Linear Regression Business Report

Anant Agarwal, Arpita Jena, Asmita Vikas, Deena Liz John

10/3/2017

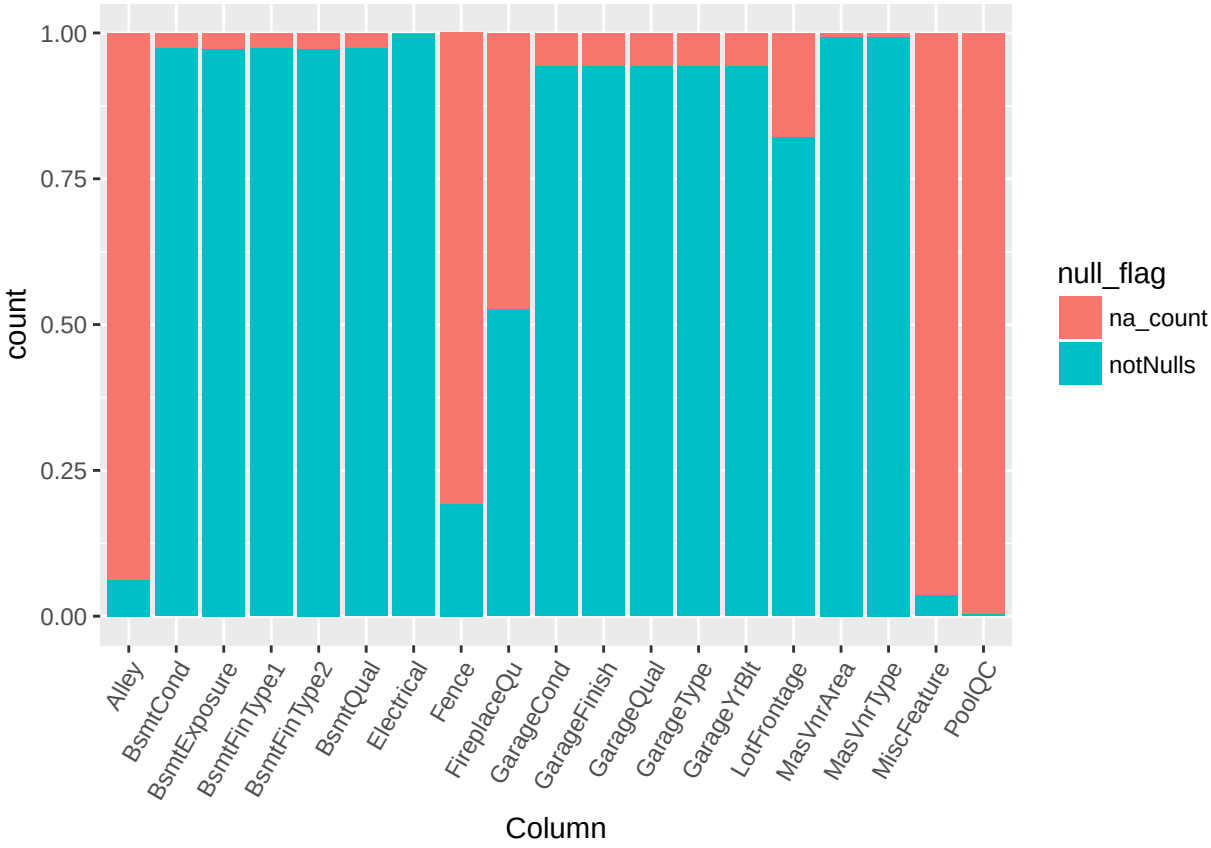
Part 1: Exploratory Data Analysis

Let us try and understand how our data looks like, before we do any analysis or try to make inferences.

1.1 Loading the data into a dataframe

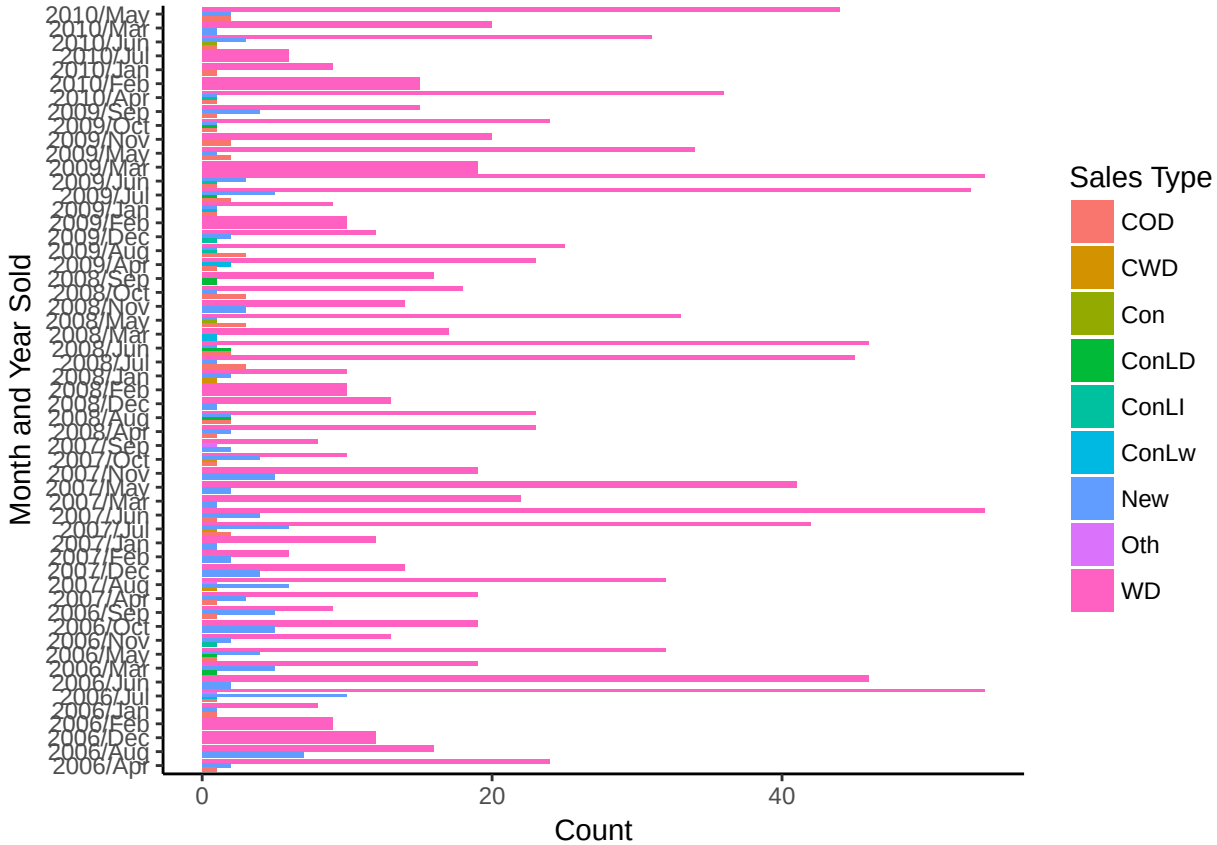
1.2 Check for NAs

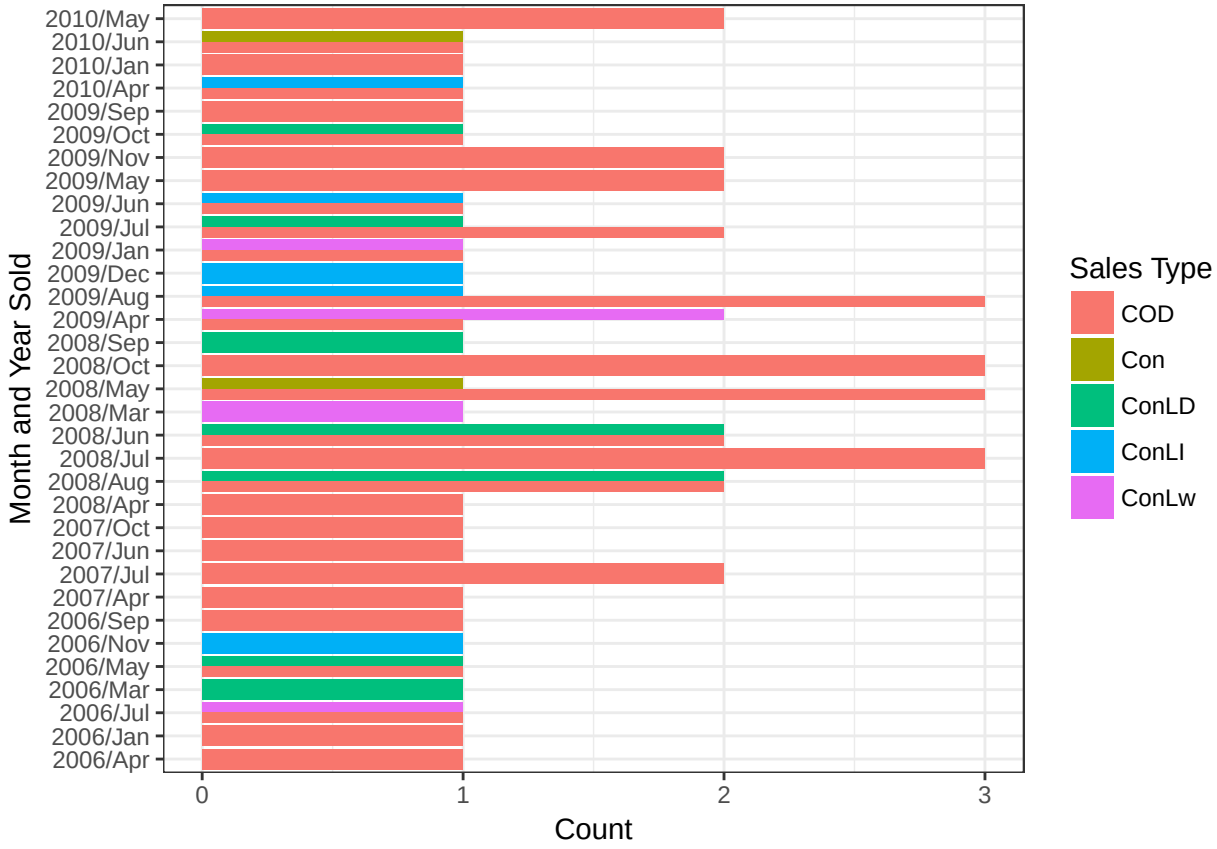
```
## Warning: Deprecated, use tibble::rownames_to_column() instead.
```



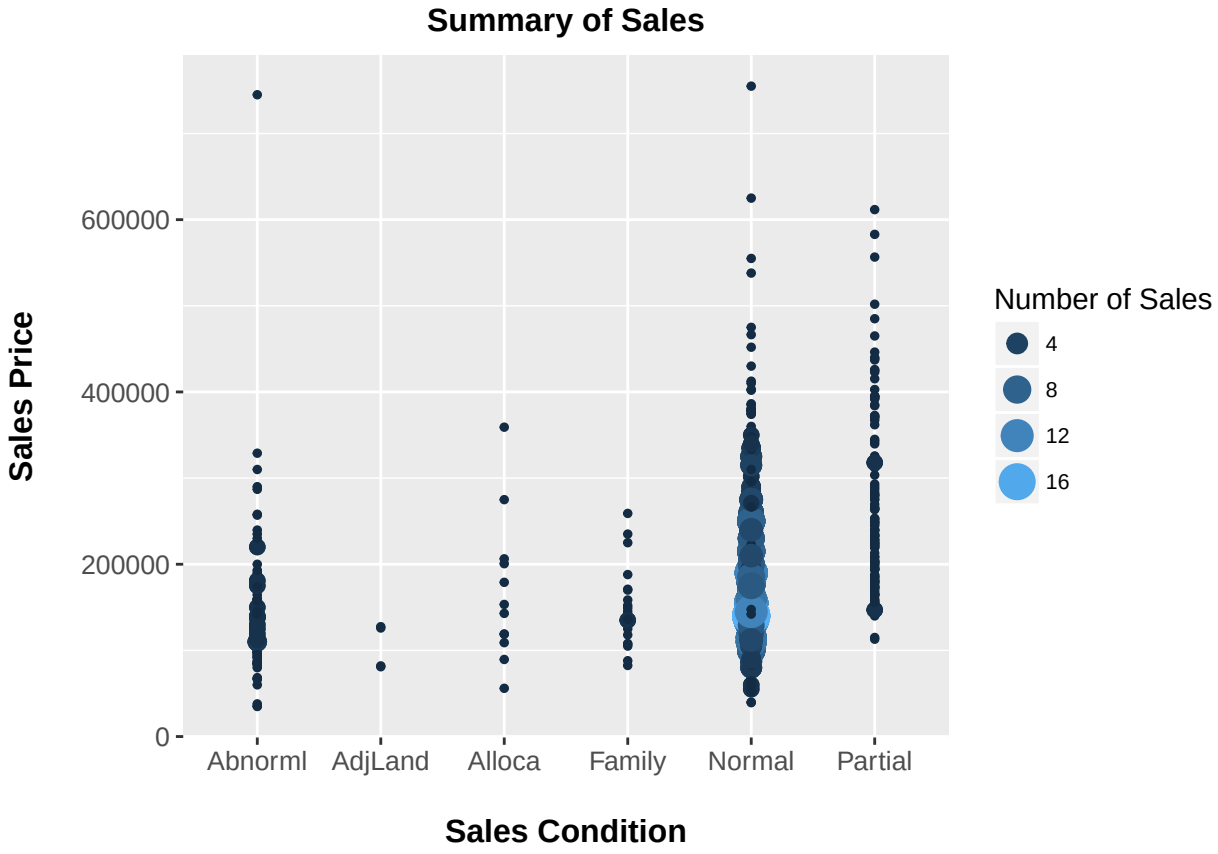
Wow! too many NAs. Going back to the data description, we see that for most of the catagorical values, a NA signifies that facility unavailable on the site. Example, PoolQC = NA means no pool. We will substitute them with Not Present later.

1.3 Sales Type per month

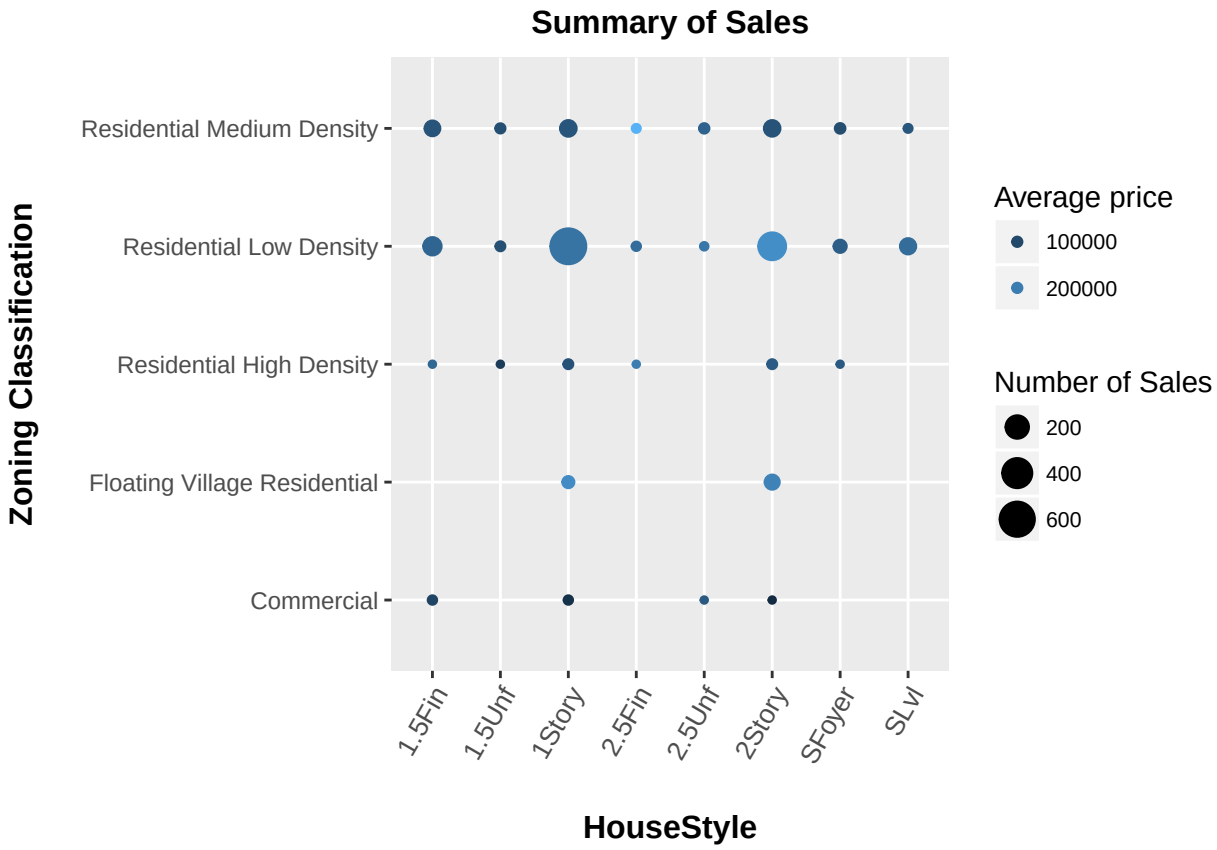




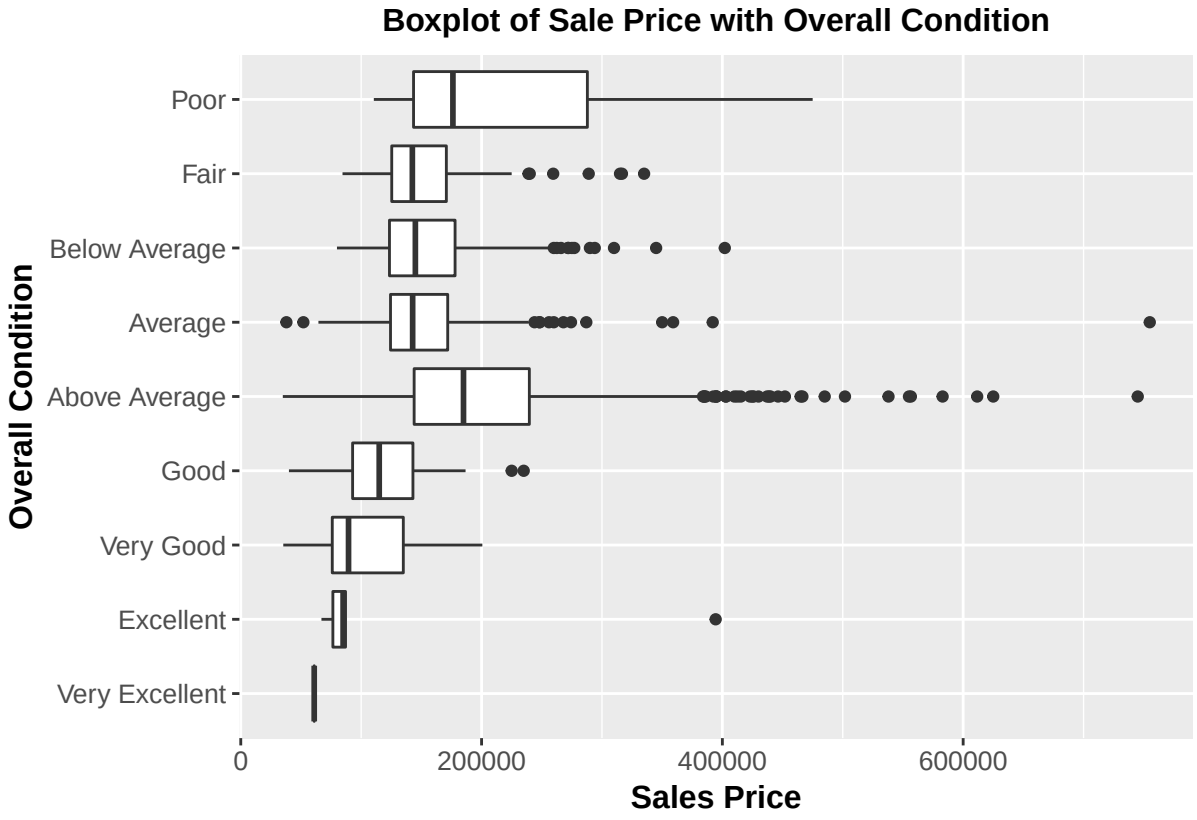
1.4 Sales price with sales condition



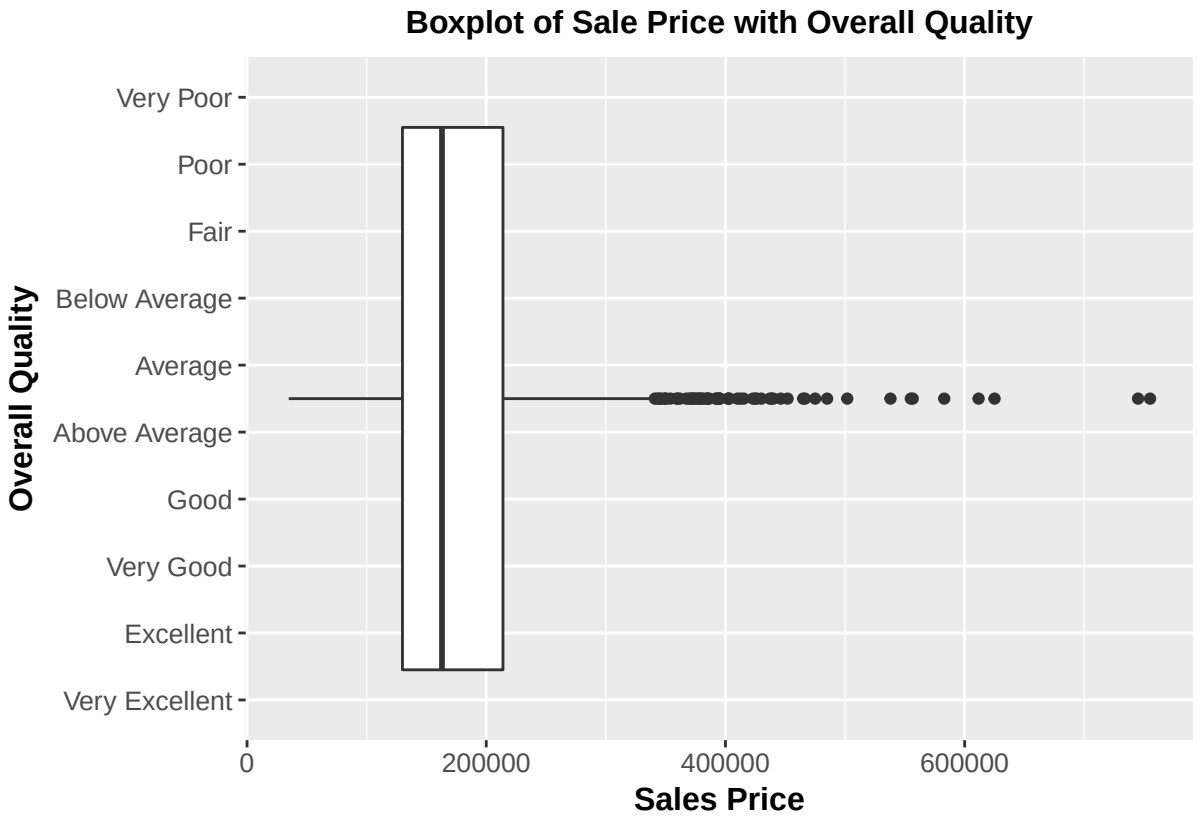
1.5 Plot of zone classification, with sales price and number


Summary of Sales
Zoning Classification
Residential Medium Density
Residential Low Density
Residential High Density
Floating Village Residential
Commercial
1.5Fin
1.5Unf
1Story
2.5Fin
2.5Unf
2Story
SFoyer
SLvl
HouseStyle
Average price
100000
200000
Number of Sales
200
400
600

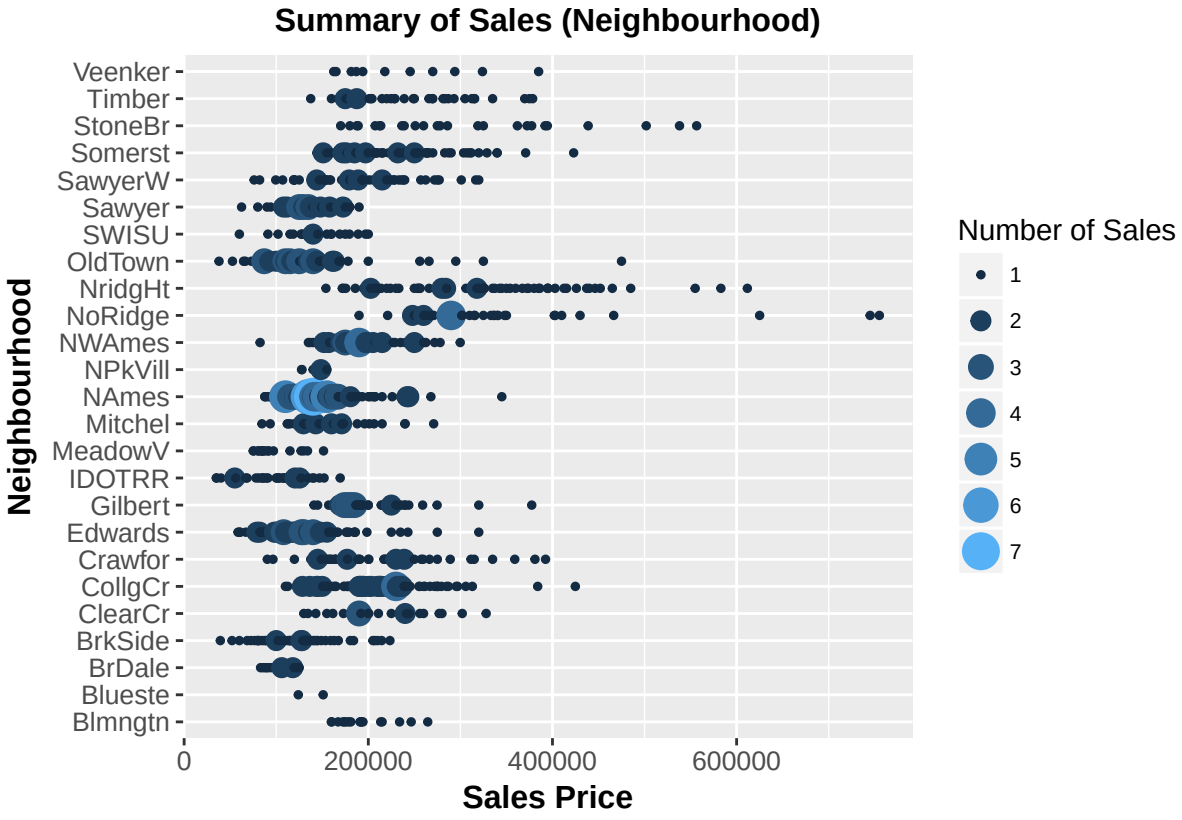
1.6 Boxplot of sales price with overall condition of the house



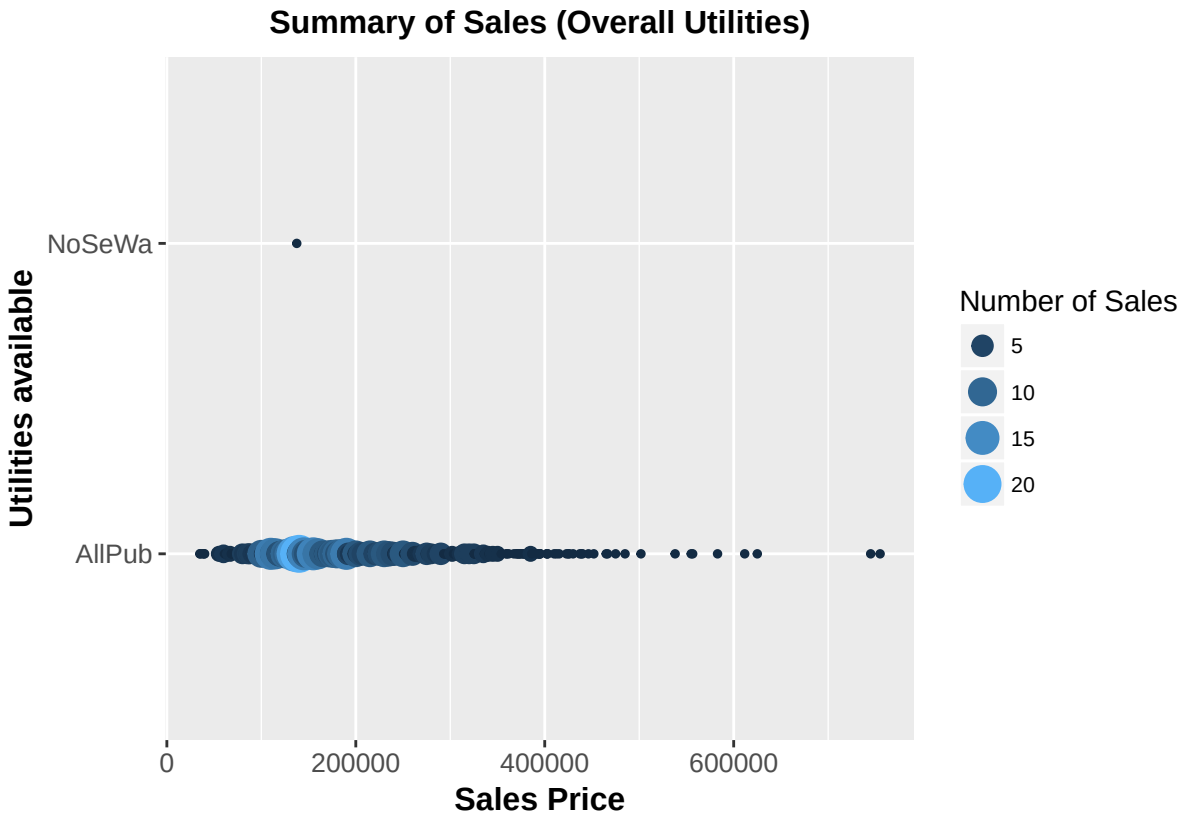
1.7 Boxplot of sales price with overall quality of material and finish of house



1.8 Dot plot of neighbourhood with sales price and number of sales

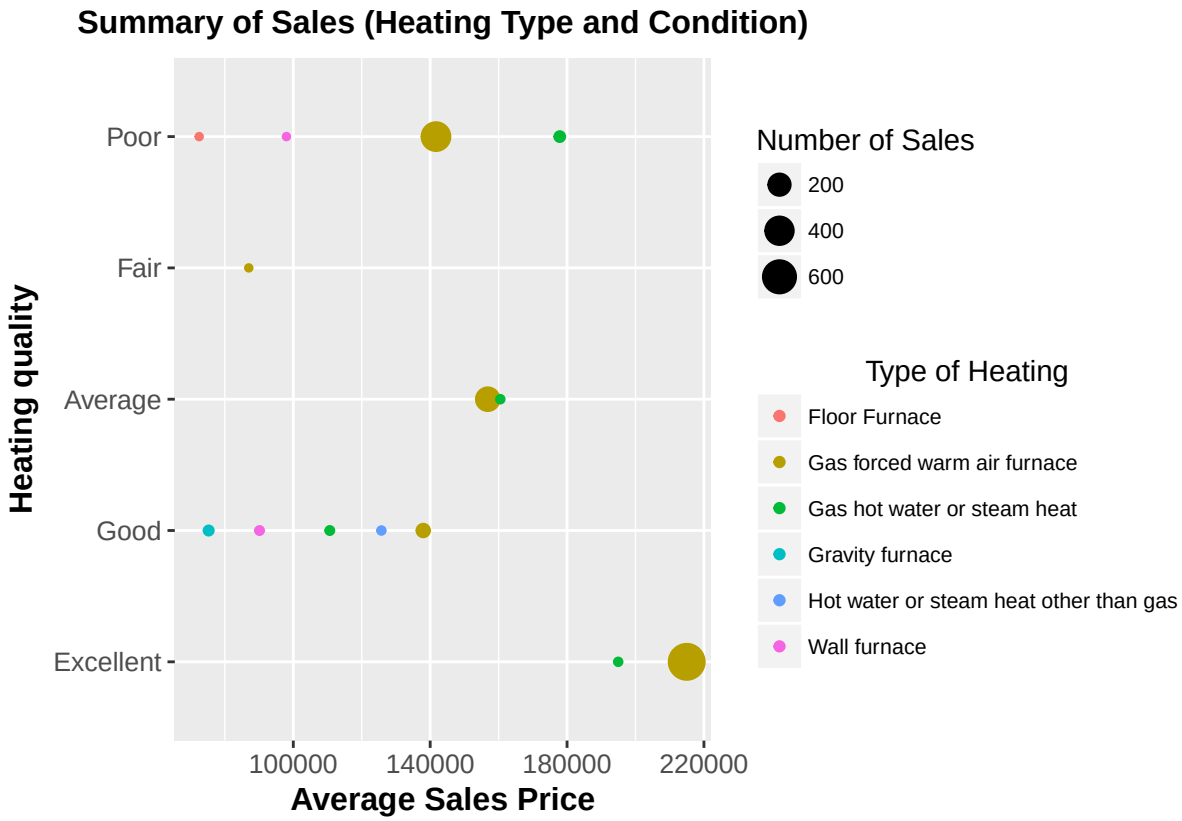


1.9 Dot plot of utilities available with sales price and number of sales

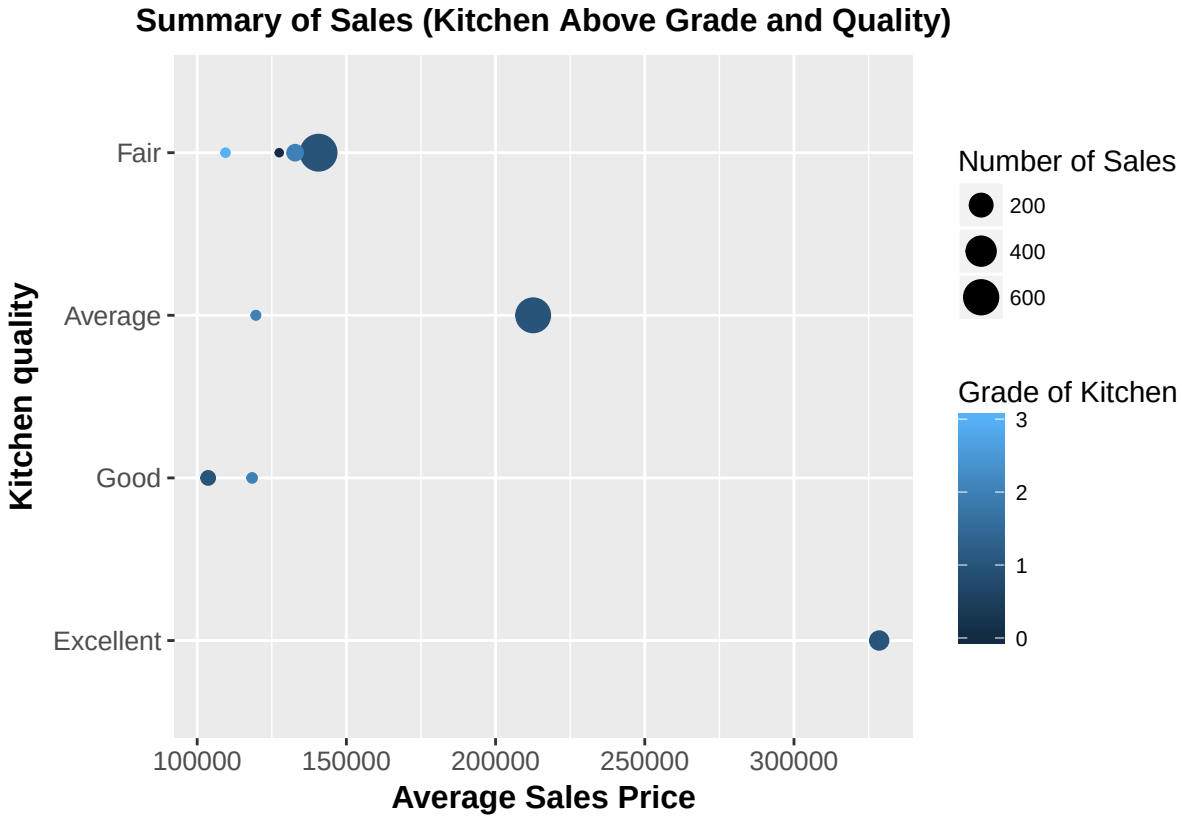


1.10 Dot plot of heating type and quality with sales price and number of sales

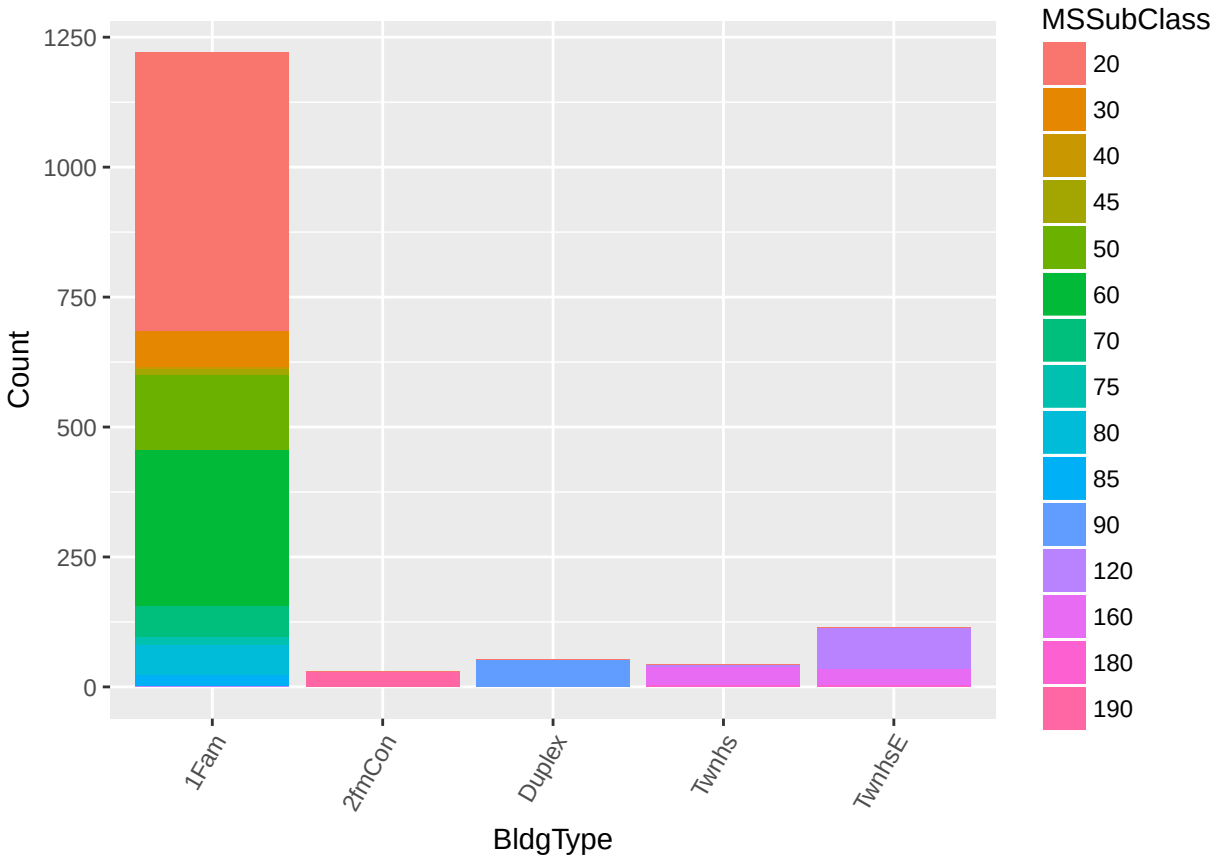
```
## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font
## metrics unknown for character 0x9

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font
## metrics unknown for character 0x9

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font width
## unknown for character 0x9

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## font width unknown for character 0x9

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## font width unknown for character 0x9

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x
## $y, : font width unknown for character 0x9
```


Summary of Sales (Heating Type and Condition)
Heating quality
Poor
Fair
Average
Good
Excellent
100000
140000
180000
220000
Average Sales Price
Number of Sales
200
400
600
Type of Heating
Floor Furnace
Gas forced warm air furnace
Gas hot water or steam heat
Gravity furnace
Hot water or steam heat other than gas
Wall furnace

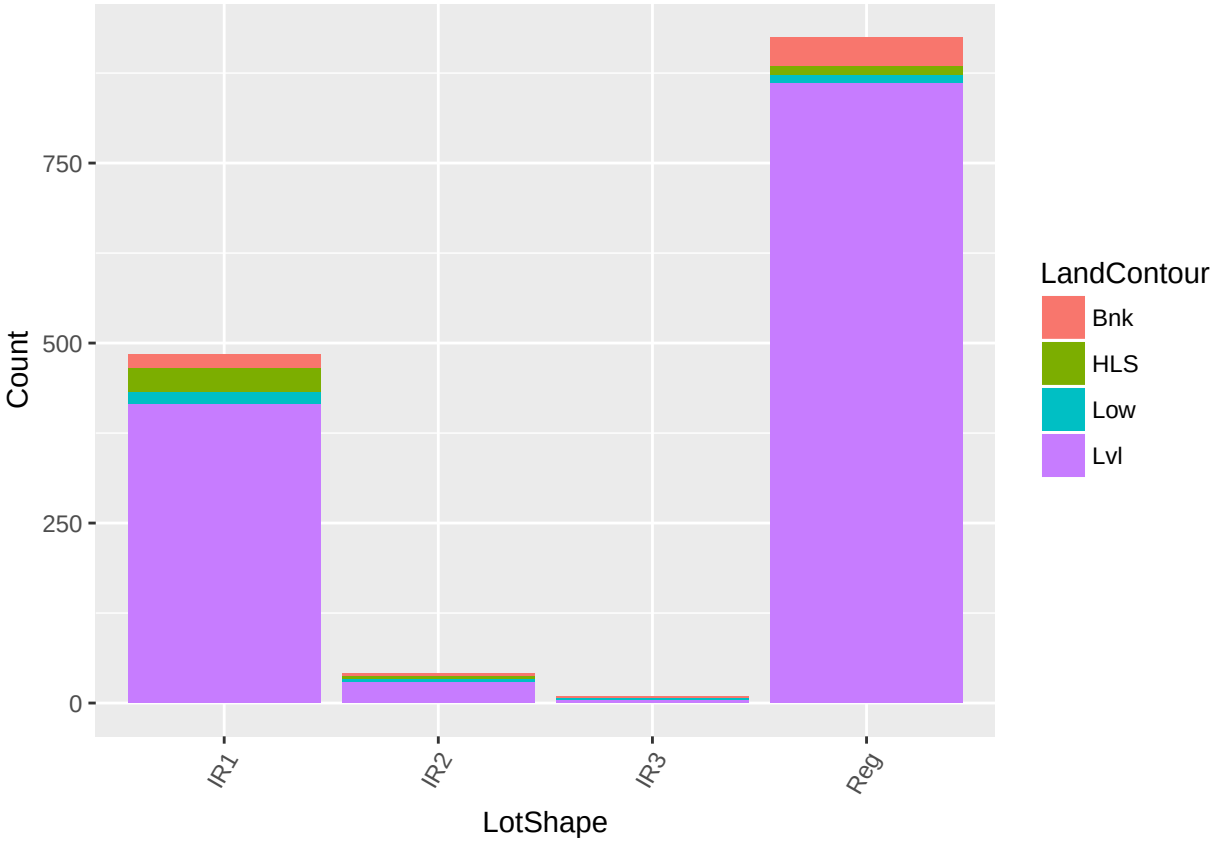
1.11 Dot plot of kitchens above grade and quality with sales price and number of sales



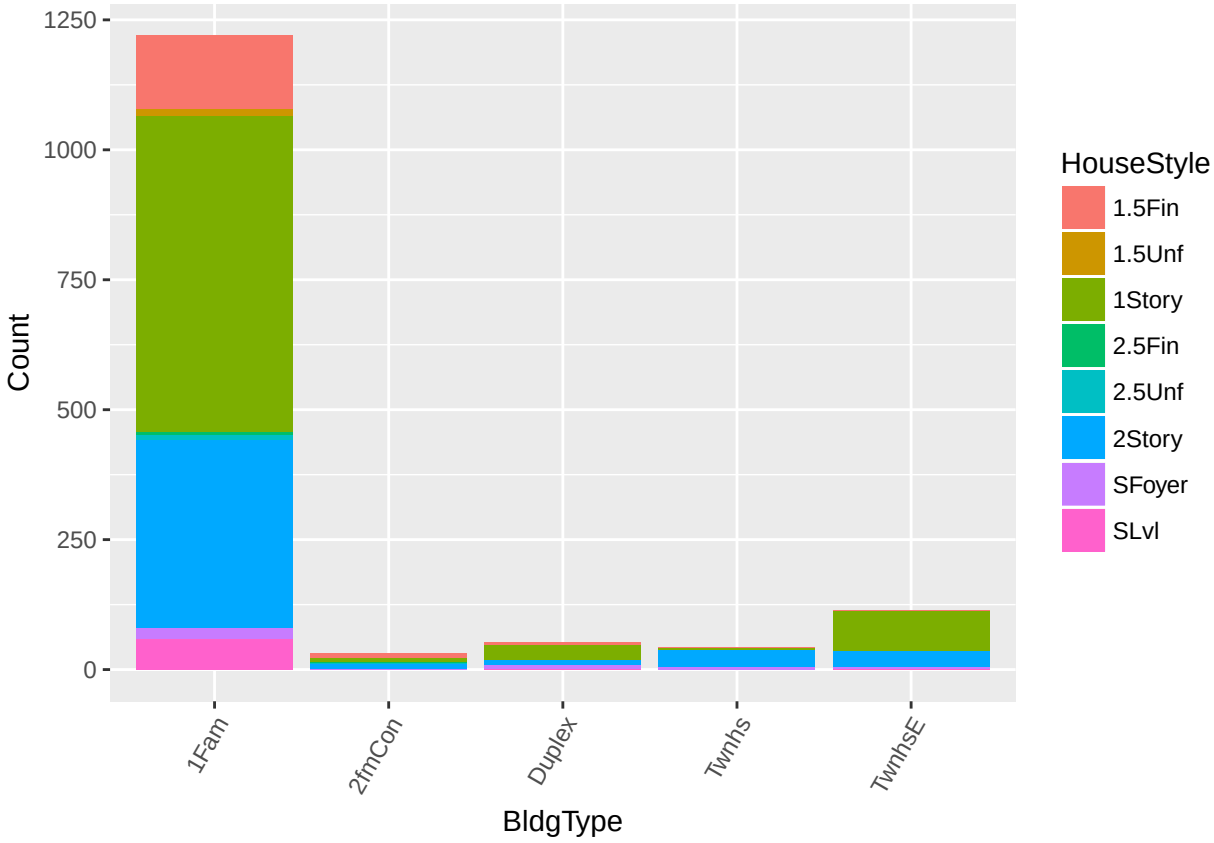
1.12 Relationship between BldgType and MSSubClass



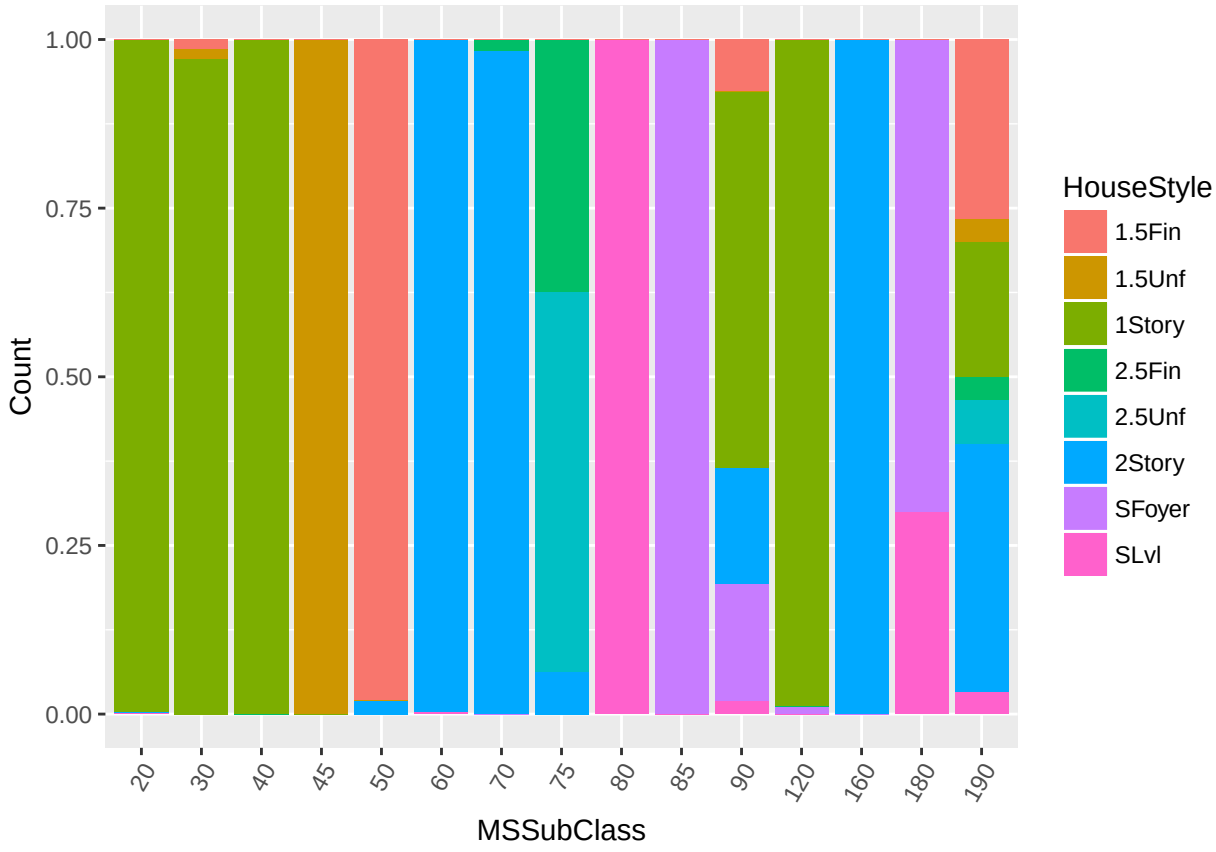
1.13 Relationship between LotShape and LandContour



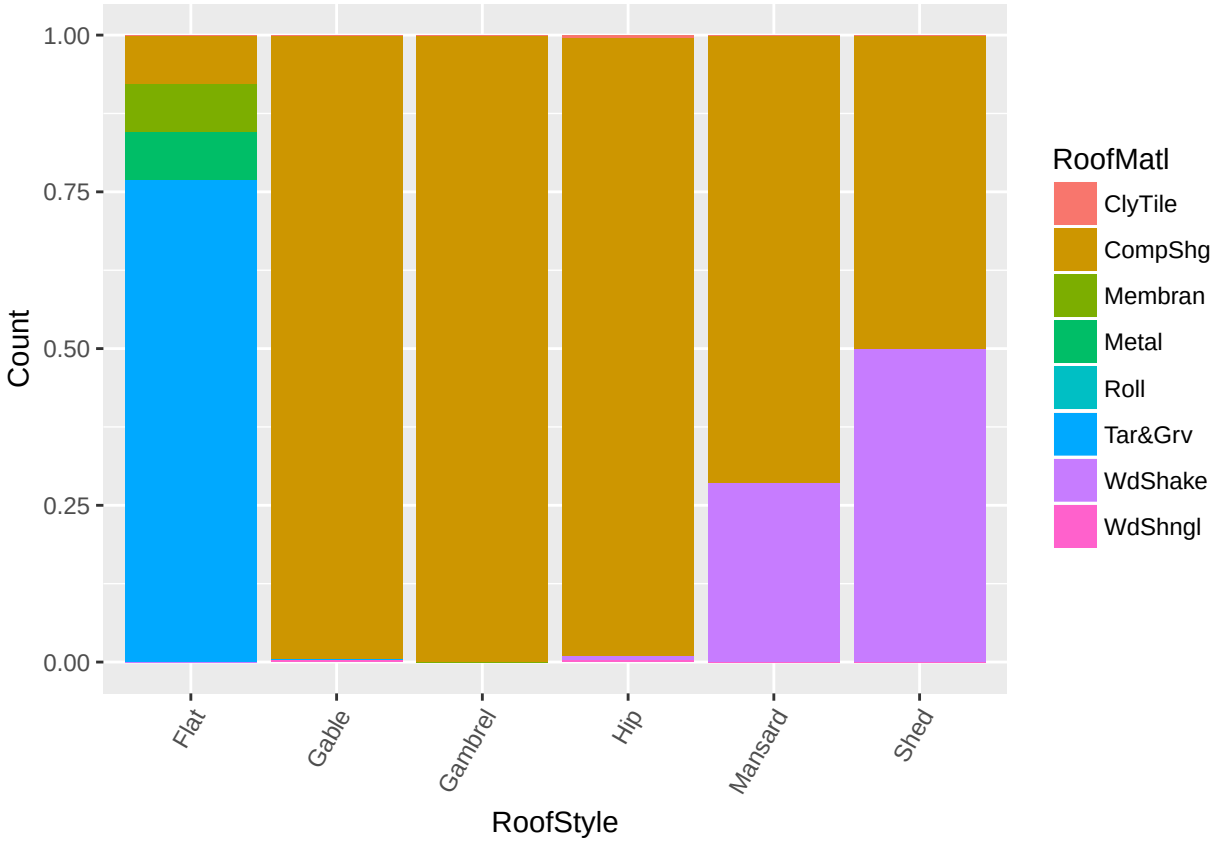
1.14 Relationship between BldgType and HouseStyle



1.15 Relationship between MSSubClass and HouseStyle

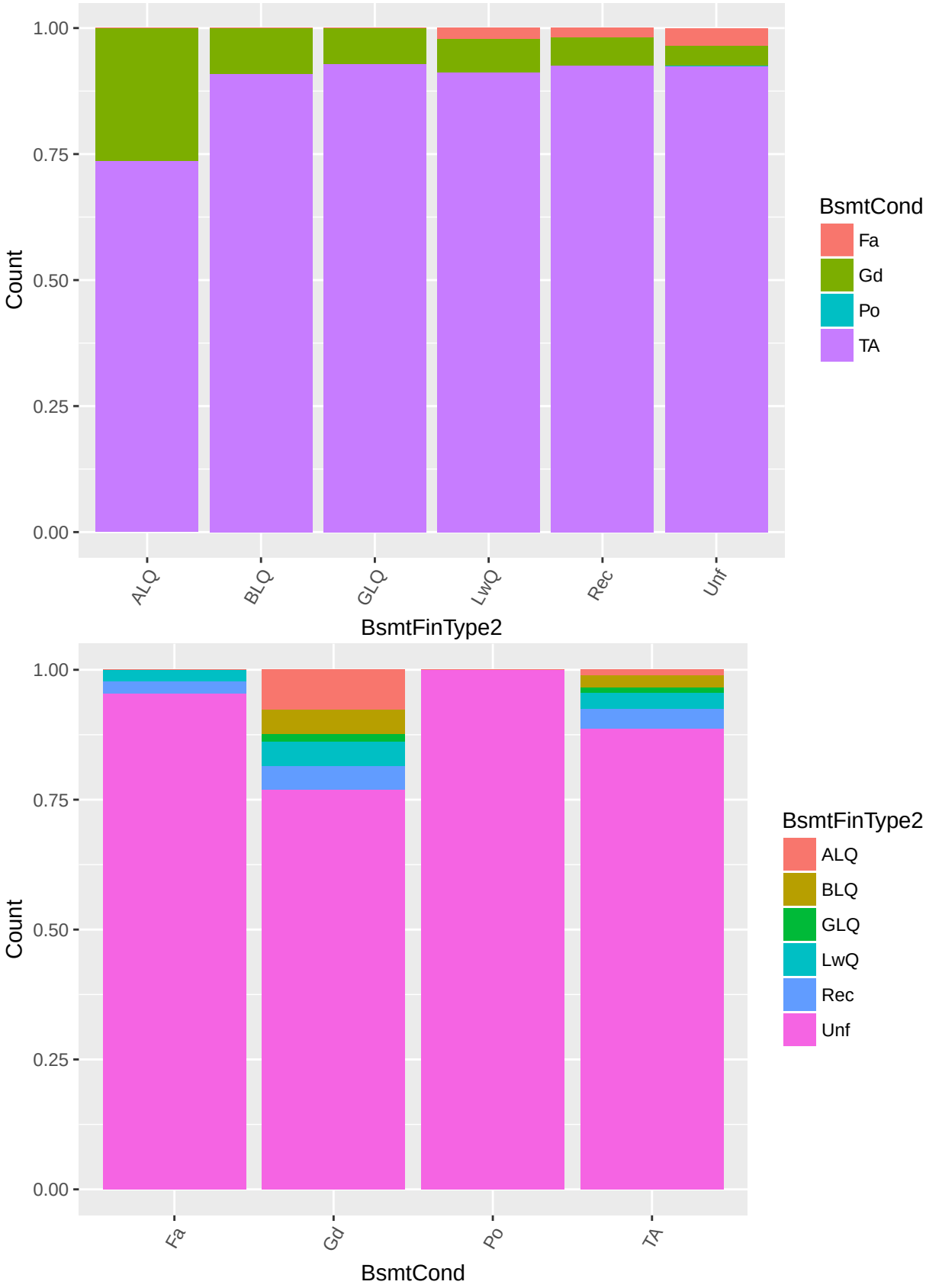


1.16 Relationship between RoofStyle and RoofMaterial



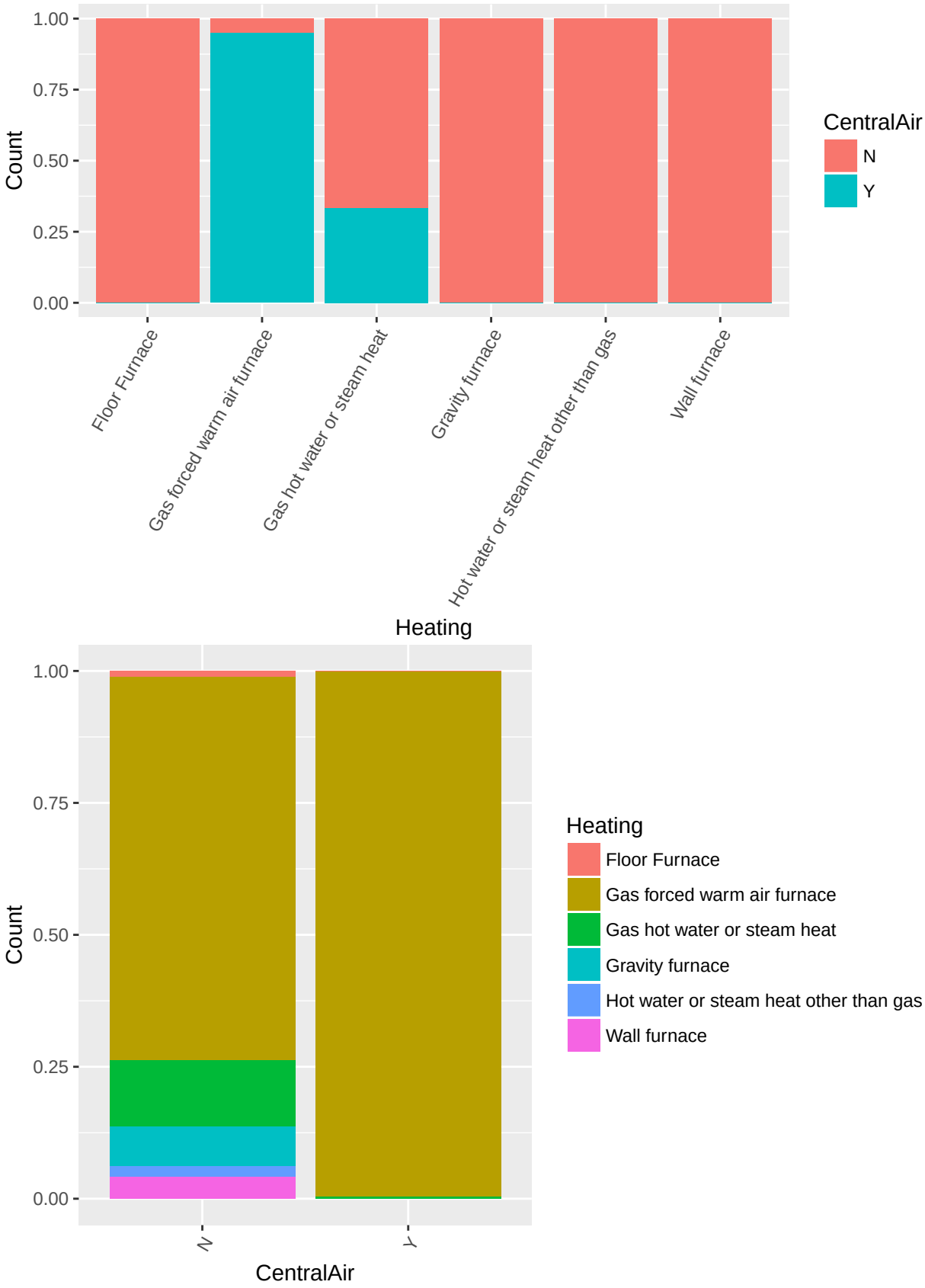
Looks like CompShg dominates the RoofMaterial area.

1.17 Relationship between BsmtFinType2 and BsmtCond



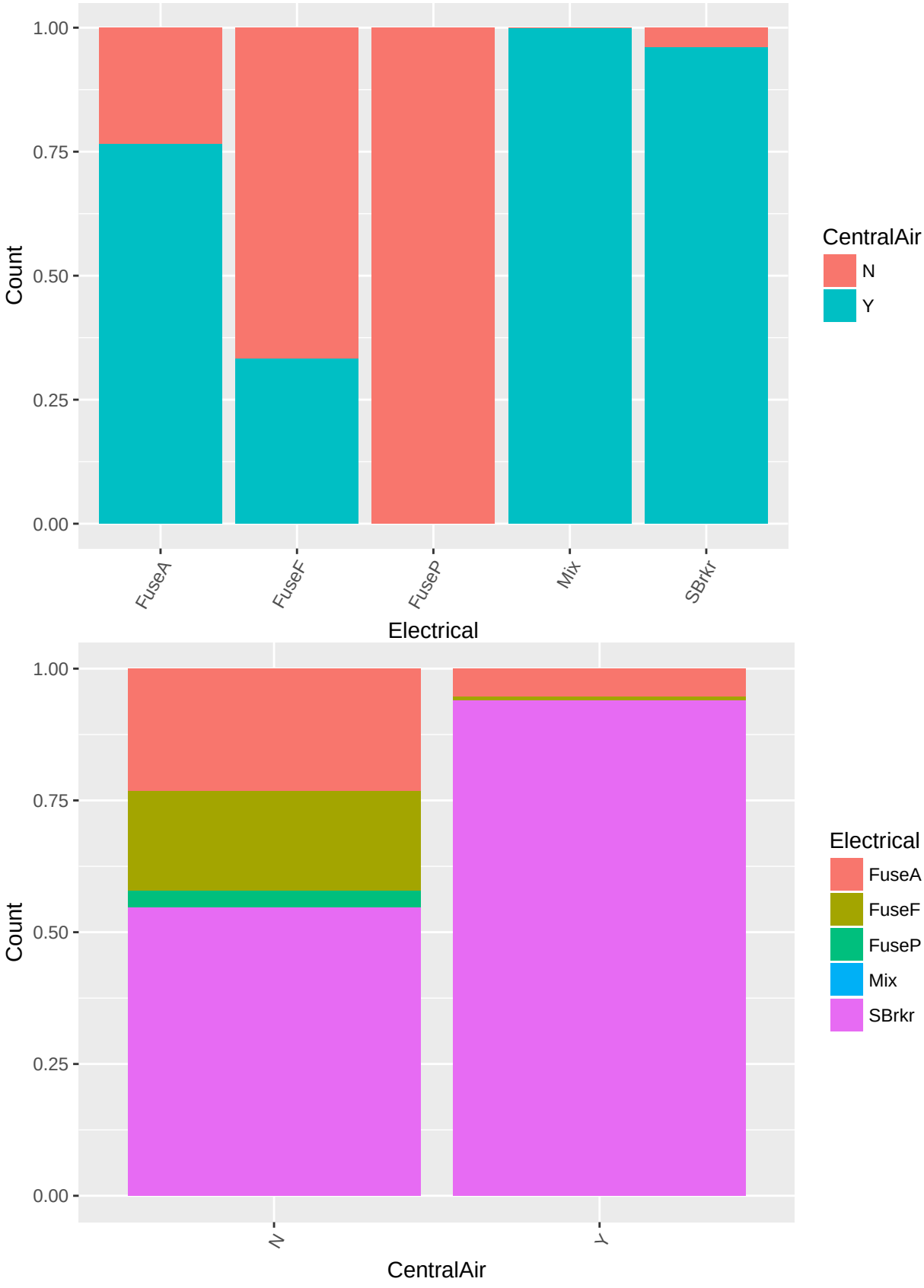
BsmtCond dominated by TA(typical/average)
BsmtFinType2 dominated by Unf (Uniform)

1.18 Relationship between Heating and CentralAir



There seems to be some kind of relationship. However, having central airconditioning would tend to increase the SalePrice. We can take a look.

1.19 Relationship between Electrical and CentralAir



Seems like there is a relationship!

1.20 Checking the correlation between areas of floors

```
[1] "Correlation between First Floor square feet and Second Floor square feet : -0.202646181002321"
[1] "Correlation between First Floor square feet and Low quality finished square feet (all floors) : -0
[1] "Correlation between Second Floor square feet and Low quality finished square feet (all floors) : 0
[1] "Correlation between First Floor square feet + Second Floor Square Feet and Above grade (ground) li
```

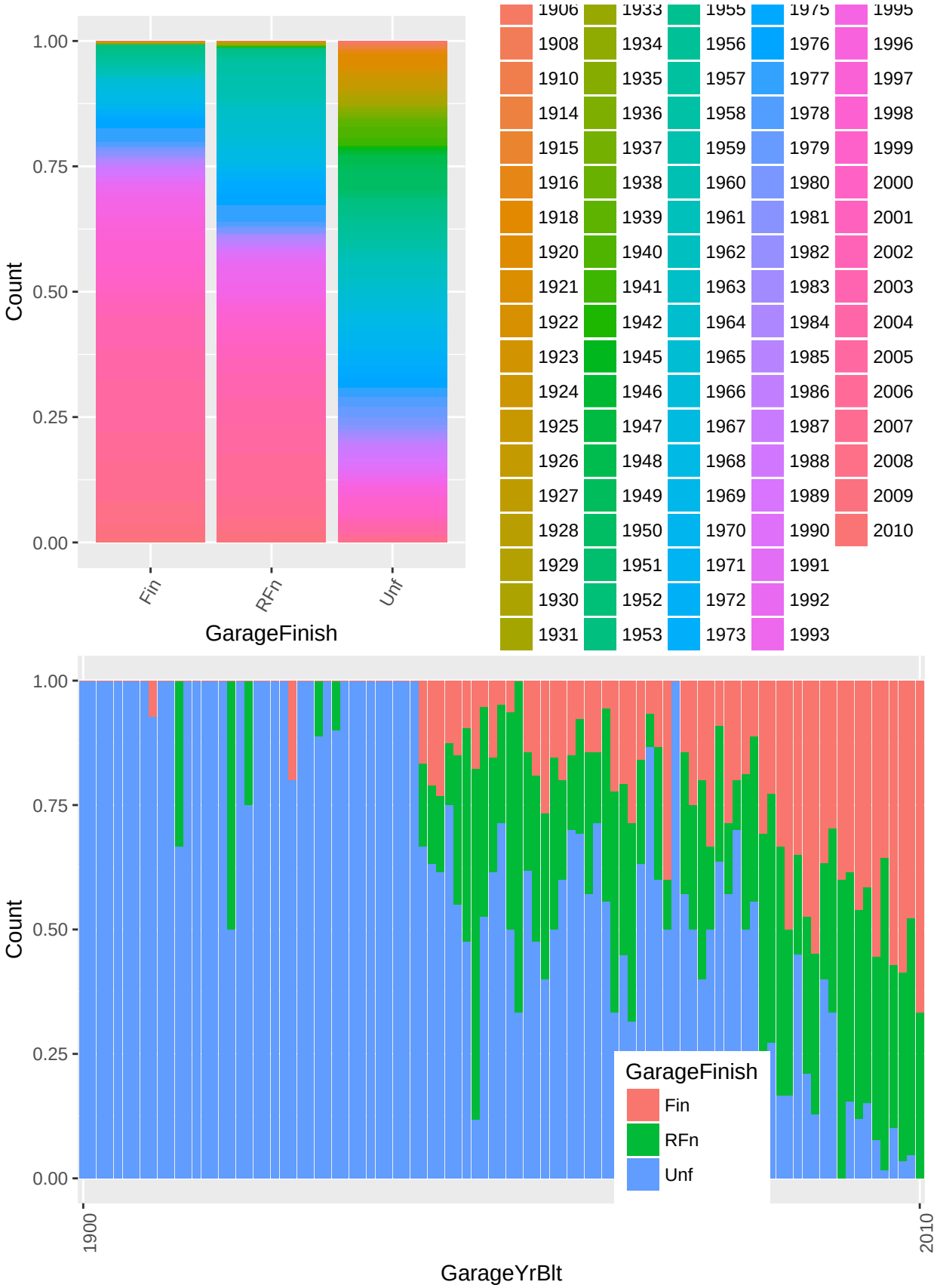
Wow! 0.99 correlation. Can remove First Floor square feet + Second Floor Square Feet and keep only Above grade (ground) living area square feet.

1.21 Checking the correlation between bedroom, bathroom, kitchen and total areas

```
## [1] "Basement full bathrooms and Bedrooms above grade : -0.150672809207956"
## [1] "Basement half bathrooms and Bedrooms above grade : 0.226651484150945"
## [1] "Basement half + full bathrooms and Bedrooms above grade : 0.0503177291776037"
## [1] "Total rooms above grade and Bedrooms above grade : 0.676619935742649"
## [1] "Total rooms above grade and Bedrooms +Kitchen above grade : 0.686475632178635"
```

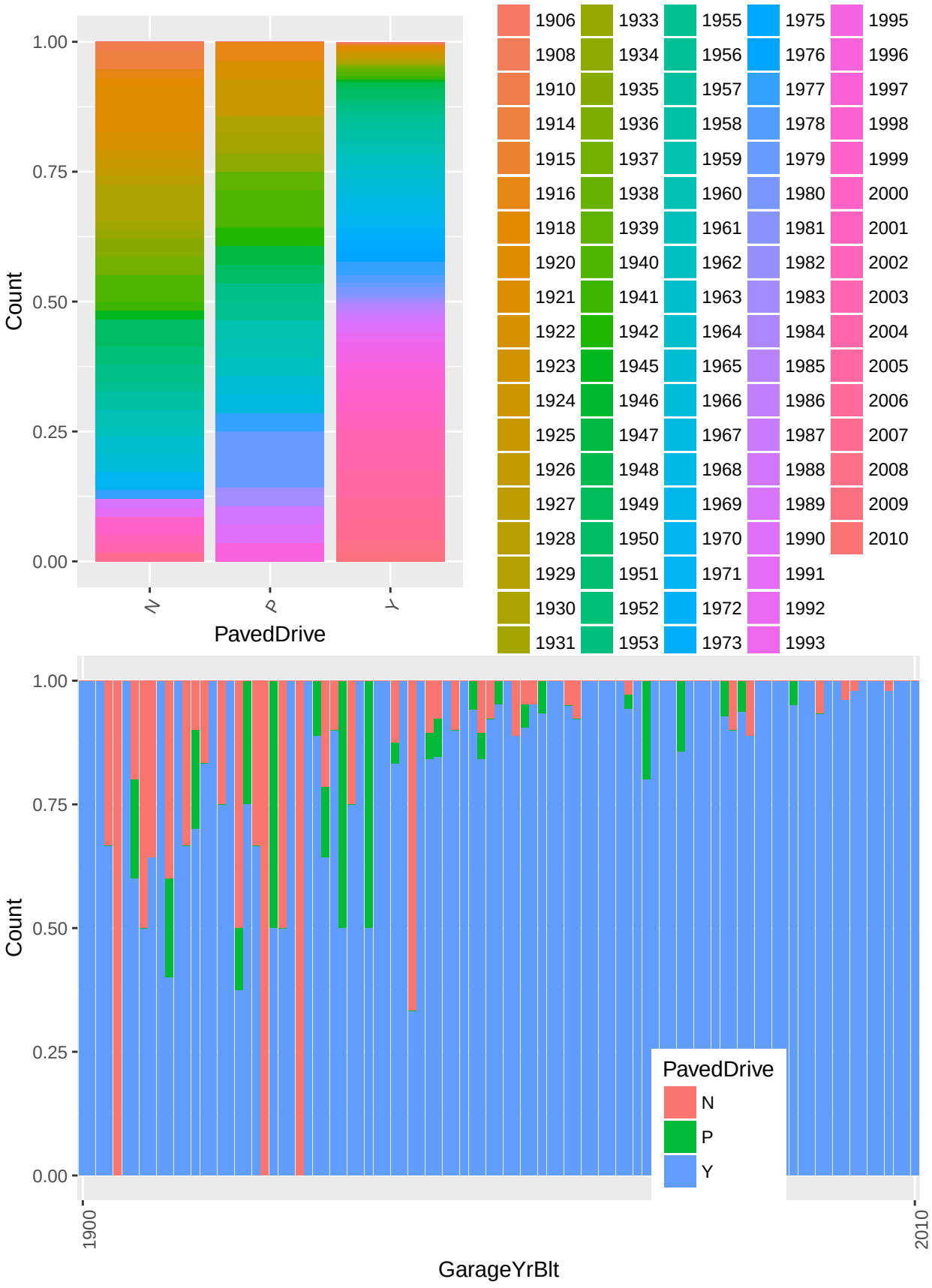
Good correlation between total rooms above grade and Bedrooms + Kitchen above grade

1.22 Relationship between when the garage was built and the finishing type



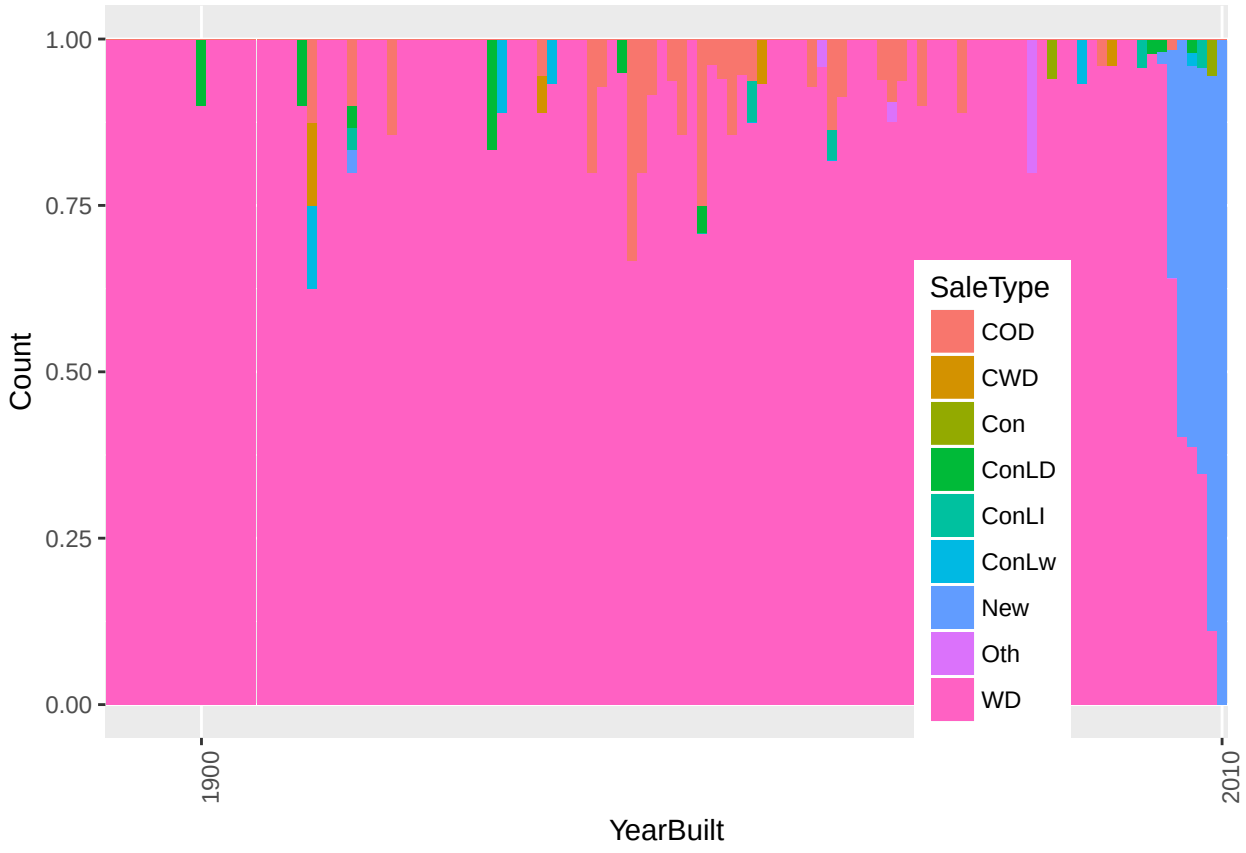
Newer garages having finished interiors vs the old ones that have unfinished

1.23 Relationship between when the garage was built and the driveway paving

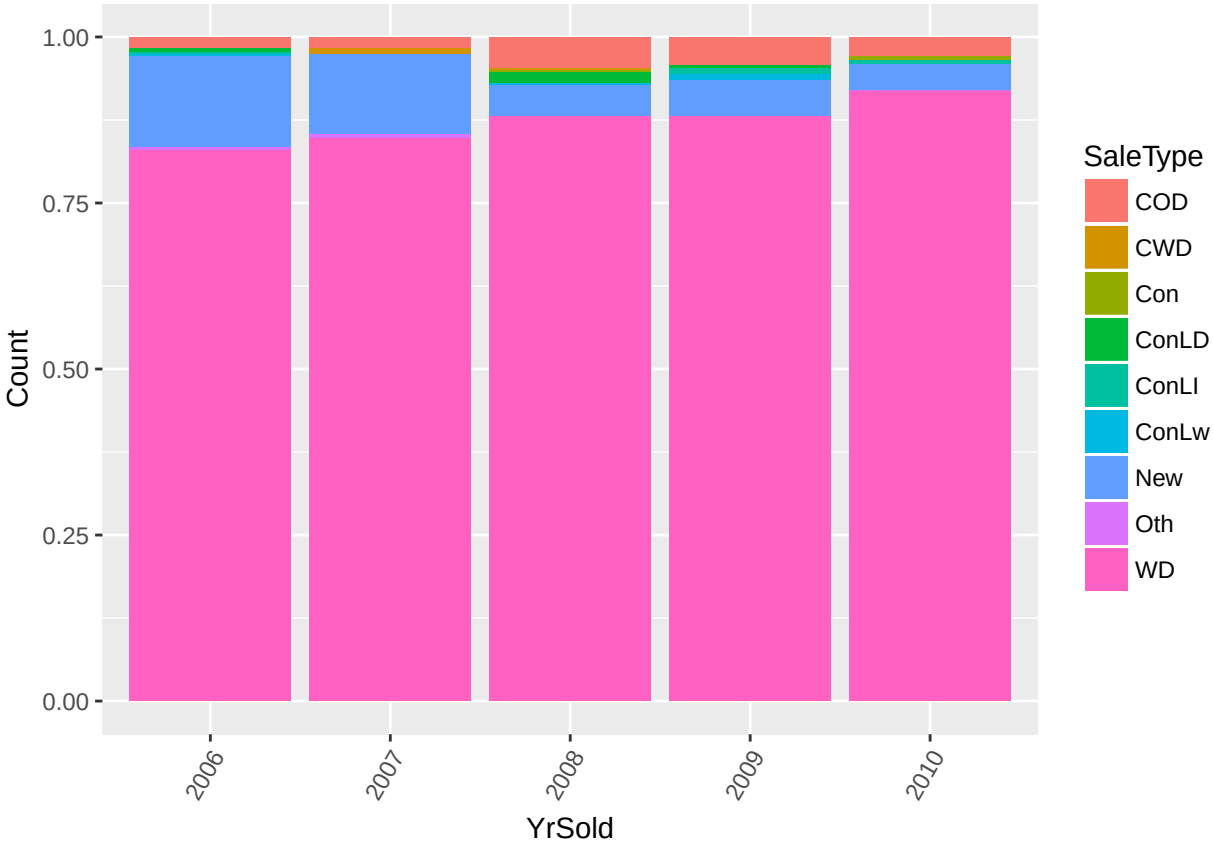


Newer garages having paved driveway vs the old ones.

1.24 Relation between Year built and sale type



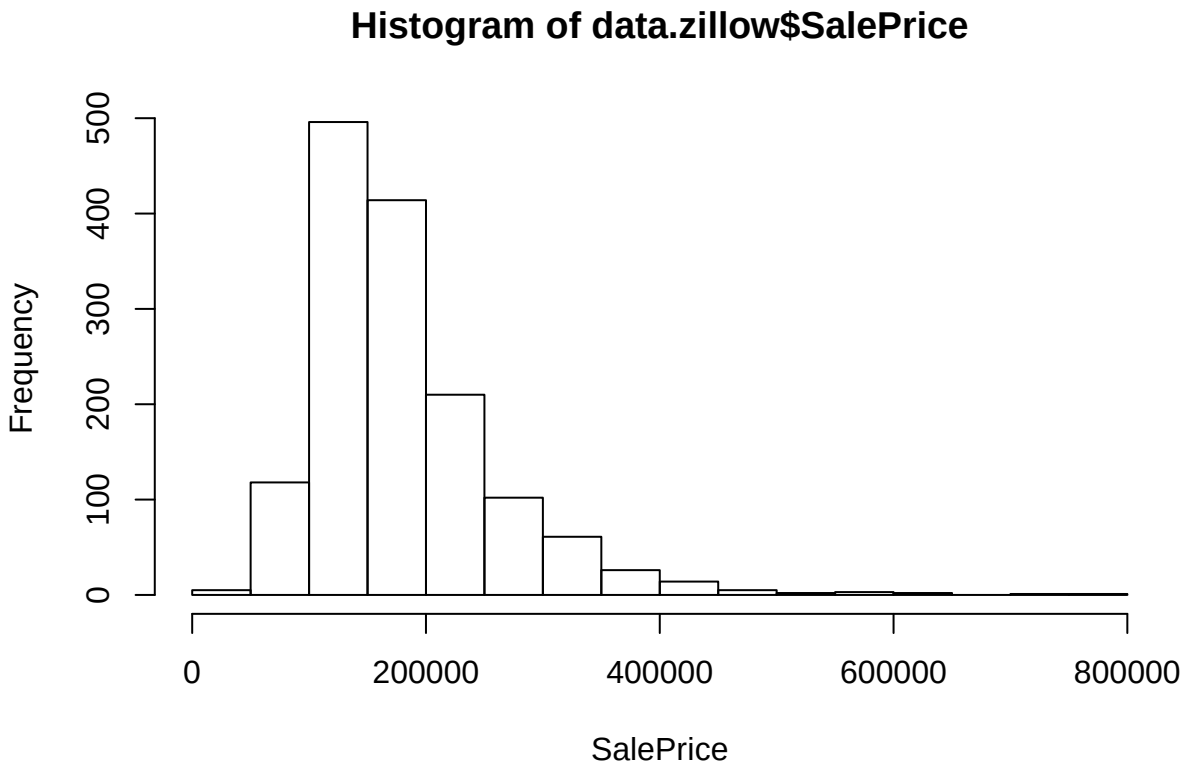
1.25 Relation between Year sold and sale type



Part 2 : Data cleaning and transformations

2.1 Treatment of outliers

2.1.1 Remove outlier from response and divide into train and test for imputation



2.1.2 Remove outliers in predictors

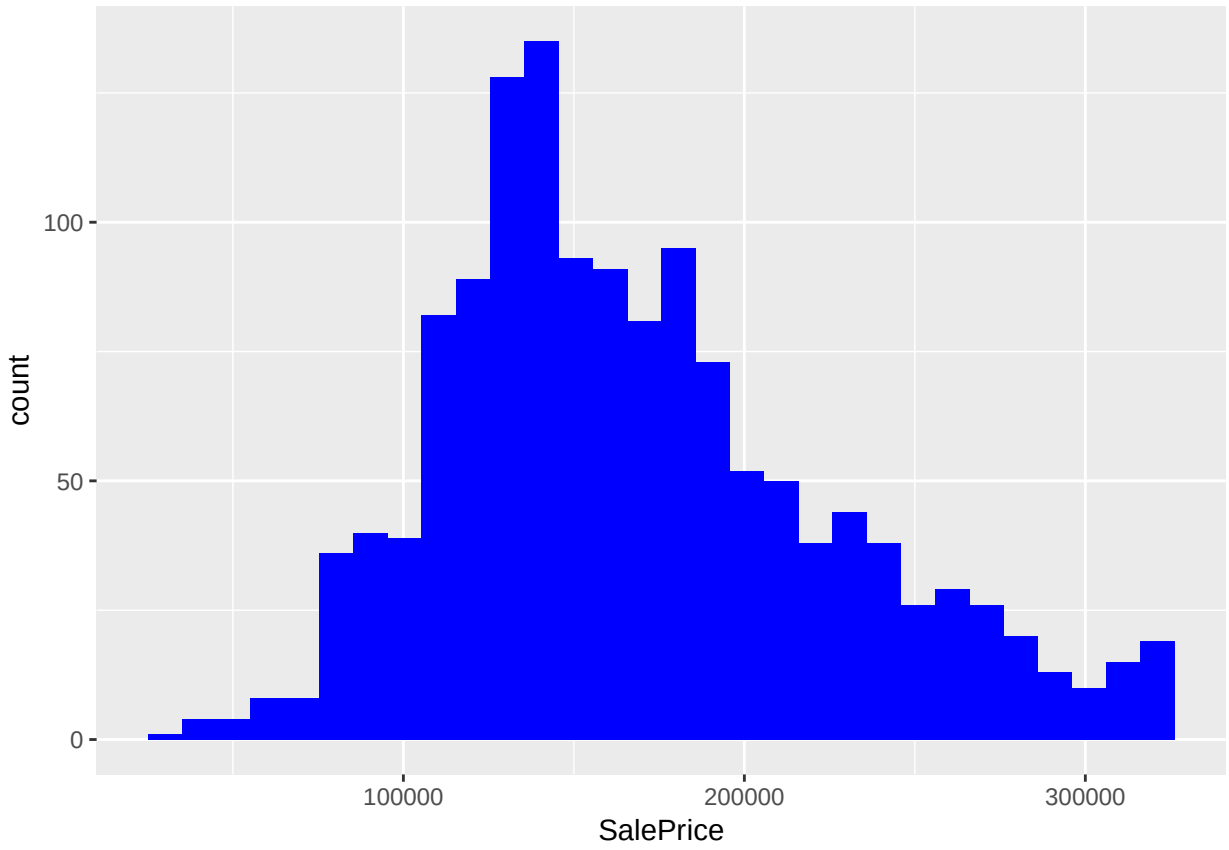
We need to treat all the outliers by squishing the data into a range, such that all data points lie between 5% and 95% of the data.

2.2 Data imputation and transformation

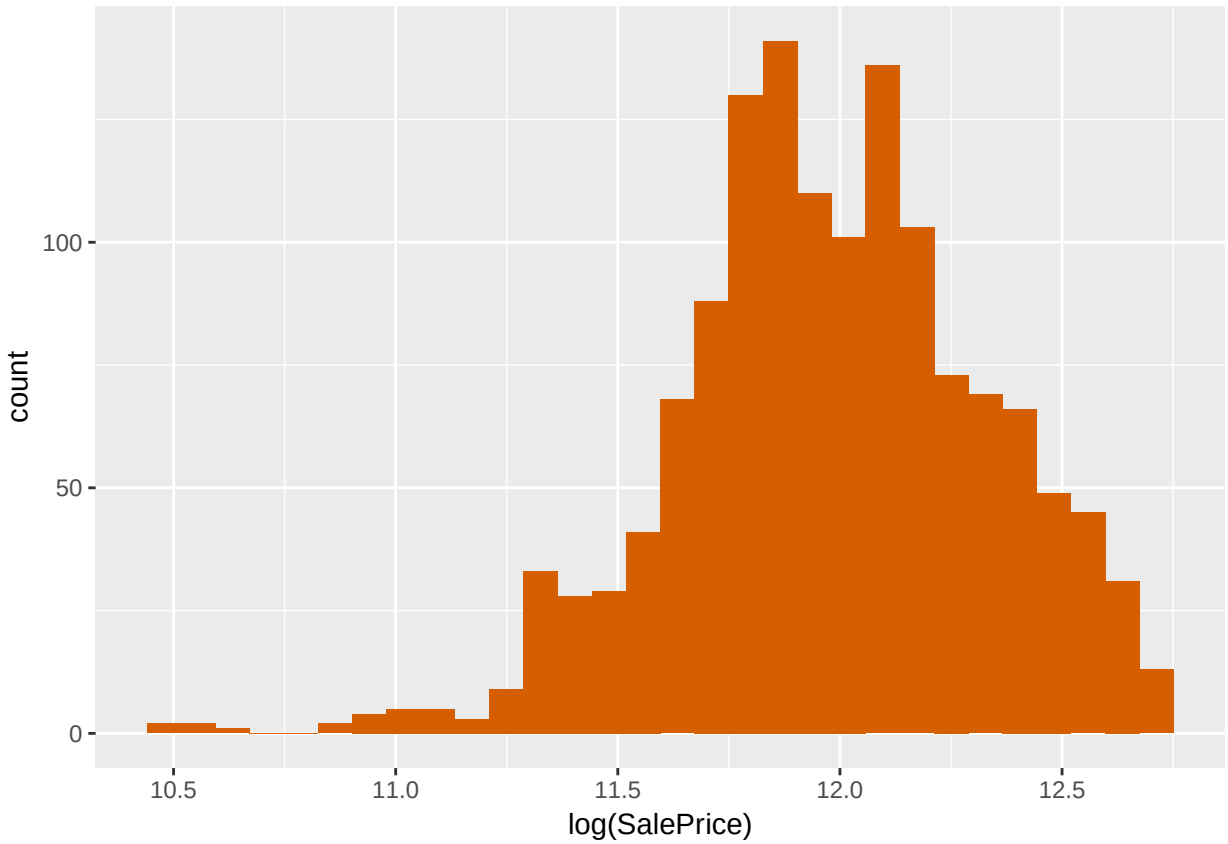
2.2.1 Transformations on SalePrice (the response variable)

For linear regression, we are making the assumption that the variables are normally distributed. We need to check for this assumption. If there are any outliers (example skewness), we need to get that fixed before we proceed.

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



From the boxcox plot, we see that we need to perform transformation, in order to make our model linear. The optimal lambda value we obtain is 0.3434343. So we apply the transformation of response_var to the power of lambda

2.2.2 Divide data into test and training set, this will be used in imputations later.

2.2.3 Creating a variable called Age, which calculates the age of the building

Also, bucketing the age into bins to identify the age category of houses.

2.2.4 Transforming numerical variables

There is a possibility that other features might be skewed as well. So let us try to identify the skewness and do a log transformation for them as well.

Now, Imputation time!

2.2.5 Fix missing categorical data for column Electrical

2.2.6 Fix LotFrontage

2.2.7 Using mice package to impute the rest of the variables. We use the cart (Classification and Regression Trees) to predict the missing values.

```
##
##  iter imp variable
##   1   1  MasVnrArea  GarageYrBlt
##   1   2  MasVnrArea  GarageYrBlt
##   1   3  MasVnrArea  GarageYrBlt
##   1   4  MasVnrArea  GarageYrBlt
##   1   5  MasVnrArea  GarageYrBlt
##   2   1  MasVnrArea  GarageYrBlt
##   2   2  MasVnrArea  GarageYrBlt
##   2   3  MasVnrArea  GarageYrBlt
##   2   4  MasVnrArea  GarageYrBlt
##   2   5  MasVnrArea  GarageYrBlt
##   3   1  MasVnrArea  GarageYrBlt
##   3   2  MasVnrArea  GarageYrBlt
##   3   3  MasVnrArea  GarageYrBlt
##   3   4  MasVnrArea  GarageYrBlt
##   3   5  MasVnrArea  GarageYrBlt
##   4   1  MasVnrArea  GarageYrBlt
##   4   2  MasVnrArea  GarageYrBlt
##   4   3  MasVnrArea  GarageYrBlt
##   4   4  MasVnrArea  GarageYrBlt
##   4   5  MasVnrArea  GarageYrBlt
##   5   1  MasVnrArea  GarageYrBlt
##   5   2  MasVnrArea  GarageYrBlt
##   5   3  MasVnrArea  GarageYrBlt
##   5   4  MasVnrArea  GarageYrBlt
##   5   5  MasVnrArea  GarageYrBlt
```

2.2.8 Transform catagorical variables into numerical by treating each catagory as a new variable.

2.2.9 Changing any missing integer value to mean

2.2.10 Making data.zillow as a mix of imputed integer and catagorical values

2.3 Test and training set creation

This will be used to train the ridge, lasso and elastic net models.

Part 3 : Explanatory Modeling

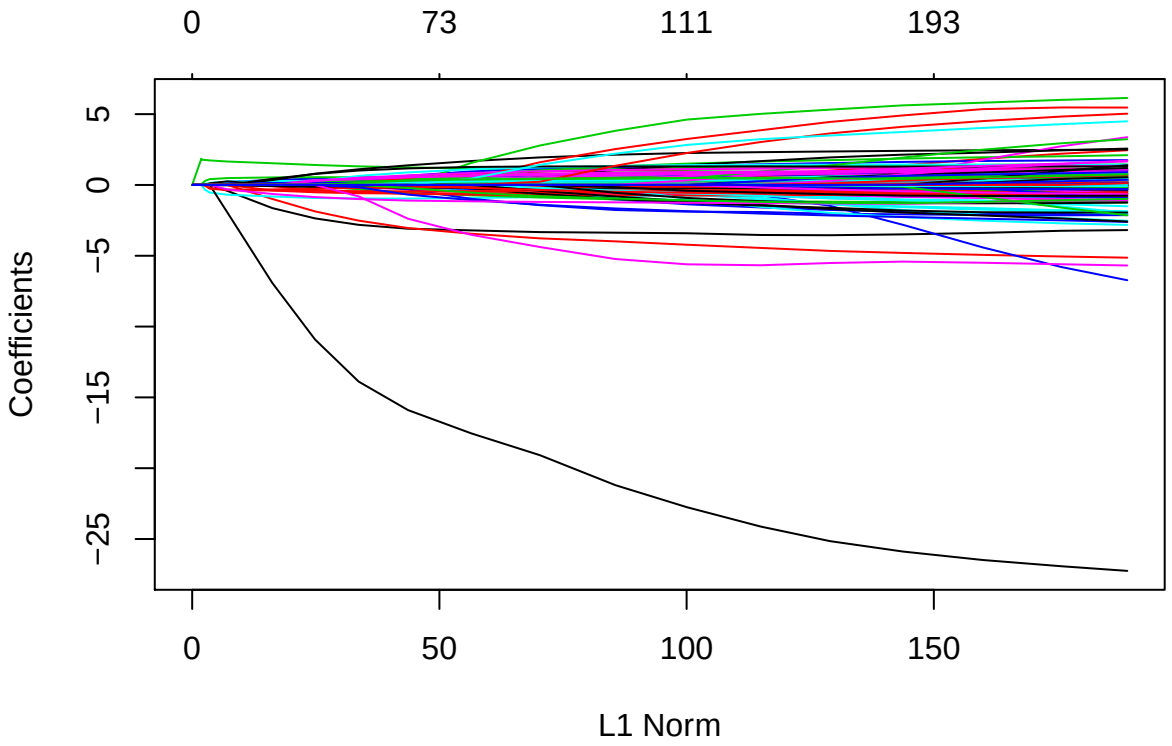
3.1 Variable selection

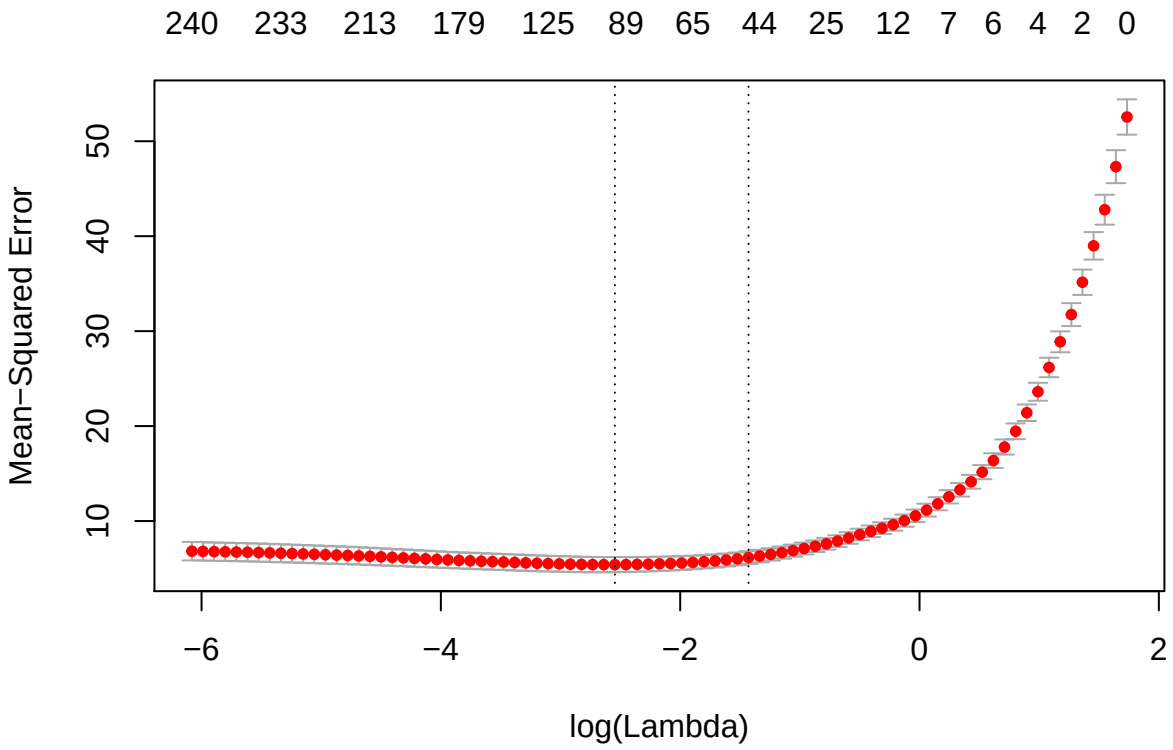
Now that we have the clean data, we see that the data has a total 290 variables. It is essential for an explanatory model to select variables over which we can run our least square regression model. For the variable selection, we are looking at the following steps.
(a) Perform Lasso regression on the training data to select an optimal value for lambda, which gives us the least MSPE.
(b) Using this lambda, we run Lasso on the full zillow dataset, to find the optimal variables for our model.
(c) From the variales that we get in lasso, we run the ols model on the full zillow dataset, and look at the

statistically significant values. In order to do this, we remove the variables that have P-value > 0.05 and then sort the variables in decending order by the absolute values of their co-efficients.
(d) Upon getting the significant variables, we process these variables using a forward subset selection process, where we select a total of 30 variables.
(f) These are the top 30 variables that we use for our explanatory modeling.

Going through the process step by step
Step(a) Lasso on the training set, ploting the model and the cross validation model

Since we have a total of 288 predictors, including dummy variables for the categorical variables, we opted to run a Lasso regression to identify those which have greater explanatory power and shrink the coefficients of other variables to zero. Firstly, we splited our original dataset in 80:20 ratio for training and testing respectively. The optimal tuning parameter for the Lasso regression was then selected using cross-validation that is executed by the function `cv.glmnet`.





Step(b) Run the Lasso on the full model, for the best lambda selected

```
## [1] "Variables selected using lasso"
```

```
##  [1] "LotArea"              "OverallQual"          "OverallCond"
##  [4] "YearRemodAdd"         "BsmtFinSF1"           "TotalBsmtSF"
##  [7] "X2ndFlrSF"            "GrLivArea"            "BsmtFullBath"
## [10] "FullBath"             "HalfBath"             "KitchenAbvGr"
## [13] "TotRmsAbvGrd"         "Fireplaces"           "GarageCars"
## [16] "GarageArea"           "WoodDeckSF"           "OpenPorchSF"
## [19] "ScreenPorch"          "Age"                  "MSZoningC (all)"
## [22] "MSZoningRM"           "AlleyPave"            "LotConfigCulDSac"
## [25] "LotConfigFR2"         "LandSlopeMod"         "NeighborhoodBrkSide"
## [28] "NeighborhoodClearCr"  "NeighborhoodCrawfor"  "NeighborhoodEdwards"
## [31] "NeighborhoodMeadowV"  "NeighborhoodMitchel"  "NeighborhoodNWAmes"
## [34] "NeighborhoodNoRidge"  "NeighborhoodNridgHt"  "NeighborhoodOldTown"
## [37] "NeighborhoodSomerst"  "NeighborhoodStoneBr"  "Condition1Artery"
## [40] "Condition1Norm"       "Condition1PosN"       "Condition1RRAe"
## [43] "Condition2PosA"       "Condition2PosN"       "BldgType1Fam"
## [46] "BldgTypeDuplex"       "BldgTypeTwnhs"        "RoofStyleMansard"
## [49] "RoofMatlClyTile"      "RoofMatlMembran"      "RoofMatlWdShngl"
## [52] "Exterior1stBrkComm"   "Exterior1stBrkFace"   "Exterior1stHdBoard"
## [55] "Exterior1stWd Sdng"   "MasVnrTypeStone"      "ExterQualTA"
## [58] "ExterCondFa"          "ExterCondTA"          "FoundationPConc"
## [61] "FoundationStone"      "FoundationWood"       "BsmtQualEx"
## [64] "BsmtQualTA"           "BsmtCondFa"           "BsmtExposureGd"
## [67] "BsmtFinType1GLQ"      "BsmtFinType1Unf"      "BsmtFinType2ALQ"
## [70] "HeatingGasW"          "HeatingGrav"          "HeatingOthW"
## [73] "HeatingQCEx"          "HeatingQCTA"          "CentralAirN"
## [76] "KitchenQualEx"        "KitchenQualTA"        "FunctionalMaj2"
## [79] "FunctionalMod"        "FunctionalSev"        "FunctionalTyp"
```

```
## [82] "FireplaceQuGd"        "GarageType2Types"     "GarageTypeBuiltIn"
## [85] "GarageFinishFin"      "GarageFinishUnf"      "GarageQualFa"
## [88] "GarageQualGd"         "GarageCondFa"         "PavedDriveY"
## [91] "PoolQCGd"             "SaleTypeCon"          "SaleTypeConLD"
## [94] "SaleTypeNew"          "SaleConditionAbnorml" "SaleConditionFamily"
```

we are left with only 96 variables out of 288 variables. But for explanatory purpose we wanted to reduce the number of variables, which would make the model simpler having better explanatory power. Hence, on these 96 variables we performed OLS to generate most significant variables. Here, our approach was to select all those variables which were significant at 5% level of significance.

Step(c) Run OLS on the selected variables to see which variables are statistically significant

```
## [1] "Variables in the decreasing order of their statistical significance"
```

```
##  [1] "RoofMatlClyTile"      "Condition2PosN"       "FunctionalSev"
##  [4] "Condition2PosA"       "MSZoningC (all"       "RoofMatlMembran"
##  [7] "SaleTypeCon"          "RoofMatlWdShngl"      "NeighborhoodStoneBr"
## [10] "NeighborhoodMeadowV"  "SaleTypeConLD"        "HeatingOthW"
## [13] "HeatingGrav"          "NeighborhoodNridgHt"  "RoofStyleMansard"
## [16] "KitchenQualEx"        "NeighborhoodNoRidge"  "FunctionalMod"
## [19] "FunctionalTyp"        "SaleConditionAbnorml" "SaleConditionFamily"
## [22] "Condition1PosN"       "OverallQual"          "NeighborhoodEdwards"
## [25] "NeighborhoodMitchel"  "OverallCond"          "MSZoningRM"
## [28] "Condition1Artery"     "SaleTypeNew"          "AlleyPave"
## [31] "LotConfigFR2"         "NeighborhoodNWAmes"   "BsmtFullBath"
## [34] "GarageCars"           "Exterior1stWd Sdng"   "NeighborhoodClearCr"
## [37] "BldgType1Fam"         "FullBath"             "BsmtFinType1GLQ"
## [40] "MasVnrTypeStone"      "NeighborhoodOldTown"  "Exterior1stHdBoard"
## [43] "HalfBath"             "GarageTypeBuiltIn"    "BsmtFinType1Unf"
## [46] "BsmtQualTA"           "HeatingQCTA"          "TotRmsAbvGrd"
## [49] "WoodDeckSF"           "KitchenQualTA"        "OpenPorchSF"
## [52] "Age"                  "FireplaceQuGd"        "YearRemodAdd"
## [55] "GrLivArea"            "TotalBsmtSF"          "GarageArea"
## [58] "BsmtFinSF1"           "X2ndFlrSF"
```

Step(d) From the variables that we selected above, we now run the forward subset selection, to get our top 30 desired variables

```
## [1] "Top 30 variables from forward selection method :"
```

```
##  [1] "RoofMatlClyTile"      "Condition2PosN"       "FunctionalSev"
##  [4] "MSZoningC (all)"      "SaleTypeCon"          "HeatingOthW"
##  [7] "KitchenQualEx"        "FunctionalTyp"        "SaleConditionAbnorml"
## [10] "OverallQual"          "NeighborhoodEdwards"  "OverallCond"
## [13] "MSZoningRM"           "Condition1Artery"     "SaleTypeNew"
## [16] "NeighborhoodNWAmes"   "BsmtFullBath"         "GarageCars"
## [19] "NeighborhoodClearCr"  "BldgType1Fam"         "BsmtQualTA"
## [22] "HeatingQCTA"          "WoodDeckSF"           "Age"
## [25] "FireplaceQuGd"        "YearRemodAdd"         "GrLivArea"
## [28] "TotalBsmtSF"          "GarageArea"           "BsmtFinSF1"
```

Finally, we modelled these top 30 variables using OLS that returns a set of unbiased coefficients of estimation.

3.2 Running the least square regression model on the selected 30 variables

Also perform the cross validation using *cv.lm(data.zillow.lasso.best.subset, fit.zillow, m=5)*

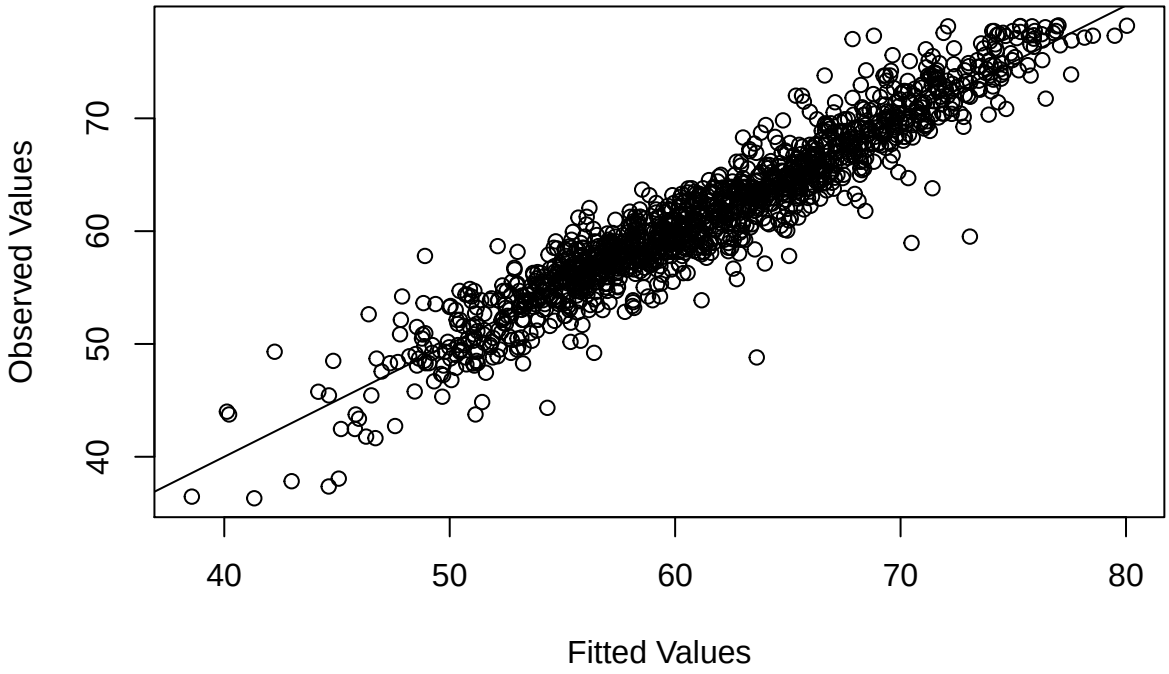
We obtain a good R^2 using the cross validation test.
Next step, is to check for model assumptions to see if they are met or not.

3.3 Checking model assumptions

Using the least square model, we check if the assumptions that we made are satisfied or not. The assumptions that we need to look for are : (a) Our model is linearly distributed around the true response values. (b) Error terms are statistically independent (c) Error terms have constant variance (d) Error terms are not correlated to each other. (e) Mean of the error terms is 0.

3.3.1 Linearity and additivity of the relationship between dependent and independent variables:

3.3.1.1 Plot of observed versus predicted values



The points is symmetrically distributed around a diagonal line, with a roughly constant variance.

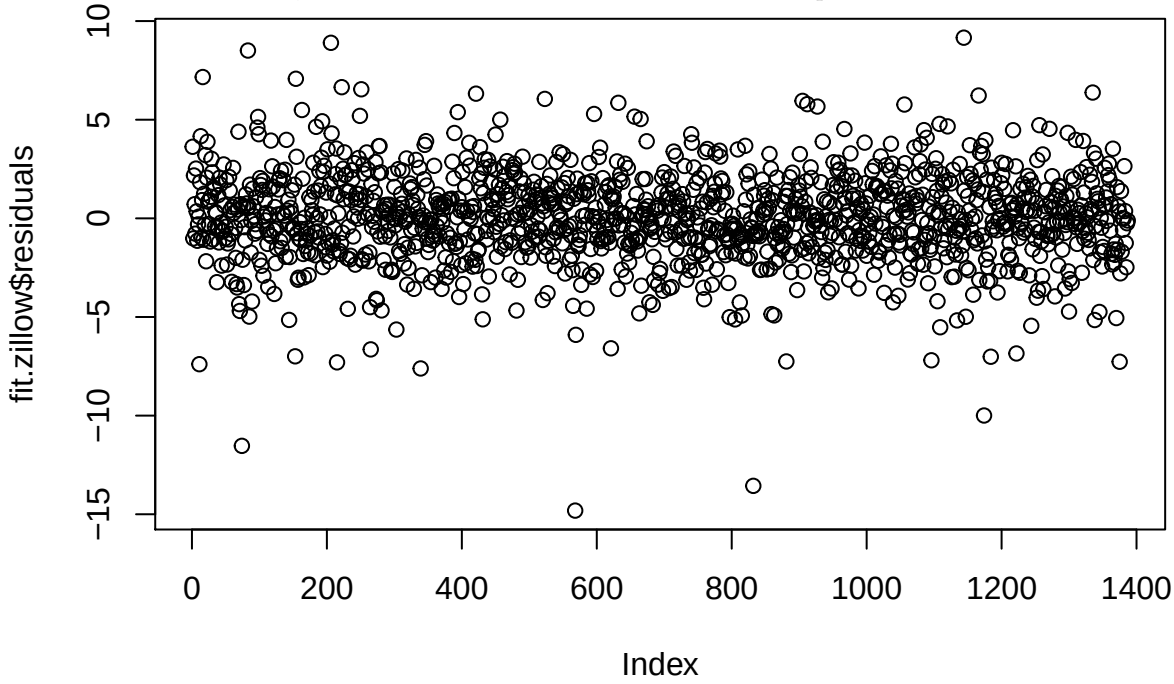
3.3.1.2 Plot of residuals versus predicted values

The points is symmetrically distributed around a horizontal line, with a roughly constant variance. This suggests homoscedasticity of the residuals, which can be interpreted as homoscedasticity of the error terms.

3.3.2 Statistical independence of the errors

3.3.2.1 plot the residual

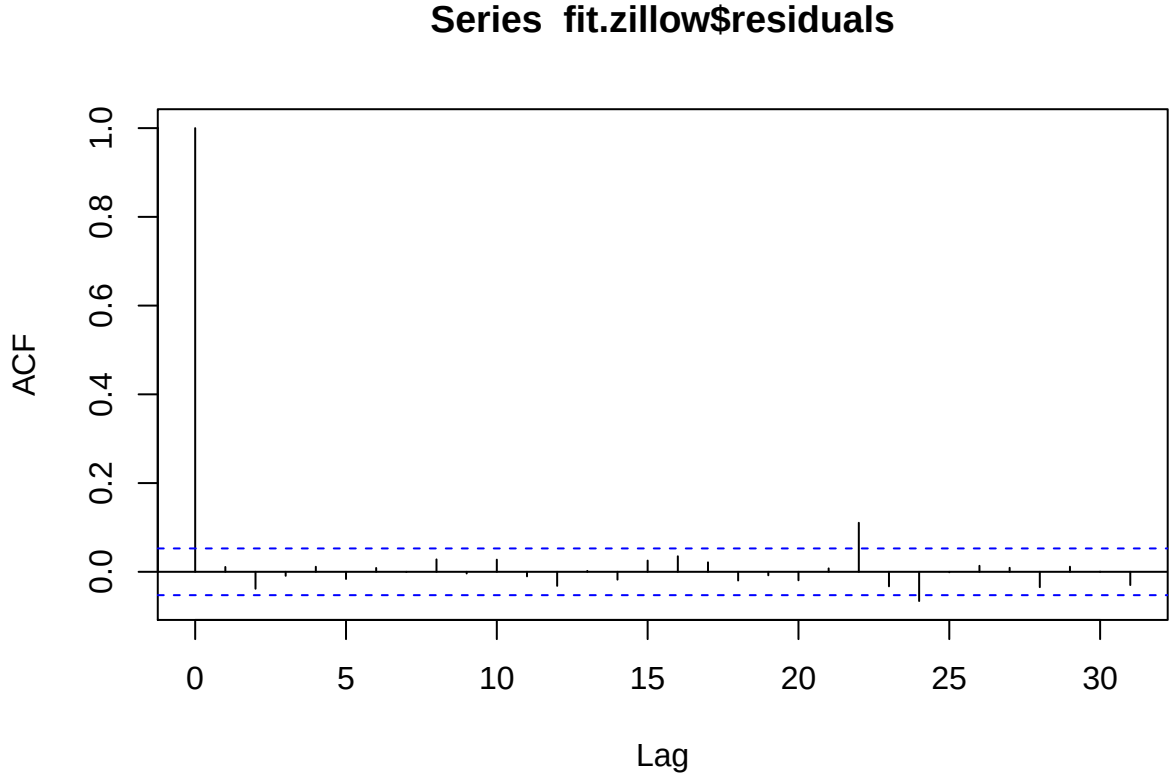
Since we are unaware of the error term, we look at the residual term instead. Let us plot the residual to check



how it looks.

3.3.2.2 correlation between the residuals

Now, I try to find the correlation between the residuals.
We use the auto-correlation function. If the residuals were not autocorrelated, the correlation (Y-axis) from the immediate next line onwards will drop to a near zero value below the dashed blue line (significance level).



The blue lines in the plot is our critical value line. We see that all autocorrelations are below our critical lines (except for the first one, which is the correlation of a variable with itself).
Since the acf drops below the significance level, we can say that the resdiuals are not auto-correlated.

3.3.2.3 Durbin-Watson statistic

In order to test the autocorrelation more formally, we look at the Durbin-Watson test. The Durbin-Watson statistic is always between 0 and 4. A value of 2 means that there is no autocorrelation in the sample. Values approaching 0 indicate positive autocorrelation and values toward 4 indicate negative autocorrelation.

The DW value of 2 suggests that there is no correlation between the residuals, which can be interpreted as no correlation between the error terms in the model. Thus our assumption is met.

3.3.3 Test for constant variance of the error term. (Homoscedasticity)

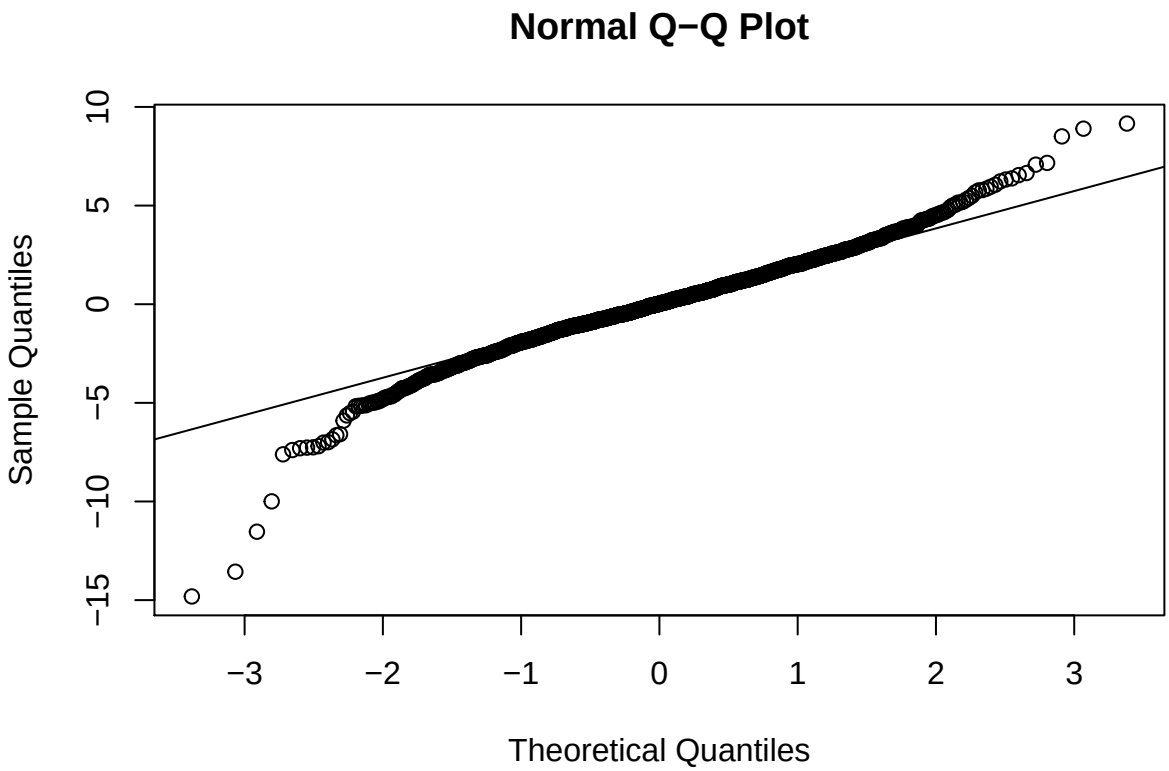
We can see that that residuals are well distributed around the fitted values with constant variance. We do not see any major outliers in the plot that could account for heteroscedasticity. Hence, we can say that the residuals and therefore the error terms are homoscedasticity in nature. Our assumptions are met.

3.3.4 Normality of the error distribution

3.3.4.1. Ploting the residuals

The residuals are approximately normally distributed around the mean of 0. This accounts for normal distribution of the error terms. Hence, our assumption is met.

3.3.4.2. Q-Q plot of residuals



A normal distribution here is indicated by the points lying close to the diagonal reference line, for the most part. Some deviations from the line suggests that we have skewness in the plot.
We tried several transformations with log(response variable), log(log(response variable)), log(response variable)^lambda (lambda from box-cox), log(response variable + 1)^lambda/lambda, but this was the closest to linearity that we could get with the data and the predictors we choose.

3.3.4.3 Kolmogorov-Smirnov test

3.3.4.4 Shapiro Wilk's W test

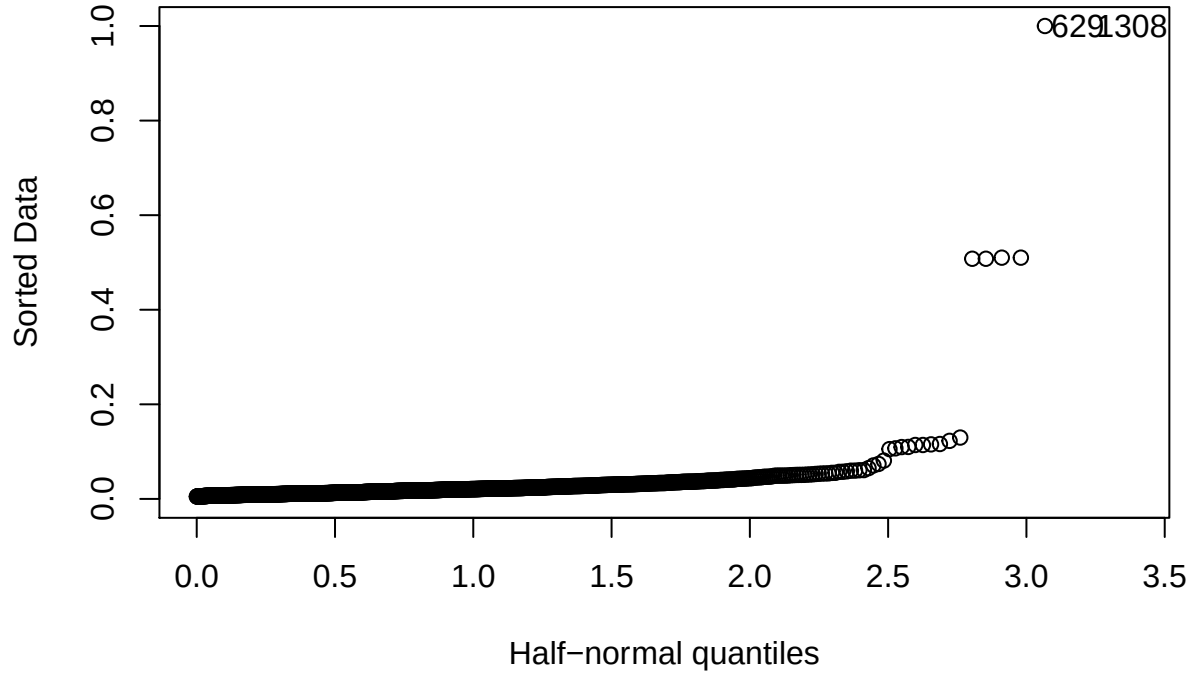
We see that the p-value is rather low for both the Kolmogorov-Smirnov test and the Shapiro-Wilk normality test. Our hypothesis is not met and we fail the "statistical normality test".
However, when we look at the histogram of the residuals and the Q-Q plot, we can see that the residuals and the error terms are approximately normally distributed. Failing the KS test and the Shapiro test could be attributed to the fact that we are dealing with the real world data, which is not always in a normal form. But since our plots suggests sufficient normality, we can assume error terms to be normal for our data.

Hence, our model meets all assumptions of linearity.

Next we look at the values to see if we have any unusual observations.

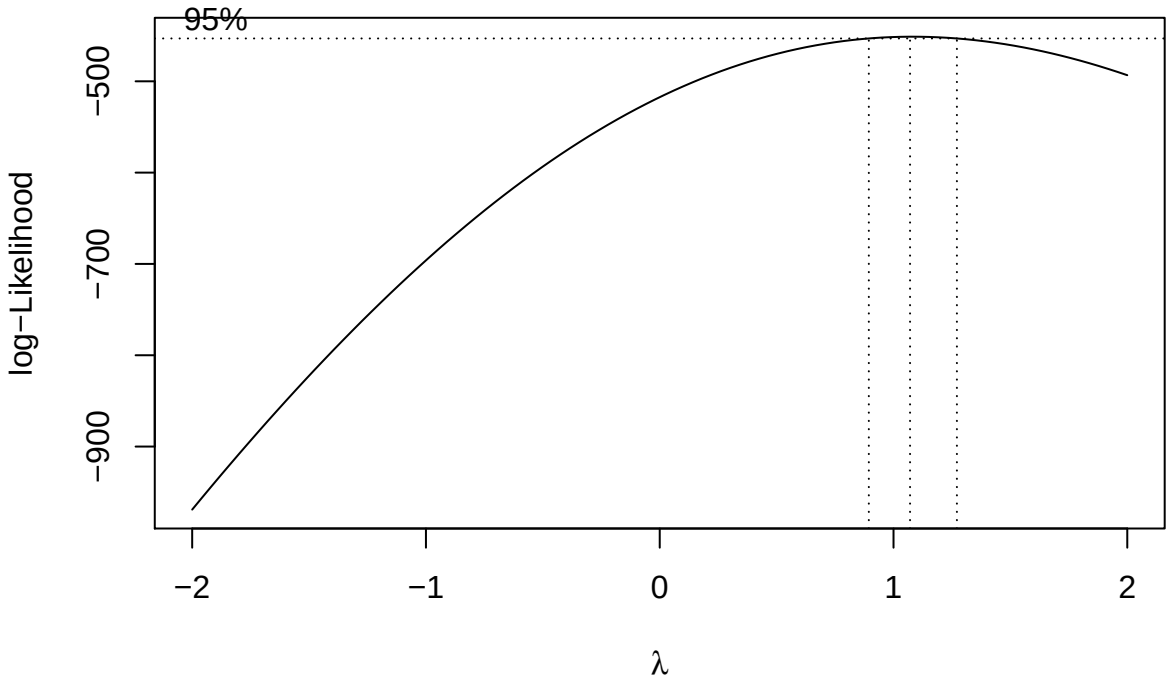
3.3.5 Finding unusual observations

3.3.5.1 Points of leverage



There are no particular points of high leverage that influences the fit.
Also, all continuous variables have already been treated for outliers at the start of the analysis.

3.3.6 Variance Stabilization



The new box-cox plot suggests that the model is variance stabilized.

3.3.7 Checking for Multicolinearity

The VIF diagnostics tell us that we do not have any multicolinearity between the variables that we have selected in our model.

Part 4 : Predicting values for Morty's house

Lets try to predict the price of Morty's house. For this, we need to clean and normalize his data first.

4.1 Loading Data

So Morty comes in with his house data. Lets try to predict the price of his house.
For this, we need to clean and normalize his data first.

4.2 Creating variables

4.3 Predicting Morty's house price

We run the linear regression model with the variables we deem fit and find the 95% confidence interval for the model. Looking at the values, we can say that we are 95% confident that if Morty sells his house now, as is, he would receieve a maximum of 160954 USD. On an average, he can sell his house for 155415 USD.

4.4 Suggesting top 3 changes that Morty can do to improve the sale price of his house

From the variables that we selected above, Morty can make changes to the following
1. Kitchen Quality : Morty can improve his kitchen quality from Typical to Excellent, which will impact his house's sale price positively.
2. Fireplaces : If Morty can add fireplaces to his property, the house price is likely to go up.
3. Basement Quality : Morty can improve his basement quality from Typical to Excellent, which will impact his house's sale price positively.

We suggest this, as these are the top variables (in terms of significance level) that can be dealt with.

We will change the above mentioned variables to see how improving some conditions increases the selling price.

If Morty changes all three variables, he can sell his house for a maximum price of 191359 USD and an average price of 182311 USD.

If Morty just changes the Kitchen Quality to make it excellent, he can sell his house for a maximum of 170960 USD and an average of 162860 USD.

If Morty just adds a fireplace to his property, he can sell his house for a maximum of 162095 USD and an average of 156315 USD.

If Morty just improves his basement quality to make it excellent, he can sell his house for a maximum of 171202 USD and an average of 164156 USD.