plot_missing(test_60, missing_only = TRUE, title = "Missing Percentages
of test dataset")
# create the staging df
staging<- train_60
# exclude X, order, PID
drop <- c("X","Order","PID")
staging <- staging[,!(names(staging) %in% drop)]
# fix each column - NA is not a missing value but indicates no presence
of this feature
staging$Pool.QC[is.na(staging$Pool.QC)] <- "None"
staging$Misc.Feature[is.na(staging$Misc.Feature)] <- "None"
staging$Alley[is.na(staging$Alley)] <- "None"
staging$Fence[is.na(staging$Fence)] <- "None"
staging$Fireplace.Qu[is.na(staging$Fireplace.Qu)] <- "None"
# test dataset imputation
test_60$Pool.QC[is.na(test_60$Pool.QC)] <- "None"
test_60$Misc.Feature[is.na(test_60$Misc.Feature)] <- "None"
test_60$Alley[is.na(test_60$Alley)] <- "None"
test_60$Fence[is.na(test_60$Fence)] <- "None"
test_60$Fireplace.Qu[is.na(test_60$Fireplace.Qu)] <- "None"
#  Garage Variables - 79 observations NA - no garage
staging$Garage.Type[is.na(staging$Garage.Type)] <- "None"
staging$Garage.Finish[is.na(staging$Garage.Finish)] <- "None"
staging$Garage.Qual[is.na(staging$Garage.Qual)] <- "None"
staging$Garage.Cond[is.na(staging$Garage.Cond)] <- "None"
staging$Garage.Yr.Blt[is.na(staging$Garage.Yr.Blt)]<- 0
# test dataset imputation
test_60$Garage.Type[is.na(test_60$Garage.Type)] <- "None"
test_60$Garage.Finish[is.na(test_60$Garage.Finish)] <- "None"
```

```
test_60$Garage.Qual[is.na(test_60$Garage.Qual)] <- "None"
test_60$Garage.Cond[is.na(test_60$Garage.Cond)] <- "None"
test_60$Garage.Yr.Blt[is.na(test_60$Garage.Yr.Blt)]<- 0
test_60$Garage.Cars[is.na(test_60$Garage.Cars)]<- 0
test_60$Garage.Area[is.na(test_60$Garage.Area)]<- 0
# 41 observations  NA - no basement
staging$Bsmt.Qual[is.na(staging$Bsmt.Qual)] <- "None"
staging$Bsmt.Cond[is.na(staging$Bsmt.Cond)] <- "None"
staging$Bsmt.Exposure[is.na(staging$Bsmt.Exposure)] <- "None"
staging$BsmtFin.SF.1[is.na(staging$BsmtFin.SF.1)] <- "None"
staging$BsmtFin.SF.2[is.na(staging$BsmtFin.SF.2)] <- "None"
staging$BsmtFin.Type.1[is.na(staging$BsmtFin.Type.1)]<- 0
staging$BsmtFin.Type.2[is.na(staging$BsmtFin.Type.2)]<- 0
# test dataset imputation
test_60$Bsmt.Qual[is.na(test_60$Bsmt.Qual)] <- "None"
test_60$Bsmt.Cond[is.na(test_60$Bsmt.Cond)] <- "None"
test_60$Bsmt.Exposure[is.na(test_60$Bsmt.Exposure)] <- "None"
test_60$BsmtFin.Type.1[is.na(test_60$BsmtFin.Type.1)]<- 0
test_60$BsmtFin.Type.2[is.na(test_60$BsmtFin.Type.2)]<- 0
# 14 observations NA - No Masonry veneer type
staging$Mas.Vnr.Type[is.na(staging$Mas.Vnr.Type)]<- "None"
staging$Mas.Vnr.Area[is.na(staging$Mas.Vnr.Area)]<- 0
# test dataset imputation
test_60$Mas.Vnr.Type[is.na(test_60$Mas.Vnr.Type)]<- "None"
test_60$Mas.Vnr.Area[is.na(test_60$Mas.Vnr.Area)]<- 0
# 1 observation NA - No Electrical
table(staging$Electrical) # SBrkris the most common
staging$Electrical[is.na(staging$Electrical)]<- "SBrkr"
# create subset4 with neighborhoods and lot.frontage
subset4_names <- names(staging) %in% c("Neighborhood","Lot.Frontage")
subset4 <- staging[subset4_names]
unique(subset4$Neighborhood)
subset4<- na.omit(subset4)
summary(subset4) # no NAs in this vlookup subset
median4<- aggregate(subset4$Lot.Frontage,
                    by = list(subset4$Neighborhood),
                    FUN = median)  # for median
colnames(median4) <- c("Neighborhood","Lot.Frontage")
staging$Lot.Frontage[is.na(staging$Lot.Frontage)] <-
median4$Lot.Frontage[match(staging$Neighborhood,median4$Neighborhood)][
which(is.na(staging$Lot.Frontage))]
test_60$Lot.Frontage[is.na(test_60$Lot.Frontage)] <-
median4$Lot.Frontage[match(test_60$Neighborhood,median4$Neighborhood)][
which(is.na(test_60$Lot.Frontage))]
summary(median4$Lot.Frontage) # after imputation still 2 NAs
staging$Lot.Frontage[is.na(staging$Lot.Frontage)] <-
median(median4$Lot.Frontage) # imputation median of subset4
# final summaries
which(colSums(is.na(staging)) > 0) # 0
which(colSums(is.na(test_60)) > 0) # 0
#--------------------------------finallyno NA values into the staging
and the test_60--------------------------#
#--------------------------------Encoding_of_Ordinal_Variables----------
----------------------------------------#
library(plyr)
rankings <- c("None" = 0,"Po" = 1,"Fa" = 2,"TA" = 3,"Gd" = 4,"Ex"= 5) #
create ranking vector for the ordinal with same levels
```

```
staging$Pool.QC<- as.integer(revalue(staging$Pool.QC, rankings)) #
staging$Pool.QC
staging$Garage.Cond<- as.integer(revalue(staging$Garage.Cond,
rankings)) # Garage.Cond
staging$Garage.Qual<- as.integer(revalue(staging$Garage.Qual,
rankings)) # Garage.Qual
staging$Fireplace.Qu<- as.integer(revalue(staging$Fireplace.Qu,
rankings)) # Fireplace.Qu
staging$Kitchen.Qual<- as.integer(revalue(staging$Kitchen.Qual,
rankings)) # Kitchen.Qual
staging$Heating.QC<- as.integer(revalue(staging$Heating.QC, rankings))
# Heating.QC
staging$Bsmt.Cond<- as.integer(revalue(staging$Bsmt.Cond, rankings)) #
Bsmt.Cond
staging$Bsmt.Qual<- as.integer(revalue(staging$Bsmt.Qual, rankings)) #
Bsmt.Qual
staging$Exter.Cond<- as.integer(revalue(staging$Exter.Cond, rankings))
# Exter.Cond
staging$Exter.Qual<- as.integer(revalue(staging$Exter.Qual, rankings))
# Exter.Qual
# test dataset encoding
test_60$Pool.QC <- as.integer(revalue(test_60$Pool.QC, rankings)) #
test_60$Pool.QC
test_60$Garage.Cond <- as.integer(revalue(test_60$Garage.Cond,
rankings)) # Garage.Cond
test_60$Garage.Qual <- as.integer(revalue(test_60$Garage.Qual,
rankings)) # Garage.Qual
test_60$Fireplace.Qu <- as.integer(revalue(test_60$Fireplace.Qu,
rankings)) # Fireplace.Qu
test_60$Kitchen.Qual <- as.integer(revalue(test_60$Kitchen.Qual,
rankings)) # Kitchen.Qual
test_60$Heating.QC <- as.integer(revalue(test_60$Heating.QC, rankings))
# Heating.QC
test_60$Bsmt.Cond <- as.integer(revalue(test_60$Bsmt.Cond, rankings)) #
Bsmt.Cond
test_60$Bsmt.Qual <- as.integer(revalue(test_60$Bsmt.Qual, rankings)) #
Bsmt.Qual
test_60$Exter.Cond <- as.integer(revalue(test_60$Exter.Cond, rankings))
# Exter.Cond
test_60$Exter.Qual <- as.integer(revalue(test_60$Exter.Qual, rankings))
# Exter.Qual
rankings2 <- c("None" = 0,"Unf" = 1,"LwQ" = 2,"Rec" = 3,"BLQ" =
4,"ALQ"= 5,"GLQ" = 6) # basement ordinal variables
staging$BsmtFin.Type.1 <- as.integer(revalue(staging$BsmtFin.Type.1,
rankings2))
staging$BsmtFin.Type.2 <- as.integer(revalue(staging$BsmtFin.Type.2,
rankings2))
#test dataset encoding
test_60$BsmtFin.Type.1 <- as.integer(revalue(test_60$BsmtFin.Type.1,
rankings2))
test_60$BsmtFin.Type.2 <- as.integer(revalue(test_60$BsmtFin.Type.2,
rankings2))
# manual encodings with no ranking vectors
staging$Bsmt.Exposure<- as.integer(revalue(staging$Bsmt.Exposure,
c("None" = 0,"No" = 1,"Mn" = 2,"Av" = 3,"Gd" = 4))) # ranking for
basement exposure
staging$Land.Slope<- as.integer(revalue(staging$Land.Slope,
c("Sev"=0,"Mod"=1,"Gtl"=2))) # ranking for Land.Slope
```

```
staging$Lot.Shape<- as.integer(revalue(staging$Lot.Shape,
c("IR3"=0,"IR2"=1,"IR1"=2,"Reg"=3))) # ranking for Lot.Shape
staging$Functional<- as.integer(revalue(staging$Functional, c("Sal" =
0,"Sev" = 1,"Maj2" = 2,"Maj1" = 3,"Mod" =
4,"Min2"=5,"Min1"=6,"Typ"=7))) # ranking for Functional
staging$Paved.Drive<- as.integer(revalue(staging$Paved.Drive,
c("N"=0,"P"=1,"Y"=2))) # ranking for Paved.Drive
staging$Garage.Finish<- as.integer(revalue(staging$Garage.Finish,
c("None"=0,"Unf"=1,"RFn"=2,"Fin"=3))) # ranking for garage Finish
# test dataset encoding
test_60$Bsmt.Exposure <- as.integer(revalue(test_60$Bsmt.Exposure,
c("None" = 0,"No" = 1,"Mn" = 2,"Av" = 3,"Gd" = 4))) # ranking for
basement exposure
test_60$Land.Slope <- as.integer(revalue(test_60$Land.Slope,
c("Sev"=0,"Mod"=1,"Gtl"=2))) # ranking for Land.Slope
test_60$Lot.Shape <- as.integer(revalue(test_60$Lot.Shape,
c("IR3"=0,"IR2"=1,"IR1"=2,"Reg"=3))) # ranking for Lot.Shape
test_60$Functional <- as.integer(revalue(test_60$Functional, c("Sal" =
0,"Sev" = 1,"Maj2" = 2,"Maj1" = 3,"Mod" =
4,"Min2"=5,"Min1"=6,"Typ"=7))) # ranking for Functional
test_60$Paved.Drive <- as.integer(revalue(test_60$Paved.Drive,
c("N"=0,"P"=1,"Y"=2))) # ranking for Paved.Drive
test_60$Garage.Finish <- as.integer(revalue(test_60$Garage.Finish,
c("None"=0,"Unf"=1,"RFn"=2,"Fin"=3))) # ranking for garage Finish
#-----------------------------------Split_into_numeric_and_Factors-----
-------------------------------------------------#
# staging df
staging$MS.SubClass<- as.character(staging$MS.SubClass)   # it is
categorical with 16 factors
staging$Garage.Yr.Blt<- as.integer(staging$Garage.Yr.Blt) # years
staging$BsmtFin.SF.1 <- as.integer(staging$BsmtFin.SF.1)   # square
feet
staging$BsmtFin.SF.2 <- as.integer(staging$BsmtFin.SF.2)   # square
feet
staging$Lot.Frontage<- as.integer(staging$Lot.Frontage)   # integers
from numeric for the next step
staging$Mas.Vnr.Area<- as.integer(staging$Mas.Vnr.Area)   # integers
from numeric for the next step
# test dataset df
test_60$MS.SubClass <- as.character(test_60$MS.SubClass)   # it is
categorical with 16 factors
test_60$Garage.Yr.Blt<- as.integer(test_60$Garage.Yr.Blt) # years
test_60$BsmtFin.SF.1 <- as.integer(test_60$BsmtFin.SF.1)   # square
feet
test_60$BsmtFin.SF.2 <- as.integer(test_60$BsmtFin.SF.2)   # square
feet
test_60$Lot.Frontage <- as.integer(test_60$Lot.Frontage)   # integers
from numeric for the next step
test_60$Mas.Vnr.Area<- as.integer(test_60$Mas.Vnr.Area)   # integers
from numeric for the next step
#
num_staging<- staging[sapply(staging, is.integer)]   ### split the
integers
cat_staging<- staging[sapply(staging, is.character)] ### split the
characters
#
num_staging<- as.data.frame(lapply(num_staging, as.numeric)) # convert
integers to numeric
```

```
cat_staging<- as.data.frame(lapply(cat_staging, as.factor))  # convert
to factors
# test_60
test_60$Garage.Area<- as.integer(test_60$Garage.Area) # turn to integer
and then numeric
test_60$Garage.Cars<- as.integer(test_60$Garage.Cars) # turn to integer
and then numeric
#
num_test_60 <- test_60[sapply(test_60, is.integer)]   ### split the
integers
cat_test_60 <- test_60[sapply(test_60, is.character)] ### split the
characters
#
num_test_60<- as.data.frame(lapply(num_test_60, as.numeric)) # convert
integers to numeric
cat_test_60<- as.data.frame(lapply(cat_test_60, as.factor))  # convert
to factors
# last encodings staging df
staging$Central.Air<- as.integer(revalue(staging$Central.Air,
c("N"=0,"Y"=1))) # revalue into 0 and 1
cat_staging$Central.Air<- NULL
num_staging$Central.Air<- as.numeric(staging$Central.Air)
# last encodings test dataset
test_60$Central.Air <- as.integer(revalue(test_60$Central.Air,
c("N"=0,"Y"=1))) # revalue into 0 and 1
cat_test_60$Central.Air<- NULL
num_test_60$Central.Air<- as.numeric(test_60$Central.Air)
#-----------------------------drop categorical columns with frequency
above 98%------------------------------------------------------#
#---------------------------Street, Utilities, Condition.2, Roof.Matl,
Heating--------------------------------------------------#
library(summarytools)
freq(cat_staging, order ="freq") ### all columns
drop_list<- c("Street","Utilities","Condition.2","Roof.Matl","Heating")
freq(cat_staging[drop_list], order ="freq") ### drop with frequency
above 98%
cat_staging$Street<- NULL # 99.53
cat_staging$Utilities<- NULL # 99.86
cat_staging$Condition.2 <- NULL # 98.93
cat_staging$Roof.Matl<- NULL # 98.60
cat_staging$Heating<- NULL # 98.40
summary(num_staging)
summary(cat_staging)
str(num_staging)
str(cat_staging)
#------------------------------------Correlation_Matrix-----------------
---------------------------------------------------------#
library(dplyr)
library(Hmisc)
library(corrplot)
corr1 <- cor(num_staging, use="pairwise.complete.obs") # Pearson
correlations of all numeric variables
corr1_sorted <- as.matrix(sort(corr1[,'SalePrice'], decreasing = TRUE))
# sort correlations - SalePrice at top
CorHigh<- names(which(apply(corr1_sorted, 1, function(x) abs(x)>0.6 )))
#select only high corelations with PRICE
corr1 <- corr1[CorHigh, CorHigh]
#
```

```
par(mfrow=c(1,1))
corrplot.mixed(corr1, tl.col="black", tl.pos="lt")
par(mfrow=c(2,3)) # plots for variables most correlated with price.
plot(num_staging$SalePrice~num_staging$Garage.Area)
plot(num_staging$SalePrice~num_staging$X1st.Flr.SF)
boxplot(num_staging$SalePrice~num_staging$Kitchen.Qual)
boxplot(num_staging$SalePrice~num_staging$Garage.Cars)
plot(num_staging$SalePrice~num_staging$Total.Bsmt.SF)
boxplot(num_staging$SalePrice~num_staging$Exter.Qual)
# flattenCorrMatrixfunction - for seeing correlations
# cormat: matrix of the correlation coefficients
# pmat: matrix of the correlation p-values
flattenCorrMatrix<- function(cormat, pmat) {
ut<- upper.tri(cormat)
data.frame(
    row = rownames(cormat)[row(cormat)[ut]],
    column = rownames(cormat)[col(cormat)[ut]],
cor=(cormat)[ut],
    p = pmat[ut]
  )
}
#
corr_num1<- rcorr(as.matrix(num_staging)) # # CORRELATION matrix for
all pairs
correlation1<- flattenCorrMatrix(corr_num1$r, corr_num1$P) # will be
used later on
#
#----------------------------EXPLORATORY-DATA-ANALYSIS-PAIRWISE-
COMPARISONS----------------------------------------#
library(ggplot2)
#create histograms for numeric
par(mfrow=c(2,4))
p<-ncol(num_staging)
for (i in 1:p){
  hist(num_staging[,i], probability = TRUE,
main=names(num_staging)[i],border='black',col='lightblue')
  lines(density(num_staging[,i]), col=2)
  index <- seq( min(num_staging[,i]), max(num_staging[,i]),
length.out=100)
num_stagingnorm<- dnorm( index, mean=mean(num_staging[,i]),
sd(num_staging[,i]) )
lines( index, num_stagingnorm, col=3, num_staging=3, lwd=1)
}
# Simple Bar Plots
par(mfrow=c(2,3))
counts1 <- table(cat_staging$Electrical)
barplot(counts1, main="Electrical")
counts2 <- table(cat_staging$MS.Zoning)
barplot(counts2, main="MS.Zoning")
counts3 <- table(cat_staging$Sale.Condition)
barplot(counts3, main="Sale.Condition")
counts4 <- table(cat_staging$Sale.Type)
barplot(counts4, main="Sale.Type")
counts7 <- table(cat_staging$Garage.Type)
barplot(counts7, main="Garage.Type")
counts8 <- table(cat_staging$House.Style)
barplot(counts8, main="House.Style")
# Sale Price - plot
```

```
ggplot(data=num_staging, aes(x=SalePrice)) +
geom_histogram(fill="blue", binwidth= 10000) +
scale_x_continuous(breaks= seq(0, 800000, by=100000), labels =
scales::comma)
# Summary for Price
summary(num_staging$SalePrice)
# OVerallQuality
ggplot(data=num_staging, aes(x=factor(Overall.Qual), y=SalePrice))+
geom_boxplot(col='darkblue') + labs(x='Overall Quality') +
scale_y_continuous(breaks= seq(0, 800000, by=100000), labels =
scales::comma)
# Gr.Liv.Area
ggplot(data=num_staging, aes(x=Gr.Liv.Area, y=SalePrice))+
geom_point(col='darkblue') + labs(x='Gr.Liv.Area') +
scale_y_continuous(breaks= seq(0, 800000, by=100000), labels =
scales::comma)
#normality tests and plots for the above
library(nortest)
par(mfrow=c(1,3))
qqnorm(num_staging$SalePrice,main="QQ-plot SalePrice")
qqline(num_staging$SalePrice, col="steelblue", lwd=2)
lillie.test(num_staging$SalePrice)
shapiro.test(num_staging$SalePrice)
qqnorm(num_staging$Gr.Liv.Area, main="QQ-plot Gr.Liv.Area")
qqline(num_staging$Gr.Liv.Area, col="steelblue", lwd=2)
lillie.test(num_staging$Gr.Liv.Area)
shapiro.test(num_staging$Gr.Liv.Area)
hist(num_staging$Overall.Qual, main="Histogram Overall.Qual",
probability=TRUE)
index <- seq( min(num_staging$Overall.Qual),
max(num_staging$Overall.Qual),
length.out=100)
ynorm<- dnorm( index, mean=mean(num_staging$Overall.Qual),
sd(num_staging$Overall.Qual) )
lines( index, ynorm, col=3, lty=3, lwd=3 )
lillie.test(num_staging$Overall.Qual)
shapiro.test(num_staging$Overall.Qual)
# Garage variable plots
par(mfrow=c(2,3))
plot(num_staging$SalePrice ~num_staging$Garage.Area, num_staging)
boxplot(num_staging$SalePrice ~num_staging$Garage.Finish, num_staging)
boxplot(num_staging$SalePrice ~num_staging$Garage.Cond, num_staging)
boxplot(num_staging$SalePrice ~num_staging$Garage.Qual, num_staging)
boxplot(num_staging$SalePrice ~num_staging$Garage.Cars, num_staging)
counts5 <- table(cat_staging$Garage.Type)
barplot(counts5, main="Garage.Type",horiz=FALSE, names.arg=c("2Types",
"Attchd", "Basment","BuiltIn",
"CarPort","Detchd","None"),cex.names=0.5)# Basement variables plots
par(mfrow=c(2,3))
boxplot(num_staging$SalePrice ~num_staging$Bsmt.Qual, num_staging,
horizontal = FALSE)
boxplot(num_staging$SalePrice ~num_staging$Bsmt.Cond, num_staging,
horizontal = FALSE)
boxplot(num_staging$SalePrice ~num_staging$Bsmt.Exposure, num_staging,
horizontal = FALSE)
boxplot(num_staging$SalePrice ~ num_staging$BsmtFin.Type.1,
num_staging, horizontal = FALSE)
```

```
plot(num_staging$SalePrice ~ num_staging$BsmtFin.SF.2, num_staging,
horizontal = FALSE)
#
library(gridExtra)
# Neighborhood PLOTTING
p1<-ggplot(data=staging, aes(x=reorder(Neighborhood, SalePrice,
FUN=median), y=SalePrice)) +
geom_bar(stat="summary", fill="steelblue")+
theme(axis.text.x= element_text(angle = 45, hjust = 1))+
scale_y_continuous(breaks= seq(0, 800000, by=50000), labels =
scales::comma)+
geom_hline(yintercept= median(staging$SalePrice), linetype="dashed",
color="red")

p2 <- ggplot(data=staging, aes(x=reorder(Neighborhood, SalePrice,
FUN=median))) +
geom_histogram(stat='count')+
geom_label(stat ="count", aes(label = ..count.., y = ..count..),
size=3)+
theme(axis.text.x= element_text(angle = 45, hjust = 1))

grid.arrange(p1, p2) # note we will further bin the neighborhoods -
justification above plot
#  MS.SubClassPLOTING
p3<-ggplot(data=staging, aes(x=reorder(MS.SubClass, SalePrice,
FUN=median), y=SalePrice)) +
geom_bar(stat="summary", fill="steelblue")+
theme(axis.text.x= element_text(angle = 45, hjust = 1))+
scale_y_continuous(breaks= seq(0, 800000, by=50000), labels =
scales::comma)+
geom_hline(yintercept= median(staging$SalePrice), linetype="dashed",
color="red")
p4 <- ggplot(data=staging, aes(x=reorder(MS.SubClass, SalePrice,
FUN=median))) +
geom_histogram(stat='count')+
geom_label(stat ="count", aes(label = ..count.., y = ..count..),
size=3)+
theme(axis.text.x= element_text(angle = 45, hjust = 1))
grid.arrange(p3, p4)
# PAIRWISE-COMPARISONS for these variables
library(GGally)
pairwise_list<- c("SalePrice","Overall.Qual","Kitchen.Qual",
"Bsmt.Exposure","Total.Bsmt.SF","Garage.Cars","Garage.Area")
pairwise_df<- num_staging[pairwise_list]
ggpairs(pairwise_df)
# boxplots for the pairwise_list
par(mfrow=c(1,3))
boxplot(num_staging$SalePrice ~num_staging$Overall.Qual, num_staging)
boxplot(num_staging$SalePrice ~num_staging$Kitchen.Qual, num_staging)
boxplot(num_staging$SalePrice ~num_staging$Exter.Qual, num_staging)
par(mfrow=c(1,3))
boxplot(num_staging$SalePrice ~num_staging$Exter.Qual, num_staging)
boxplot(num_staging$SalePrice ~num_staging$Garage.Cars, num_staging)
counts5 <- table(cat_staging$Garage.Type)
#--------------------------PREPARE_DATA_FOR_MODEL_BUILDING------------
----------------------------------------#
View(correlation1) # correlation table for numerics - see line 224
```

```
# decide to drop some variables due to high correlation with each other
in order to neutralize the
# colinearityeffect - we keep the variables thar are higher correlated
with price
# we keep the high correlated variables with the price and
# we drop for garage and basement the most variables in order to deal
with multi-colinearity
num_staging$Garage.Cond<- NULL
num_staging$Garage.Qual<- NULL
num_staging$Garage.Yr.Blt<- NULL
num_staging$Garage.Area<- NULL   # i kept the garage.cars and drop this
num_staging$Bsmt.Cond<- NULL
num_staging$Bsmt.Exposure<- NULL
num_staging$Bsmt.Full.Bath<- NULL
num_staging$Bsmt.Half.Bath<- NULL
num_staging$BsmtFin.SF.1 <- NULL
num_staging$BsmtFin.SF.2 <- NULL
num_staging$BsmtFin.Type.1 <- NULL
num_staging$BsmtFin.Type.2 <- NULL
num_staging$Bsmt.Unf.SF<- NULL

#-----------------------------Bining_Further_the_Neighborhood_Variable--
-----------------------------------------#
rich <- c("StoneBr","NridgHt","NoRidge")
poor <- c("MeadowV","BrDale","IDOTRR")
# staging dataset
cat_staging$Binhood[cat_staging$Neighborhood%in% rich ] <- 2
cat_staging$Binhood[!cat_staging$Neighborhood%in% c(rich,poor) ] <- 1
cat_staging$Binhood[cat_staging$Neighborhood%in% poor] <- 0
cat_staging$Binhood<- as.factor(cat_staging$Binhood)
cat_staging$Neighborhood<- NULL # delete the old column
#test dataset
cat_test_60$Binhood[cat_test_60$Neighborhood %in% rich ] <- 2
cat_test_60$Binhood[!cat_test_60$Neighborhood %in% c(rich,poor) ] <- 1
cat_test_60$Binhood[cat_test_60$Neighborhood %in% poor] <- 0
cat_test_60$Binhood <- as.factor(cat_test_60$Binhood)
cat_test_60$Neighborhood <- NULL # delete the old column
#
library(dummies) # create dummies in order to insert data into the
variable selection algorithms
dummies1 <- dummy.data.frame(cat_staging, sep =".", all = FALSE)
dummies_test_60 <- dummy.data.frame(cat_test_60, sep =".", all = FALSE)
corr_dummies<- rcorr(as.matrix(dummies1))
correlation2<- flattenCorrMatrix(corr_dummies$r, corr_dummies$P)
View(correlation2) # no further filtering because Lasso will deal with
multicolinearity
# create of staging2 and test_final
staging2 <- data.frame(dummies1, num_staging)
test_final<- data.frame(dummies_test_60,num_test_60)
write.csv(staging2,"staging2.csv") # create new csv for staging2 186
variables
write.csv(test_final,"final_test") # create the final dataset for the
out of sample prediction
#---------------------------------------LASSO---------------------------
-----------------------------------------#
staging2<-read.csv(file.choose(),sep=",") # load staging2 csv
par(mfrow=c(1,1))
require(glmnet)
```

```
X <- model.matrix(SalePrice~., staging2 )[,-staging2$SalePrice]
lasso <- glmnet(X, staging2$SalePrice)
plot(lasso, xvar="lambda", label = T)
lasso1 <- cv.glmnet(X, staging$SalePrice, alpha = 1)
plot(lasso1)
plot(lasso1$glmnet.fit, xvar="lambda")
abline(v=log(c(lasso1$lambda.min, lasso1$lambda.1se)), lty=2)
c<-coef(lasso1,s='lambda.1se',exact=TRUE)
inds<-which(c!=0)
variables<-row.names(c)[inds]
variables<-variables[variables%nin%'(Intercept)']
variables
length(variables)
lasso1$lambda.1se # 3896.063 we choose this because it has less penalty
for the same level of error
#-----------------------------------------------------------------------
-----------------------------------------#
#
"Land.Contour.HLS""Lot.Config.CulDSac""Bldg.Type.1Fam""Binhood.2""Lot.A
rea"
#[6]
"Overall.Qual""Year.Built""Year.Remod.Add""Mas.Vnr.Area""Exter.Qual"
#[11]
"Bsmt.Qual""Total.Bsmt.SF""X1st.Flr.SF""Gr.Liv.Area""Kitchen.Qual"
#[16] "Fireplaces""Garage.Finish""Garage.Cars""Wood.Deck.SF"
finaldata<- staging2[variables]
finaldata$SalePrice<- staging2$SalePrice
write.csv(finaldata,"final_data.csv") # data to use for model
prediction
                                      # save this variables into a new
df named final_data
                                      # and procceed to the model
building
#-----------------------------------------------------------------------
-----------------------------------------#
finaldata<-read.csv(file.choose(),sep=",") # load final_data csv
finaldata<- as.data.frame(lapply(finaldata, as.numeric)) # convert into
numeric
model_step<- lm(SalePrice~. -X, data = finaldata) # all variables
step_both<- step(model_step, direction ="both") # lowest AIC rate
30870.93
step_back<- step(model_step, direction ="backward") #lowest AIC rate
30870.93
#-----------------------------------------------------------------------
-----------------------------------------#
model1 <- lm(SalePrice ~Land.Contour.HLS + Lot.Config.CulDSac +
Bldg.Type.1Fam +
               Binhood.2 + Lot.Area+ Overall.Qual + Year.Built +
Year.Remod.Add +
Mas.Vnr.Area+ Exter.Qual + Bsmt.Qual + Total.Bsmt.SF + X1st.Flr.SF +
Gr.Liv.Area+ Kitchen.Qual + Fireplaces + Garage.Cars + Wood.Deck.SF,
data = finaldata)
summary(model1) # R2 0.8634 - pvalues ok
# check normality of the residuals - plots and tests
par(mfrow=c(2,2))
plot(model1, which=2)
library(nortest)
lillie.test(residuals(model1))
```

```
shapiro.test(residuals(model1))
# check constant variance with plots - Homoscedasticity
Stud.residuals<-rstudent(model1)
yhat<- fitted(model1)
plot(yhat, Stud.residuals) # stud.residuals
abline(h=c(-2,2), col=2, lty=2)
# check constant variance with tests
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)
# NON LINEARITY
library(car)
residualPlot(model1, type='rstudent')
# Independence of errors
plot(rstudent(model1), type='l')
durbinWatsonTest(model1)
#-----------------------------------------------------------------------
----------------------------------------#
model2 <- lm(log(SalePrice) ~Land.Contour.HLS + Lot.Config.CulDSac +
Bldg.Type.1Fam +
               Binhood.2 + Lot.Area+ Overall.Qual + Year.Built +
Year.Remod.Add +
Mas.Vnr.Area+ Exter.Qual + Bsmt.Qual + Total.Bsmt.SF + X1st.Flr.SF +
Gr.Liv.Area+ Kitchen.Qual + Fireplaces + Garage.Cars + Wood.Deck.SF,
data = finaldata)
summary(model2) # drop Lot.Config.CulDSac - Mas.Vnr.Area - Exter.Qual
#-----------------------------------------------------------------------
----------------------------------------#
model2_log <- lm(log(SalePrice) ~Land.Contour.HLS  + Bldg.Type.1Fam +
                       Binhood.2 + Lot.Area+ Overall.Qual + Year.Built
+ Year.Remod.Add
                     + Bsmt.Qual+ Total.Bsmt.SF + X1st.Flr.SF +
Gr.Liv.Area+ Kitchen.Qual + Fireplaces + Garage.Cars + Wood.Deck.SF,
data = finaldata)
summary(model2_log) # p-values ok - R2 adj 0.87
# check normality of the residuals - plots and tests
par(mfrow=c(2,2))
plot(model2_log, which=2)
library(nortest)
lillie.test(residuals(model2_log))
shapiro.test(residuals(model2_log))
# check constant variance with plots - Homoscedasticity
Stud.residuals<-rstudent(model2_log)
yhat<- fitted(model2_log)
plot(yhat, Stud.residuals) # stud.residuals
abline(h=c(-2,2), col=2, lty=2)
# check constant variance with tests
library(car)
ncvTest(model2_log)
yhat.quantiles<-cut(yhat, breaks=quantile(yhat, probs=seq(0,1,0.25)),
dig.lab=6)
table(yhat.quantiles)
leveneTest(rstudent(model2_log)~yhat.quantiles)
# NON LINEARITY
library(car)
```

```
residualPlot(model2_log, type='rstudent')
# Independence of errors
plot(rstudent(model2_log), type='l')
durbinWatsonTest(model2_log)
#-----------------------------------------------------------------------
----------------------------------------#
residualPlots(model2_log) # use polynomials to the statistical
significant terms
model3_poly <- lm(log(SalePrice) ~ + Bldg.Type.1Fam +
                    Binhood.2 + poly(Lot.Area,2) + poly(Overall.Qual,2)
+ poly(Year.Built,2)+
Bsmt.Qual+ poly(Total.Bsmt.SF,2) + poly(X1st.Flr.SF,2) +
poly(Gr.Liv.Area,2) +
Kitchen.Qual+ Fireplaces + poly(Garage.Cars,2) + Wood.Deck.SF, data =
finaldata)
summary(model3_poly)
# check normality of the residuals - plots and tests
par(mfrow=c(2,2))
plot(model3_poly, which=2)
library(nortest)
lillie.test(residuals(model3_poly))
shapiro.test(residuals(model3_poly))
# check constant variance with plots - Homoscedasticity
Stud.residuals<-rstudent(model3_poly)
yhat<- fitted(model3_poly)
plot(yhat, Stud.residuals) # stud.residuals
abline(h=c(-2,2), col=2, lty=2)
# check constant variance with tests
library(car)
ncvTest(model3_poly)
yhat.quantiles<-cut(yhat, breaks=quantile(yhat, probs=seq(0,1,0.25)),
dig.lab=6)
table(yhat.quantiles)
leveneTest(rstudent(model3_poly)~yhat.quantiles)
# NON LINEARITY
library(car)
residualPlot(model3_poly, type='rstudent')
# Independence of errors
plot(rstudent(model3_poly), type='l')
durbinWatsonTest(model3_poly)
#-----------------------------------------------------------------------
-----------------------------------------#
#MODEL_TRAINNING_WITH LOOCV AND 10FOLD_CV
library(caret)
#model1
#use 10 fold cross validation to evaluate model1
set.seed(1)
train_control_CV<- trainControl(method ="CV", number = 10)
model1_cv10 <- train(SalePrice ~Land.Contour.HLS + Lot.Config.CulDSac +
Bldg.Type.1Fam +
                       Binhood.2 + Lot.Area+ Overall.Qual + Year.Built
+ Year.Remod.Add +
Mas.Vnr.Area+ Exter.Qual + Bsmt.Qual + Total.Bsmt.SF + X1st.Flr.SF +
Gr.Liv.Area+ Kitchen.Qual + Fireplaces + Garage.Cars + Wood.Deck.SF,
                     data = finaldata, method="lm", trControl =
train_control_CV)
model1_cv10$results
print(model1_cv10) # RMSE = 29594.54
```

```
model1_cv10$finalModel
# Leave one out cross validation
# defining training control
# as Leave One Out Cross Validation
train_control_LOOCV<- trainControl(method ="LOOCV")
# training the model
model1_LOOCV <- train(SalePrice ~Land.Contour.HLS + Lot.Config.CulDSac
+ Bldg.Type.1Fam +
                        Binhood.2 + Lot.Area+ Overall.Qual + Year.Built
+ Year.Remod.Add +
Mas.Vnr.Area+ Exter.Qual + Bsmt.Qual + Total.Bsmt.SF + X1st.Flr.SF +
Gr.Liv.Area+ Kitchen.Qual + Fireplaces + Garage.Cars + Wood.Deck.SF,
                      data = finaldata, method="lm", trControl =
train_control_LOOCV)
# printing model performance metrics
# along with other details
model1_LOOCV$results
print(model1_LOOCV) #  RMSE = 29783.59
model1_LOOCV$finalModel
#-----------------------------------------------------------------------
-----------------------------------------#
#model2_log
set.seed(1)
train_control_CV<- trainControl(method ="CV", number = 10)
model2_cv10 <- train(log(SalePrice) ~Land.Contour.HLS  + Bldg.Type.1Fam
+
                       Binhood.2 + Lot.Area+ Overall.Qual + Year.Built
+ Year.Remod.Add
                     + Bsmt.Qual+ Total.Bsmt.SF + X1st.Flr.SF +
Gr.Liv.Area+ Kitchen.Qual + Fireplaces + Garage.Cars + Wood.Deck.SF,
                     data = finaldata, method="lm", trControl =
train_control_CV)
model2_cv10$results
print(model2_cv10) # RMSE =  0.1478678
model2_cv10$finalModel
# Leave one out cross validation
# defining training control
# as Leave One Out Cross Validation
train_control_LOOCV<- trainControl(method ="LOOCV")
# training the model
model2_LOOCV <- train(log(SalePrice) ~Land.Contour.HLS  +
Bldg.Type.1Fam +
                        Binhood.2 + Lot.Area+ Overall.Qual + Year.Built
+ Year.Remod.Add
                      + Bsmt.Qual+ Total.Bsmt.SF + X1st.Flr.SF +
Gr.Liv.Area+ Kitchen.Qual + Fireplaces + Garage.Cars + Wood.Deck.SF,
                      data = finaldata, method="lm", trControl =
train_control_LOOCV)
# printing model performance metrics
# along with other details
model2_LOOCV$results
print(model2_LOOCV) #  RMSE = 0.1494884
model2_LOOCV$finalModel
#-----------------------------------------------------------------------
-----------------------------------------#
#model3_poly
#use 10 fold cross validation to evaluate model3
set.seed(1)
```

```
train_control_CV<- trainControl(method ="CV", number = 10)
model3_cv10 <- train(log(SalePrice) ~ + Bldg.Type.1Fam +
                       Binhood.2 + poly(Lot.Area,2) +
poly(Overall.Qual,2) + poly(Year.Built,2)+
Bsmt.Qual+ poly(Total.Bsmt.SF,2) + poly(X1st.Flr.SF,2) +
poly(Gr.Liv.Area,2) +
Kitchen.Qual+ Fireplaces + poly(Garage.Cars,2) + Wood.Deck.SF,
                     data = finaldata, method="lm", trControl =
train_control_CV)
model3_cv10$results
print(model3_cv10) # RMSE =   0.1381455
model3_cv10$finalModel
# Leave one out cross validation
# defining training control
# as Leave One Out Cross Validation
train_control_LOOCV<- trainControl(method ="LOOCV")
# training the model
model3_LOOCV <- train(log(SalePrice) ~ + Bldg.Type.1Fam +
                        Binhood.2 + poly(Lot.Area,2) +
poly(Overall.Qual,2) + poly(Year.Built,2)+
Bsmt.Qual+ poly(Total.Bsmt.SF,2) + poly(X1st.Flr.SF,2) +
poly(Gr.Liv.Area,2) +
Kitchen.Qual+ Fireplaces + poly(Garage.Cars,2) + Wood.Deck.SF,
                      data = finaldata, method="lm", trControl =
train_control_LOOCV)
# printing model performance metrics
# along with other details
model3_LOOCV$results
print(model3_LOOCV) #  RMSE = 0.1384827
model3_LOOCV$finalModel
train_results<- rbind(model1_cv10$results[1:4],model1_LOOCV$results,
                         model2_cv10$results[1:4],model2_LOOCV$results,
                         model3_cv10$results[1:4],model3_LOOCV$results)
rownames(train_results)<-
c("model1_cv10","model1_LOOCV","model2_cv10","model2_LOOCV",
"model3_cv10","model3_LOOCV")
train_results
model3_LOOCV$finalModel
# ----------------------------------------------------------------------
-----------------------------------------#
# Predict - model out of sample Accuracy_with_test_dataset-------------
-----------------------------------------#
final_test<-read.csv(file.choose(),sep=",") # load the final_test
dataset (converted common test dataset)
summary(model3_poly)
pred3_poly<-as.data.frame(predict(model3_poly, newdata= final_test,
interval ="prediction"))
pred_table<- as.data.frame(cbind(final_test$SalePrice,
exp(pred3_poly$fit)))
colnames(pred_table)<-c("Actual_Values","Predicted_Values")
pred_table$diff<- abs(pred_table$Actual_Values-
pred_table$Predicted_Values)
summary(pred_table$diff)
var(pred_table$diff) # 671069367
sd(pred_table$diff) # 25905.01
library(ggplot2)
ggplot(pred_table, aes(x=Predicted_Values, y=Actual_Values)) +
geom_point(size=3, shape=20, color="black") +
```

```
geom_smooth(method = lm,
linetype="dashed",color="blue",fill="darkgrey")+
geom_abline(intercept=0, slope=1, color="red")
geom_text(label=rownames(pred_table))
```