Predicting Housing Prices using Regression model

Aruna Harini Yandrapally

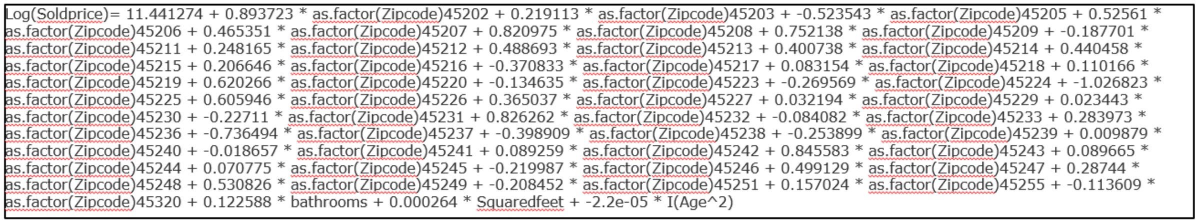
Date: 06th December 2019

The **Purpose of this project** is to first study and analyse what has been the trend in the selling price of houses in Cincinnati area by looking at the data collected from websites - Zillow (https://www.zillow.com/) and Trulia (https://www.trulia.com/). Next is creating a model based on selected features which could predict the selling price of houses that come up for sale in the future with high degree of accuracy.

Quick Summary of Final Model : We have built 2 models, one using a bigger dataset with less number of covariates, and other small dataset with additional variables, because we have sparse dataset. The plan is to average the two models to arrive at the final prediction.

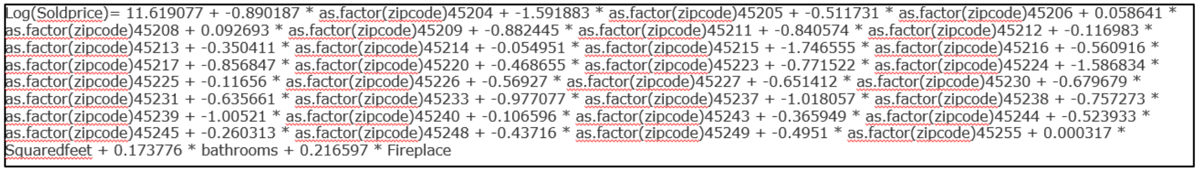
We arrived at the below equations finally:

Model 1



```
Log(Soldprice)= 11.441274 + 0.893723 * as.factor(Zipcode)45202 + 0.219113 * as.factor(Zipcode)45203 + -0.523543 * as.factor(Zipcode)45205 + 0.52561 *
as.factor(Zipcode)45206 + 0.465351 * as.factor(Zipcode)45207 + 0.820975 * as.factor(Zipcode)45208 + 0.752138 * as.factor(Zipcode)45209 + -0.187701 *
as.factor(Zipcode)45211 + 0.248165 * as.factor(Zipcode)45212 + 0.488693 * as.factor(Zipcode)45213 + 0.400738 * as.factor(Zipcode)45214 + 0.440458 *
as.factor(Zipcode)45215 + 0.206646 * as.factor(Zipcode)45216 + -0.370833 * as.factor(Zipcode)45217 + 0.083154 * as.factor(Zipcode)45218 + 0.110166 *
as.factor(Zipcode)45219 + 0.620266 * as.factor(Zipcode)45220 + -0.134635 * as.factor(Zipcode)45223 + -0.269569 * as.factor(Zipcode)45224 + -1.026823 *
as.factor(Zipcode)45225 + 0.605946 * as.factor(Zipcode)45226 + 0.365037 * as.factor(Zipcode)45227 + 0.032194 * as.factor(Zipcode)45229 + 0.023443 *
as.factor(Zipcode)45230 + -0.22711 * as.factor(Zipcode)45231 + 0.826262 * as.factor(Zipcode)45232 + -0.084082 * as.factor(Zipcode)45233 + 0.283973 *
as.factor(Zipcode)45236 + -0.736494 * as.factor(Zipcode)45237 + -0.398909 * as.factor(Zipcode)45238 + -0.253899 * as.factor(Zipcode)45239 + 0.009879 *
as.factor(Zipcode)45240 + -0.018657 * as.factor(Zipcode)45241 + 0.089259 * as.factor(Zipcode)45242 + 0.845583 * as.factor(Zipcode)45243 + 0.089665 *
as.factor(Zipcode)45244 + 0.070775 * as.factor(Zipcode)45245 + -0.219987 * as.factor(Zipcode)45246 + 0.499129 * as.factor(Zipcode)45247 + 0.28744 *
as.factor(Zipcode)45248 + 0.530826 * as.factor(Zipcode)45249 + -0.208452 * as.factor(Zipcode)45251 + 0.157024 * as.factor(Zipcode)45255 + -0.113609 *
as.factor(Zipcode)45320 + 0.122588 * bathrooms + 0.000264 * Squaredfeet + -2.2e-05 * I(Age^2)
```

Model 2



```
Log(Soldprice)= 11.619077 + -0.890187 * as.factor(zipcode)45204 + -1.591883 * as.factor(zipcode)45205 + -0.511731 * as.factor(zipcode)45206 + 0.058641 *
as.factor(zipcode)45208 + 0.092693 * as.factor(zipcode)45209 + -0.882445 * as.factor(zipcode)45211 + -0.840574 * as.factor(zipcode)45212 + -0.116983 *
as.factor(zipcode)45213 + -0.350411 * as.factor(zipcode)45214 + -0.054951 * as.factor(zipcode)45215 + -1.746555 * as.factor(zipcode)45216 + -0.560916 *
as.factor(zipcode)45217 + -0.856847 * as.factor(zipcode)45220 + -0.468655 * as.factor(zipcode)45223 + -0.771522 * as.factor(zipcode)45224 + -1.586834 *
as.factor(zipcode)45225 + -0.11656 * as.factor(zipcode)45226 + -0.56927 * as.factor(zipcode)45227 + -0.651412 * as.factor(zipcode)45230 + -0.679679 *
as.factor(zipcode)45231 + -0.635661 * as.factor(zipcode)45233 + -0.977077 * as.factor(zipcode)45237 + -1.018057 * as.factor(zipcode)45238 + -0.757273 *
as.factor(zipcode)45239 + -1.00521 * as.factor(zipcode)45240 + -0.106596 * as.factor(zipcode)45243 + -0.365949 * as.factor(zipcode)45244 + -0.523933 *
as.factor(zipcode)45245 + -0.260313 * as.factor(zipcode)45248 + -0.43716 * as.factor(zipcode)45249 + -0.4951 * as.factor(zipcode)45255 + 0.000317 *
Squaredfeet + 0.173776 * bathrooms + 0.216597 * Fireplace
```

Data Exploration & Data Cleaning

This dataset has 18 covariates and 389 observations collected manually from the mentioned websites. The dataset initially had 606 observations collected by all the students in class. Since there was a lot of data entry & duplication issues in the collected data, data cleaning was very important at this stage. Hence as part of Data Cleaning step, all such problematic records were deleted. Here is the small sample of the dataset used and the type of columns used for further analysis.

```
housing <- read.csv("final_housing.csv",h=T)
head(housing,5)
```

Apart from removal of problematic records, we have also made below updates to standardise the data:
* Segregated Column ‘Parking’ into ‘Parking spaces’ and ‘Parking’ columns which describes the type of parking space and the no. of parking spaces available respectively.
* Tried Collecting more information about the arrangement of cooling and heating in the house through columns named ‘Cooling’ and ‘Heating’ in our dataset but the no. of records with these extra information weren’t enough to be considered in the model hence these were ignored in the final model.
* Created a new column named ‘Age’ from the Original column Yearbuilt by using the formula: Yearbuilt - Currentyear

Following are the descriptions of few important variables used in modelling going forward:
Sold Price: This is the selling price of the house. This is our dependent variable/ y variable for which value is to be predicted.
Zipcode : This field tells the zipcode of the area where the house is located. This is one of the most important variables because its a known fact that if it is a lavishing area, the cost of the property will be high and if it’s not then cost will be low in most of the cases.

```
##   Index Zipcode Soldprice bedrooms bathrooms Stories Yearbuilt Age Squaredfeet
## 1     1   45208    220000        3       2.0       2      1913 106        1276
## 2     8   45245    175000        3       2.5       1      1979  40        1570
## 3     9   45208    399900        3       2.5       2      1916 103        1653
## 4    11   45217    205000        4       2.0       2      1930  89        1920
## 5    12   45238    110000        4       1.5       1      1949  70        1534
##   Lotsizeinsqft         Parking Parkingspaces           Basement    Roof
## 1          3528                             0
## 2         10018 Attached Garage             1 Partially Finished Shingle
## 3          9147                             0
## 4         11326 Attached Garage             1 Partially Finished Shingle
## 5          6098    1/ On Street             0 Partially Finished shingle
##   ExteriorWallType ofFireplaces    neighborhood Cooling Heating
## 1                            NA
## 2           Stucco            1  North Avondale
## 3                            NA                     gas central
```

```
str(housing)
```

```
## 'data.frame':    389 obs. of  19 variables:
##  $ Index          : chr  "1" "8" "9" "11" ...
##  $ Zipcode        : int  45208 45245 45208 45217 45238 45209 45208 45238 45211 45211 ...
##  $ Soldprice      : int  220000 175000 399900 205000 110000 622000 378000 118500 198000 139000 ...
##  $ bedrooms       : int  3 3 3 4 4 5 4 4 3 4 ...
##  $ bathrooms      : num  2 2.5 2.5 2 1.5 5 3 3 2 3 ...
##  $ Stories        : num  2 1 2 2 1 2 2 1 1 2 ...
##  $ Yearbuilt      : int  1913 1979 1916 1930 1949 1913 1925 1950 2007 1929 ...
##  $ Age            : int  106 40 103 89 70 106 94 69 12 90 ...
##  $ Squaredfeet    : int  1276 1570 1653 1920 1534 2505 2011 1314 1276 2270 ...
##  $ Lotsizeinsqft  : int  3528 10018 9147 11326 6098 8494 7535 7623 6843 9147 ...
##  $ Parking        : chr  "" "Attached Garage" "" "Attached Garage" ...
##  $ Parkingspaces  : int  0 1 0 1 0 0 1 1 0 1 ...
##  $ Basement       : chr  "" "Partially Finished" "" "Partially Finished" ...
##  $ Roof           : chr  "" "Shingle" "" "Shingle" ...
##  $ ExteriorWallType: chr  "" "Stucco" "" "Stucco"
```

SquarefeetArea: This field tells the total built in area of the house. Another important feature as the cost of property definitely depends on how big/small the house is.
Bathrooms : This field tells the no. of bathrooms in the house.
Built In: This field tells the year in which the house was constructed. This may/may not be a significant feature because sometimes ancient houses are super costly being antique whereas sometimes they are not.
ParkingSpaces: This field tells the total no. of parking spaces available with the property purchased,

Let's now have a look at the summary statistics of Selling Price in each area. This gives us an idea about those particular areas where the selling price of houses in general is at higher or lower end. Also it tells us the count of houses in each area/zipcode in our dataset.

```
library(plyr)
ddply(housing, c("Zipcode"), summarise,
               No.ofhouses    = length(Zipcode),
               Soldprice_Mean = mean(Soldprice),
               Soldprice_Max = max(Soldprice),
               Soldprice_Min = min(Soldprice),
               Soldprice_Median = median(Soldprice),
               Soldprice_sd   = sd(Soldprice)
)
```

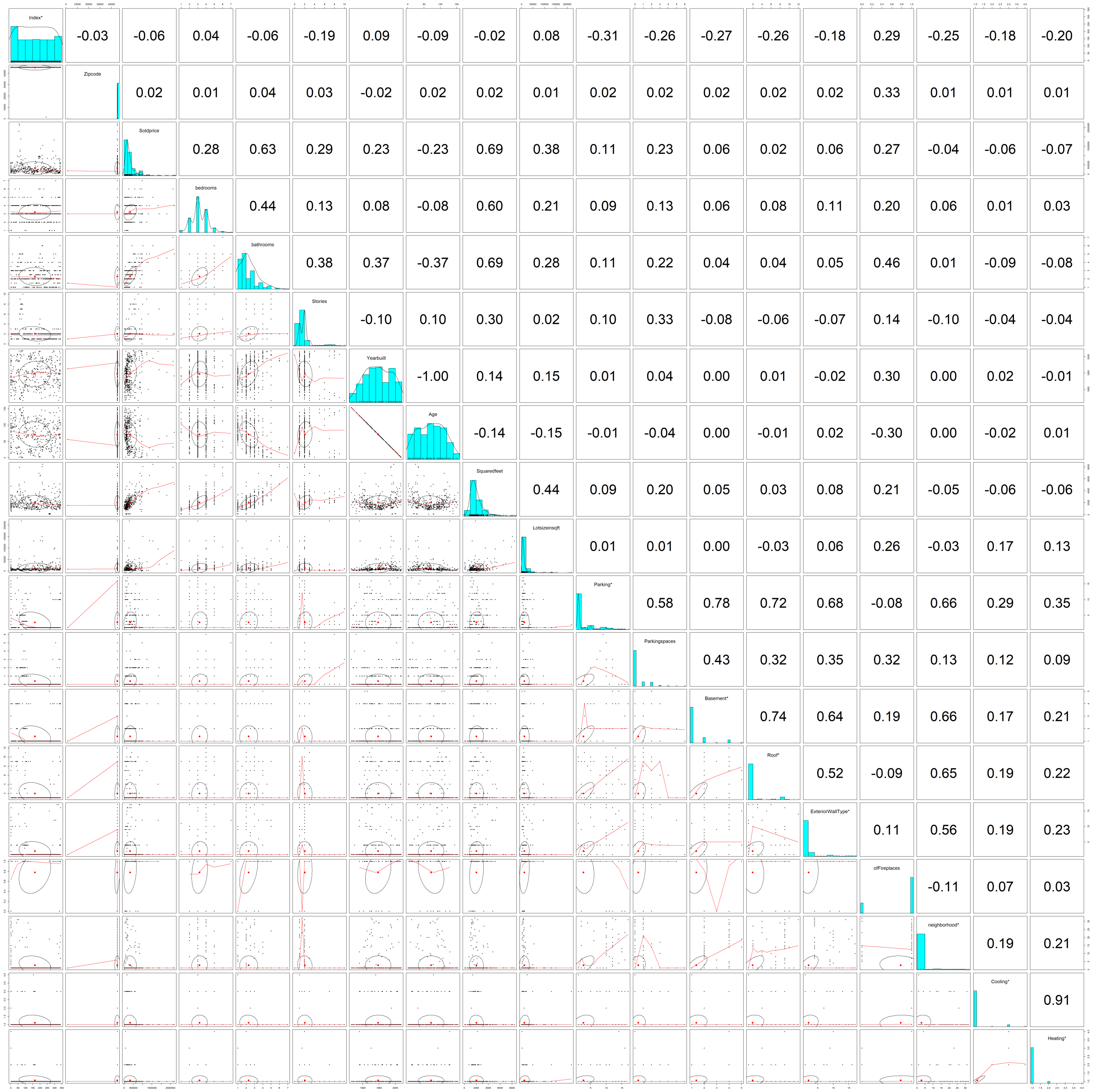
Data Visualization

Now let's look at some visualization graphs to better understand our data:

```
##    Zipcode No.ofhouses Soldprice_Mean Soldprice_Max Soldprice_Min
## 1     1547           1      174900.00        174900        174900
## 2    45002           1      155000.00        155000        155000
## 3    45202          36      621506.56       1890000         86000
## 4    45203          13      268671.15        369000        187000
## 5    45205           9       90211.11        138500         20000
## 6    45206          16      424225.31        945000         26900
## 7    45207           1      285000.00        285000        285000
## 8    45208          25      789866.00       2700000        120000
## 9    45209          11      321590.91        622000        155000
## 10   45211          13      147015.38        247500         52700
## 11   45212           8      259037.50        416000         80000
## 12   45213           3      235723.33        380000        137800
## 13   45214           2      236500.00        298000        175000
## 14   45215           7      264200.00        419000        145000
## 15   45216           2      162000.00        189000        135000
```

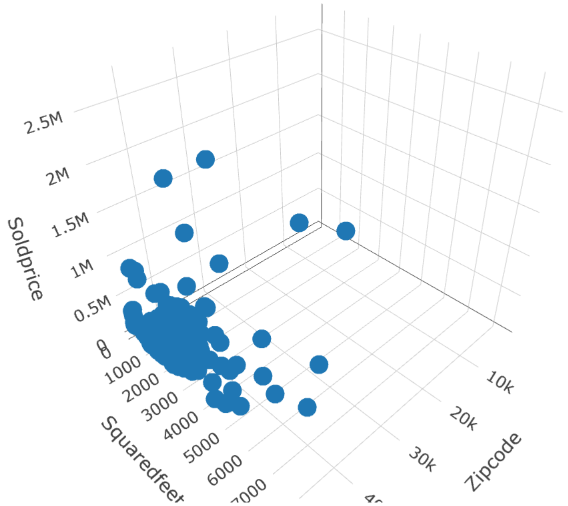
Plotting Scatter Plots , Histograms and Correlation:

```
library(psych)
pairs.panels(housing)
```



Looking at the output of pairs.panels(housing), we notice variables (Cooling, Heating) looks highly correlated with a correlation coeff. of 0.92 but we cannot consider them as we do not have sufficient data for them. Also because of insufficient data, we cannot even say whether this stats is correct or not, next (Basement, Roof) and (Parking, Roof) variables looks somewhat correlated with correlation coeff. of 0.76 and 0.74 respectively which is comparitively high.

Now let’s just try to visualise how would our data distribution in 3d space look like if we plot our response variable Soldprice with two independent variables Zipcode and Squaredfeet.



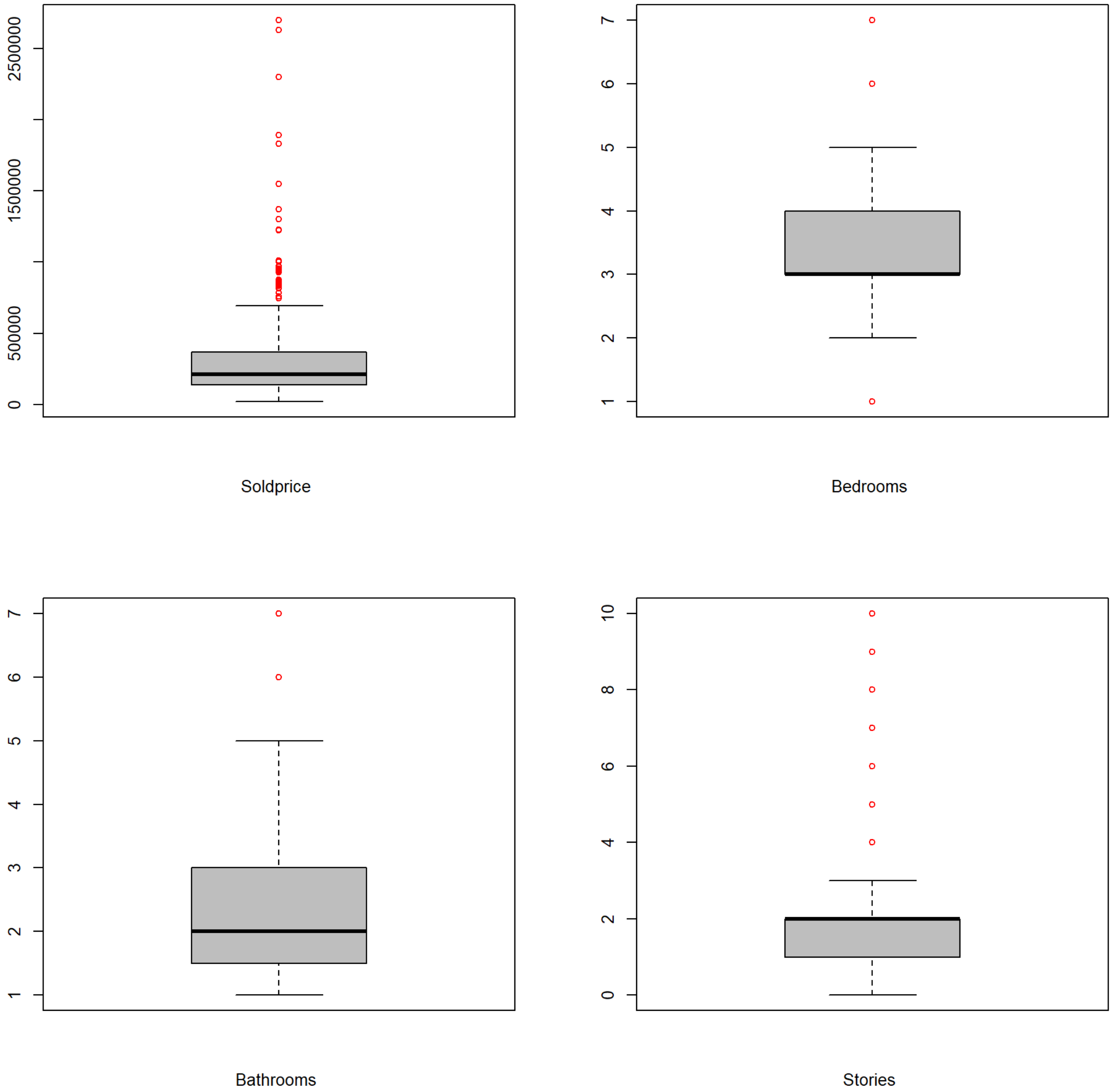
Hover data points to see more info

Plotting BoxPlot for our variables:

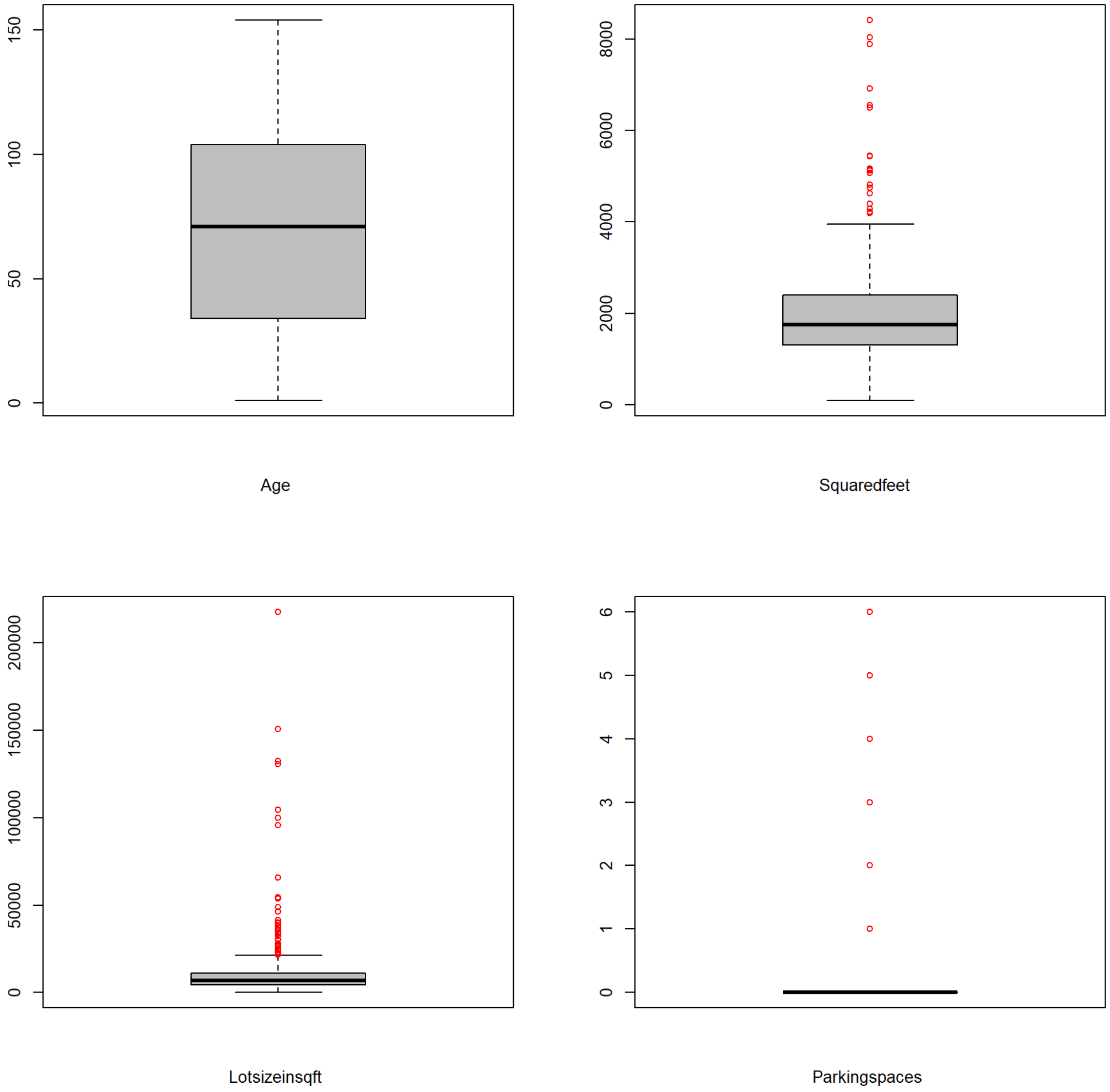
```
attach(housing)
```

```
## The following objects are masked from housing (pos = 5):
##
##     Age, Basement, bathrooms, bedrooms, Cooling, ExteriorWallType,
##     Heating, Index, Lotsizeinsqft, neighborhood, ofFireplaces, Parking,
##     Parkingspaces, Roof, Soldprice, Squaredfeet, Stories, Yearbuilt,
##     Zipcode
```

```
par(mfrow=c(2,2))
boxplot(Soldprice, xlab="Soldprice ",col="grey", outcol="red")
boxplot(bedrooms ,xlab="Bedrooms ",col="grey", outcol="red")
boxplot(bathrooms ,xlab="Bathrooms ",col="grey", outcol="red")
boxplot(Stories ,xlab="Stories ",col="grey", outcol="red")
```



```
boxplot(Age, xlab="Age ",col="grey", outcol="red")
boxplot(Squaredfeet  ,xlab="Squaredfeet  ",col="grey", outcol="red")
boxplot(Lotsizeinsqft  ,xlab="Lotsizeinsqft ",col="grey", outcol="red")
boxplot(Parkingspaces ,xlab="Parkingspaces ",col="grey", outcol="red")
```



In all the above box plots, we see there are few red circles in each of them. These indicate the outliers in respective variables. In the upcoming sections we will see how to deal with these outliers to come up with the best model possible with least error rate.

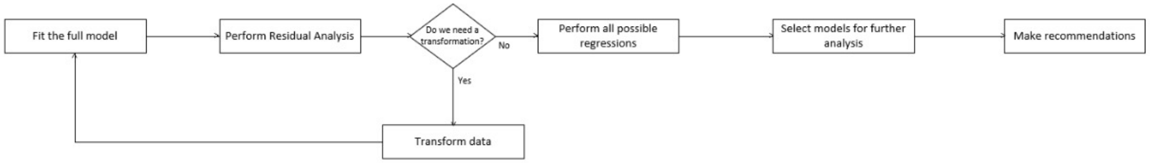
Data Modeling

Understanding the data : There are 389 records with 19 variables in the initial dataset. However, the data corresponding to the 9 variables is very sparse,and we believe that they add a lot of significance to the modelling.

Therefore, we have decided to build two models, one with the 389 rows and first 9 variables which is a dense table, we call this as Data1 and the next model with around 159 rows with all the variables which is also now almost a dense table, let us call this as Data2.

So the entire modelling is based on applying the following steps in each of the data set to receive two separate models: **Model Adequacy Checking - Transformations - Variable Selection - Model Validation - Final Model**

Our plan is to combine the predictions from both these models to arrive at the final prediction.


Fit the full model
Perform Residual Analysis
Do we need a transformation?
No
Yes
Perform all possible regressions
Select models for further analysis
Make recommendations
Transform data

Model Specification on Data1:

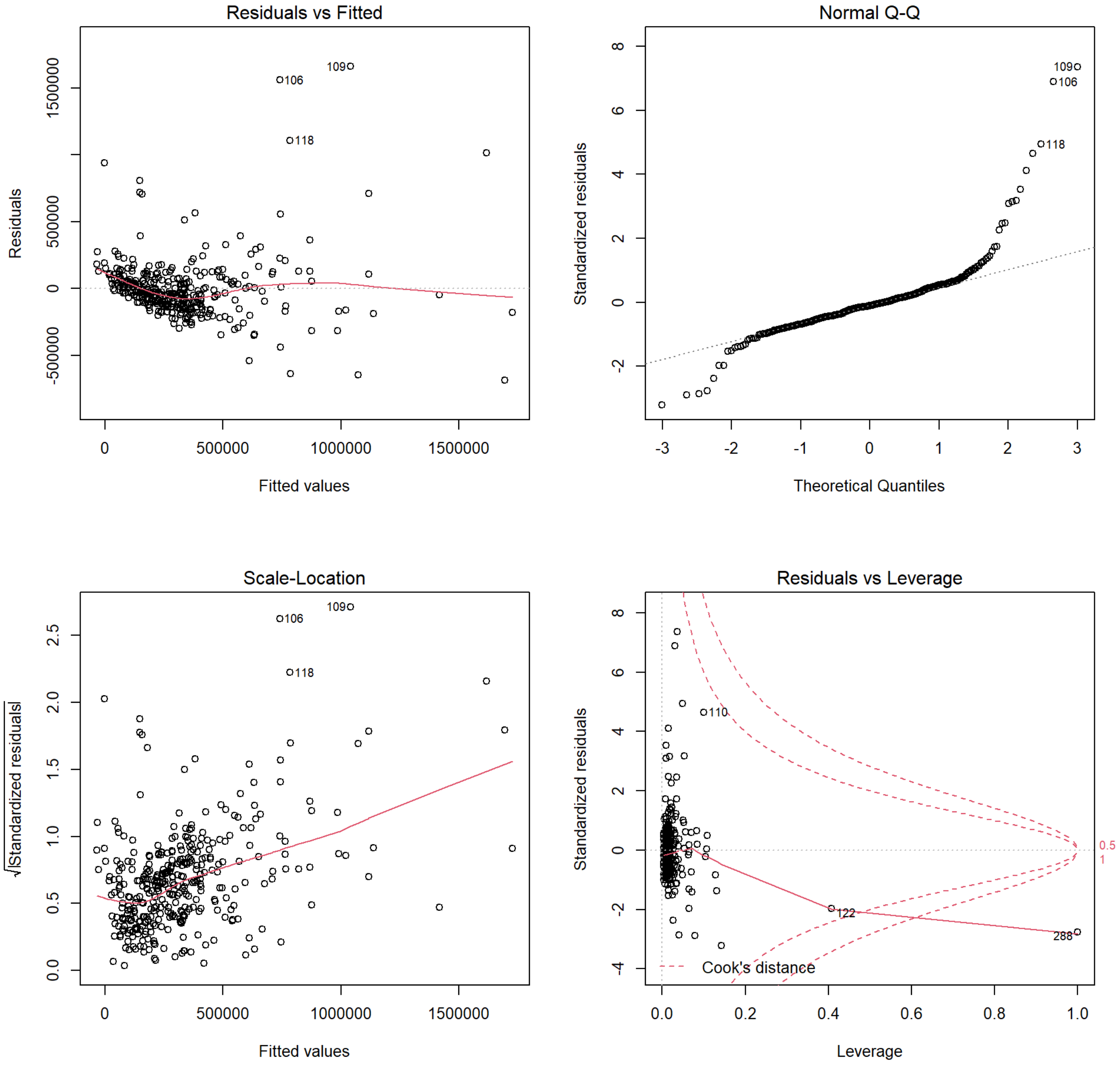
Initially we have included all the base variables and obtained the below model:

```
housing <- read.csv("final_housing.csv",h=T)
attach(housing)
```

```
model<- lm(Soldprice~ Zipcode+bedrooms+bathrooms+Stories+Age+Squaredfeet+Lotsizeinsqft)
```

Residual Analysis on the Initial Model of Data1:

Now as part of Model Adequacy checking we need to validate the LINE assumptions of linear Regression:



Comments on Residual Analysis results on the above model:

1. Linearity & Equal Variance Assumptions : Plot between Residuals Vs Fitted values indicates that the assumptions are not satisfied as the dots are not evenly distributed around zero.

2. Normality Assumption: As we see in the Q-Q Plot, though most of the datapoints are around the 45-degree line, the plot is still heavily skewed on both the ends. So this is not satisfied as well.

3. Outliers: The instance 288 in the fourth plot between Standardized residuals and leverage has a cooks distance greater 1 which clearly poses like an outlier. Hence removing this instance. This will not be used for further analysis.

```
housing[288,]
```

```
##     Index Zipcode Soldprice bedrooms bathrooms Stories Yearbuilt Age
## 288   460    1547    174900        3       1.5       1      1963  56
##     Squaredfeet Lotsizeinsqft Parking Parkingspaces Basement Roof
## 288        1547          7405                      0
##     ExteriorWallType ofFireplaces neighborhood Cooling Heating
## 288                            NA
```

```
housing_new<-housing[-c(288),]
```

We see that the initial model is not good enough to predict the sold price of houses in Cincinnati and has scope of improvement. Therefore, using various methods like Transformations, Variable Selection, Indicator Variable, etc., we will work towards improving the model.

Model Specification on Data2:

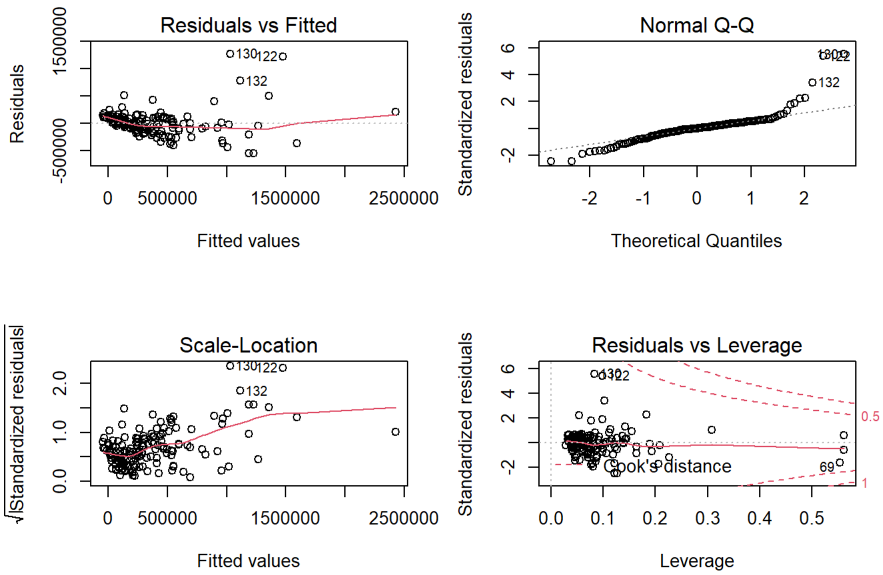
Repeated the same steps as above for the 159 rows and 19 variables (but some of the variables has too less data to include it in the model, hence using only 10 variables rather than 19). Initially we have included all the variables and obtained the below model:

```
model1 <- lm(Soldprice ~zipcode+bedrooms+bathrooms+Age+Squaredfeet+Lotsizeinsqft+Parkingspaces+Basement+Stories+Fireplace)
```

Residual Analysis on the Initial Model of Data2:

Now as part of Model Adequacy checking we need to validate the LINE assumptions of linear Regression.

```
par(mfrow=c(2,2))
plot(model1)
```



The plots here also imply that they do not satisfy the LINE assumptions. Also, no data points has cook distance > 1.Hence we can say there are no outliers.

Model Checking

We see that the LINE assumptions are violated in the above section - Residual Analysis, hence we would be transforming the y and possibly x as suggested by BoxCox.

Transformations on Data1:

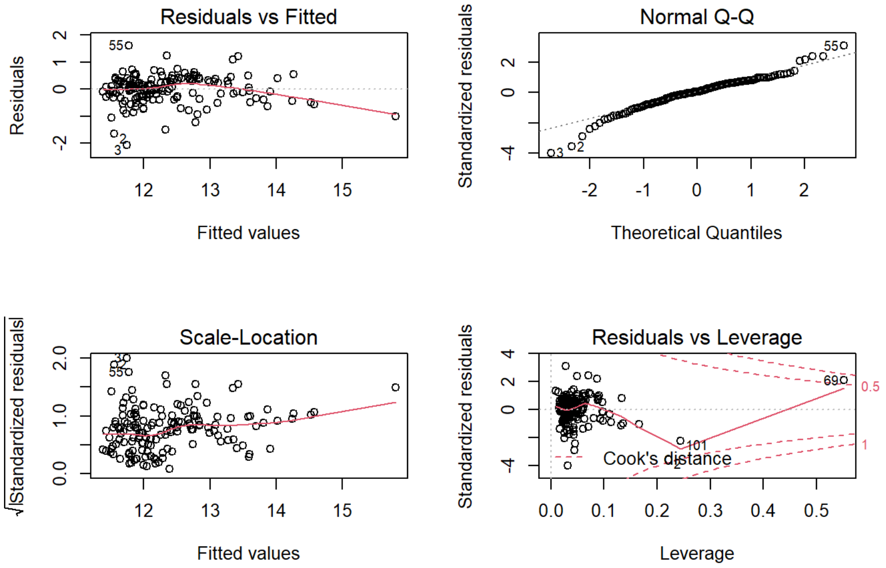
Goal : We want to choose the model that is simple and at the same time closer to the following LINE assumptions.

BOXCOX on the present model:

We have considered to go ahead with the log transformation as the lambda value is close to 0 but the true value is about 0.141, for the ease of understanding of the transformed response variable.

After applying the log transformation on sold price, we create a new model and below are the plots:

```
model<- lm(log(Soldprice)~ zipcode+bedrooms+bathrooms+Stories+Age+Squaredfeet+Lotsizeinsqft)
par(mfrow=c(2,2))
plot(model)
```



```
summary(model)$sigma^2
```

```
## [1] 0.2770548
```

```
standardized_res1=model$residuals/summary(model)$sigma
```

We have adopted trial and error and come up with various combinations of models, but keeping in mind the goal of this step ie., to achieve a model as simple as possible, we have decided on the below model:

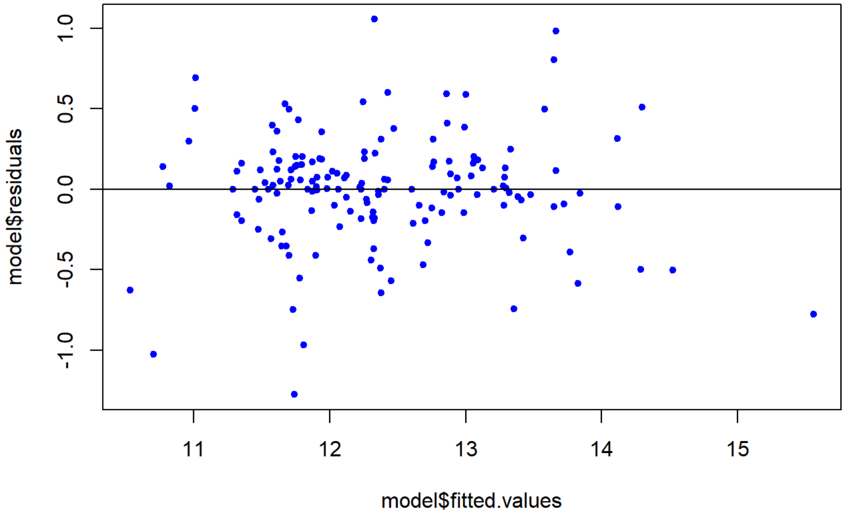
zipcode - as.factor

Age - Squared

```
model<- lm(log(Soldprice)~ as.factor(zipcode)+bedrooms+(bathrooms)+(Stories)+I(Age^2)+Squaredfeet+Lotsizeinsqft)
summary(model)$sigma^2
```

```
## [1] 0.1544141
```

```
standardized_res2=model$residuals/summary(model)$sigma
plot(model$fitted.values,model$residuals,pch=20,col="blue")
abline(h=0)
```

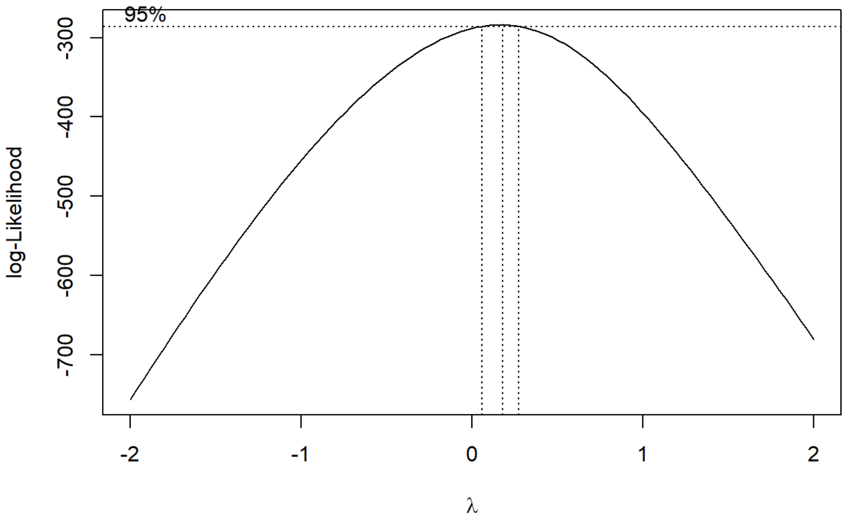


The transformed data did improve very much on the Residuals vs Fitted Values graph. At this point we are moving ahead with this data.

Transformations on Data2:

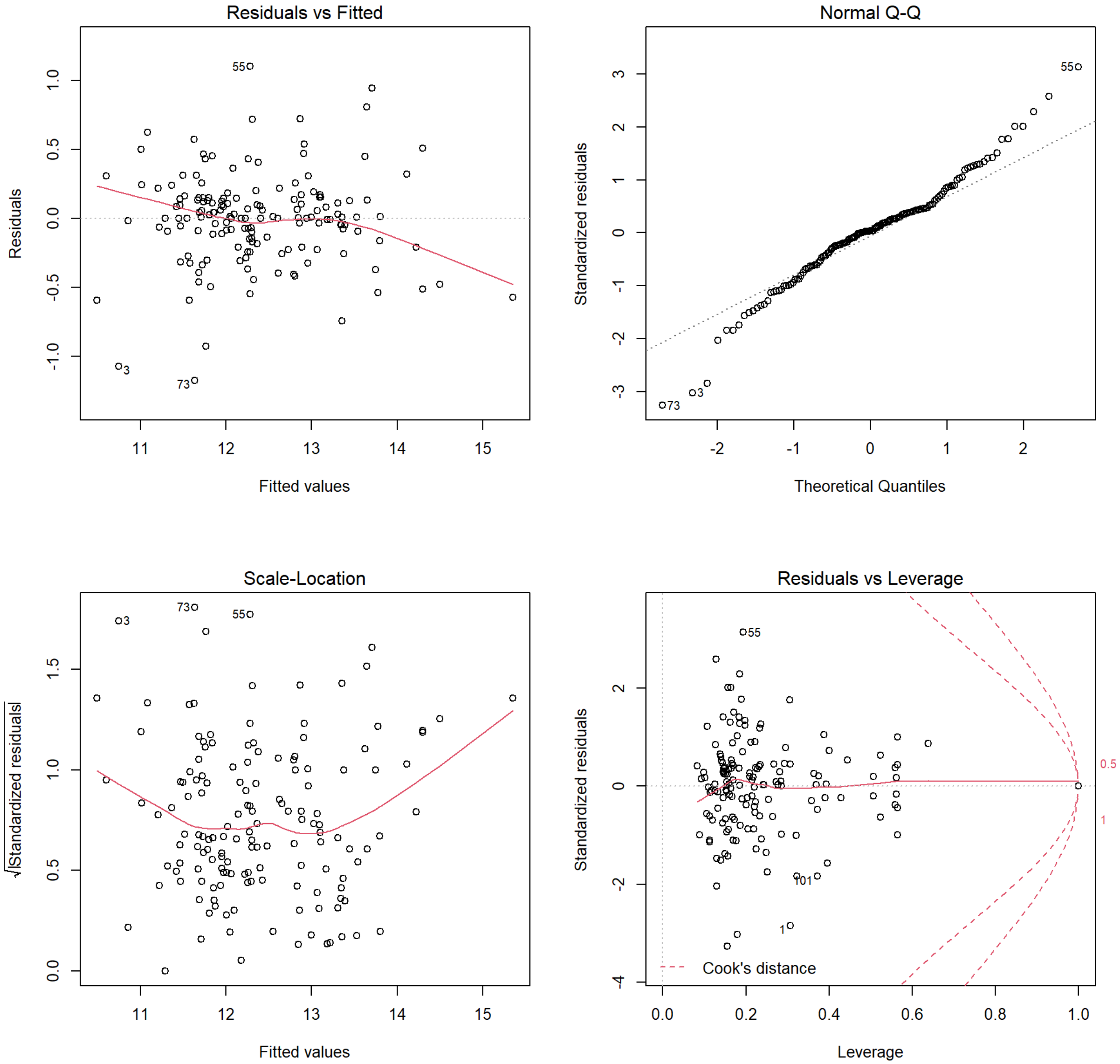
We will be transforming the x and y values to improve the model but to know which transformation to use, we perform the boxcox.

```
boxcox(model1)
```



```
model1<- lm(log(Soldprice)~ as.factor(zipcode)+bedrooms+bathrooms+I(Age^2)+Squaredfeet+Lotsizeinsqft+Parkingspaces+Basement+
Stories+Fireplace)
```

Similar to the rationale for the Data1, we have used log transformation on response variable and as.factor for Zipcode and square of age for the sake of simplicity. And the below graph indicates an improvement on the assumptions.



Re-modeling

We will be re-modeling the models created earlier to get better prediction model using Variable Selection.

Variable Selection on Data1 (after transformations):

We chose Stepwise regression to do the Variable Selection: Started with the null model and added the most significant variable at each step from the Data1 (after transformations).

Stopped the steps as we see that there are no more significant coeffecients for the regressors.

```
add1(lm(log(Soldprice)~as.factor(Zipcode)+bathrooms+Squaredfeet+I(Age^2)), log(Soldprice)~ as.factor(Zipcode)+bedrooms+bathr
ooms+Stories+I(Age^2)+Squaredfeet+Lotsizeinsqft, test="F")
```

```
## Single term additions
##
## Model:
## log(Soldprice) ~ as.factor(Zipcode) + bathrooms + Squaredfeet +
##     I(Age^2)
##               Df Sum of Sq    RSS     AIC F value Pr(>F)
## <none>                     70.750 -526.75
## bedrooms       1 0.0082437 70.742 -524.79  0.0393 0.8430
## Stories        1 0.0006969 70.750 -524.75  0.0033 0.9541
## Lotsizeinsqft  1 0.0303087 70.720 -524.91  0.1444 0.7042
```

The final model obtained from the Data1 is below:

```
model_final1<-lm(log(Soldprice)~as.factor(Zipcode)+bathrooms+Squaredfeet+I(Age^2))
summary(model_final1)
```

```
##
## Call:
## lm(formula = log(Soldprice) ~ as.factor(Zipcode) + bathrooms +
##     Squaredfeet + I(Age^2))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.16491 -0.19221  0.00611  0.20552  1.88680
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)              1.144e+01  4.658e-01  24.560  < 2e-16 ***
## as.factor(Zipcode)45202  8.937e-01  4.788e-01   1.867 0.062799 .
## as.factor(Zipcode)45203  2.191e-01  4.855e-01   0.451 0.652035
## as.factor(Zipcode)45205 -5.235e-01  4.917e-01  -1.065 0.287708
## as.factor(Zipcode)45206  5.256e-01  4.830e-01   1.088 0.277303
```

Variable Selection on Data2 (after transformations):

```
attach(housing_small)
```

```
## The following objects are masked from housing_new:
##
##     Age, Basement, bathrooms, bedrooms, Lotsizeinsqft, Parkingspaces,
##     Soldprice, Squaredfeet, Stories
```

```
## The following objects are masked from housing (pos = 5):
##
##     Age, Basement, bathrooms, bedrooms, Lotsizeinsqft, Parkingspaces,
##     Soldprice, Squaredfeet, Stories
```

```
## The following objects are masked from housing (pos = 8):
##
##     Age, Basement, bathrooms, bedrooms, Lotsizeinsqft, Parkingspaces,
##     Soldprice, Squaredfeet, Stories
```

Stopped the steps as we see that there are no more significant coeffecients for the regressors.

```
add1(lm(log(Soldprice)~as.factor(zipcode)+Squaredfeet+bathrooms+Fireplace), log(Soldprice)~ as.factor(zipcode)+bedrooms+bath
rooms+I(Age^2)+Squaredfeet+Lotsizeinsqft+Parkingspaces+Basement+Stories+Fireplace, test="F")
```

```
## Single term additions
## 
## Model:
## log(Soldprice) ~ as.factor(zipcode) + Squaredfeet + bathrooms + 
##     Fireplace
##               Df Sum of Sq    RSS     AIC F value Pr(>F)
## <none>                     18.020 -273.04               
## bedrooms       1   0.00246 18.018 -271.06  0.0166 0.8976
## I(Age^2)       1   0.36437 17.656 -274.26  2.5178 0.1152
## Lotsizeinsqft  1   0.07258 17.948 -271.67  0.4934 0.4838
## Parkingspaces  1   0.08850 17.931 -271.81  0.6022 0.4393
## Basement       4   0.28786 17.732 -267.58  0.4830 0.7482
## Stories        1   0.00278 18.017 -271.06  0.0188 0.8911
```

```
drop1(lm(log(Soldprice)~as.factor(zipcode)+Squaredfeet+bathrooms+Fireplace),data=housing_small,test="F")
```

```
## Single term deletions
## 
## Model:
## log(Soldprice) ~ as.factor(zipcode) + Squaredfeet + bathrooms + 
##     Fireplace
##                    Df Sum of Sq    RSS     AIC F value    Pr(>F)    
## <none>                          18.020 -273.04                      
## as.factor(zipcode) 31   27.7455 45.766 -187.77  6.1092 1.492e-13 ***
## Squaredfeet         1    5.8262 23.846 -230.77 39.7681 4.673e-09 ***
## bathrooms           1    2.1810 20.201 -256.98 14.8868 0.0001832 ***
## Fireplace           1    1.2834 19.303 -264.17  8.7601 0.0036948 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The final model obtained from the Data2 is below:

```
model_final<-lm(log(Soldprice)~as.factor(zipcode)+Squaredfeet+bathrooms+Fireplace)
summary(model_final)
```

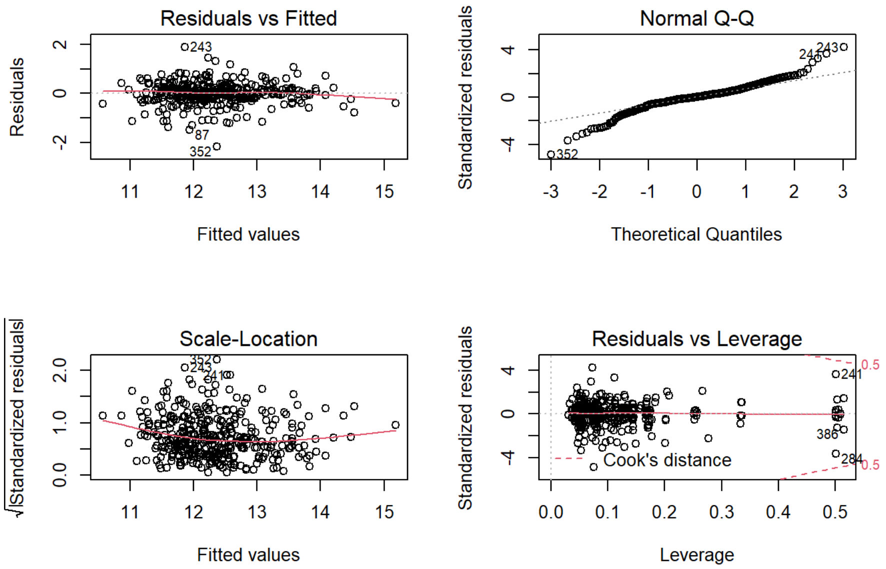
```
## 
## Call:
## lm(formula = log(Soldprice) ~ as.factor(zipcode) + Squaredfeet + 
##     bathrooms + Fireplace)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -1.1845 -0.1313  0.0000  0.1431  1.1090 
## 
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)    
## (Intercept)             1.162e+01  1.377e-01  84.406  < 2e-16 ***
## as.factor(zipcode)45204 -8.902e-01  2.121e-01  -4.197 5.15e-05 ***
## as.factor(zipcode)45205 -1.592e+00  1.738e-01  -9.158 1.45e-15 ***
## as.factor(zipcode)45206 -5.117e-01  1.633e-01  -3.133 0.002162 ** 
## as.factor(zipcode)45208  5.864e-02  1.188e-01   0.494 0.622385
```

Model Checking & Validaton

For Data1, model_final1:

```
par(mfrow=c(2,2))
plot(model_final1)
```

```
## Warning: not plotting observations with leverage one:
##   41, 71, 250, 312, 382
```



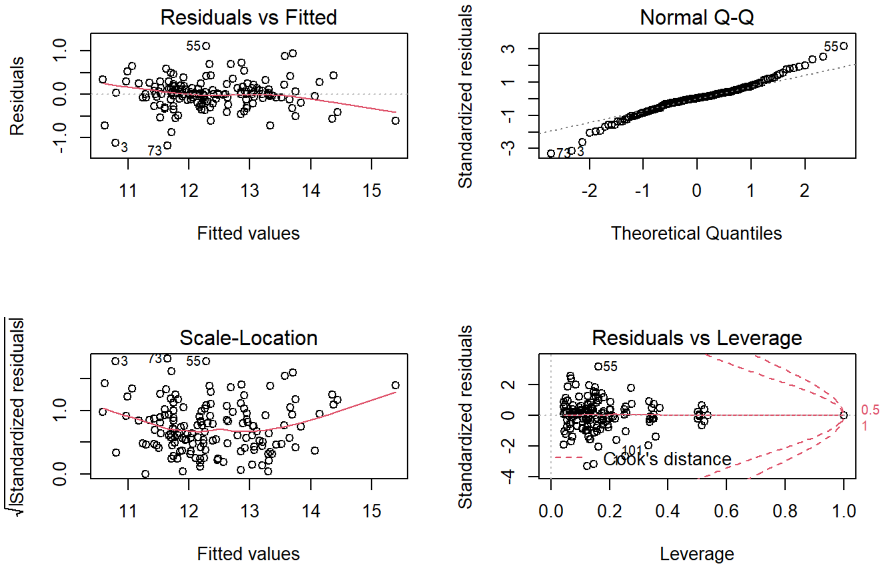
For Dat2, model_final1:

```
par(mfrow=c(2,2))
plot(model_final)
```

```
## Warning: not plotting observations with leverage one:
##   19, 94, 108, 111, 123, 147, 153
```

```
## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced

## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced
```



We notice that the QQ-plot is quite good and other plots also look good for both the models.

In order to validate the model, we have collected 15 new data points manually similar to our raw data acquisition and have performed the predictions on them using the final models after variable selection step.

We will calculate the MPSE, PRESS, and R-square values to validate the two models.

Predicted residual error sum of squares (PRESS) - form of cross-validation used in regression analysis to provide a summary measure of the fit of a model to a sample of observations that were not themselves used to estimate the model.The lower value of PRESS represents the good model.

Mean squared prediction error (MSPE) - the expected value of the squared difference between the fitted values implied by the predictive function and the values of the unobservable function. Lower MSPE is favorable.

Prediction R square - how well a regression model predicts responses for new observations. This statistic helps us determine when the model fits the original data but is less capable of providing valid predictions for new observations. High value of R-square is favorable.

Validation of 15 datapoints on the model built using DATA1

```
pred<- predict(model_final1,test_data,interval = c("confidence"), level = 0.95, type="response")

prediction_error_actual_sold_price = test_data$Soldprice-exp(pred[,1])
prediction_error_log_scale = log(test_data$Soldprice)-pred[,1]
head(cbind(Actual_soldprice=test_data[,6],exp(pred),prediction_error_actual_sold_price,pred,prediction_error_log_scale),15)
```

The table has the original data along with the predictions on both actual and log scale along with the intervals at 95% confidence.

We have low MSPE and PRESS rations and high R-square value with 71.78% which means we expect the model to explain about 71.78% of the variability in prediction of a new observation.

Validation of 15 datapoints on the model built using DATA2

The table has the original data along with the predictions on both actual and log scale along with the intervals at 95% confidence.

We have low MSPE and PRESS rations and high R-square value with 86.03% which means we expect the model to explain about 86.03% of the

```
##    Actual_soldprice      fit       lwr      upr
## 1            360000 455428.7 347052.08 597648.9
## 2            249000 249025.2  99891.54 620808.6
## 3            190000 211735.2 110152.94 406996.0
## 4            291000 264864.4 154348.70 454510.7
## 5            465000 360854.4 189256.07 688040.9
## 6            145100 132651.2  91357.11 192610.6
## 7            431000 352146.1 240935.74 514688.7
## 8            550000 448662.1 322181.71 624795.6
## 9            119000 161362.1 133678.69 194778.5
## 10           215000 208235.1 131791.95 329017.5
##    prediction_error_actual_sold_price      fit      lwr      upr
## 1                        -95428.68008 13.02899 12.75723 13.30076
## 2                           -25.15746 12.42531 11.51184 13.33878
## 3                        -21735.23319 12.26309 11.60963 12.91656
## 4                         26135.61189 12.48697 11.94697 13.02698
```

```
MSPE = sum( (log(test_data$Soldprice) - pred[,1])^2 ) / dim(test_data)[1]
MSPE
```

```
## [1] 0.03242767
```

```
PRESS = sum( (log(test_data$Soldprice) - pred[,1])^2)
PRESS
```

```
## [1] 0.3242767
```

```
pred_Rsq = 1-PRESS/sum((log(test_data$Soldprice)-mean(log(test_data$Soldprice)))^2)
pred_Rsq
```

```
## [1] 0.8618709
```

```
summary(model_final1)$r.squared
```

```
## [1] 0.7178256
```

```
test_data <- read.csv("test_data2.csv",h=T)
```

```
pred<- predict(model_final,test_data,interval = c("confidence"), level = 0.95, type="response")
pred
```

```
##        fit      lwr      upr
## 1 13.01507 12.56598 13.46415
## 2 11.13540 10.35953 11.91128
## 3 12.71474 12.25896 13.17053
## 4 13.26778 12.63891 13.89664
## 5 11.78227 11.02243 12.54211
## 6 13.02107 12.61118 13.43096
## 7 12.75671 12.28077 13.23265
## 8 11.83019 11.55107 12.10930
```

variability in prediction of a new observation.

```
prediction_error_actual_sold_price = test_data$Soldprice-exp(pred[,1])
prediction_error_log_scale = log(test_data$Soldprice)-pred[,1]
head(cbind(Actual_soldprice=test_data[,6],exp(pred),prediction_error_actual_sold_price,pred,prediction_error_log_scale),15)
```

```
##   Actual_soldprice       fit       lwr       upr
## 1           360000 449129.74 286640.13  703730.9
## 2           190000  68555.91  31556.24  148937.7
## 3           291000 332615.02 210861.18  524671.0
## 4           465000 578259.60 308325.87 1084515.4
## 5           145100 130910.73  61232.27  279878.9
## 6           431000 451834.38 299893.08  680757.0
## 7           550000 346870.03 215510.69  558296.3
## 8           119000 137336.07 103887.97  181553.2
##   prediction_error_actual_sold_price      fit      lwr      upr
## 1                          -89129.74 13.01507 12.56598 13.46415
## 2                          121444.09 11.13540 10.35953 11.91128
## 3                          -41615.02 12.71474 12.25896 13.17053
## 4                         -113259.60 13.26778 12.63891 13.89664
## 5                           14189.27 11.78227 11.02243 12.54211
## 6                          -20834.38 13.02107 12.61118 13.43096
```

```
MSPE = sum( (log(test_data$Soldprice) - pred[,1])^2 ) / dim(test_data)[1]
MSPE
```

```
## [1] 0.174911
```

```
PRESS = sum( (log(test_data$Soldprice) - pred[,1])^2)
PRESS
```

```
## [1] 1.399288
```

```
pred_Rsq = 1-PRESS/sum((log(test_data$Soldprice)-mean(log(test_data$Soldprice)))^2)
pred_Rsq
```

```
## [1] 0.3858557
```

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

```
## [1] 0.8603452
```

Final Model

Model_1 :

```
cc<-model_final$coef
paste("log_sold_price =", paste(cc[1], paste(cc[-1], names(cc[-1]), sep=" * ", collapse=" + "), sep=" + "), "+ e")
```

Model_2 :

Conclusion:

We have successfully built a model for predicting the selling price of a house in an area based on multiple features. At this point we cannot say that this linear regression model is the best model possible for this dataset but yes we did get satisfactory results with different combinations of features

```
## [1] "log_sold_price = 11.6190772878651 + -0.890187223630162 * as.factor(zipcode)45204 + -1.59188322738937 * as.factor(zip
code)45205 + -0.511731107699983 * as.factor(zipcode)45206 + 0.0586409287787164 * as.factor(zipcode)45208 + 0.092692602811741
8 * as.factor(zipcode)45209 + -0.882445040492639 * as.factor(zipcode)45211 + -0.840573991724727 * as.factor(zipcode)45212 +
-0.116982801209134 * as.factor(zipcode)45213 + -0.350411219139989 * as.factor(zipcode)45214 + -0.0549510391202829 * as.facto
r(zipcode)45215 + -1.74655543903114 * as.factor(zipcode)45216 + -0.560915991957784 * as.factor(zipcode)45217 + -0.8568473375
62379 * as.factor(zipcode)45220 + -0.468655275263002 * as.factor(zipcode)45223 + -0.771522296967222 * as.factor(zipcode)4522
4 + -1.58683402624584 * as.factor(zipcode)45225 + -0.116560468940458 * as.factor(zipcode)45226 + -0.569270268051442 * as.fac
tor(zipcode)45227 + -0.651411796141133 * as.factor(zipcode)45230 + -0.679679085674543 * as.factor(zipcode)45231 + -0.6356614
83634158 * as.factor(zipcode)45233 + -0.977077111550304 * as.factor(zipcode)45237 + -1.01805659072529 * as.factor(zipcode)45
238 + -0.757273451543244 * as.factor(zipcode)45239 + -1.00520967190277 * as.factor(zipcode)45240 + -0.106596084201303 * as.f
actor(zipcode)45243 + -0.365948785282052 * as.factor(zipcode)45244 + -0.523932925782483 * as.factor(zipcode)45245 + -0.26031
3101709197 * as.factor(zipcode)45248 + -0.437159786461135 * as.factor(zipcode)45249 + -0.495100035615371 * as.factor(zipcod
e)45255 + 0.000316903869909259 * Squaredfeet + 0.173775996062323 * bathrooms + 0.21659653986284 * Fireplace + e"
```

```
summary(model_final)
```

```
##
## Call:
## lm(formula = log(Soldprice) ~ as.factor(zipcode) + Squaredfeet +
##     bathrooms + Fireplace)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1845 -0.1313  0.0000  0.1431  1.1090
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)              1.162e+01  1.377e-01  84.406  < 2e-16 ***
## as.factor(zipcode)45204 -8.902e-01  2.121e-01  -4.197 5.15e-05 ***
## as.factor(zipcode)45205 -1.592e+00  1.738e-01  -9.158 1.45e-15 ***
## as.factor(zipcode)45206 -5.117e-01  1.633e-01  -3.133 0.002162 **
## as.factor(zipcode)45208  5.864e-02  1.188e-01   0.494 0.622385
```

```
cc<-model_final1$coef
paste("log_sold_price =", paste(cc[1], paste(cc[-1], names(cc[-1]), sep=" * ", collapse=" + "), sep=" + "), "+ e")
```

```
## [1] "log_sold_price = 11.4412740845421 + 0.893723280907505 * as.factor(Zipcode)45202 + 0.219113075620117 * as.factor(Z
ipcode)45203 + -0.523542691216496 * as.factor(Zipcode)45205 + 0.525610188786905 * as.factor(Zipcode)45206 + 0.46535107096
0193 * as.factor(Zipcode)45207 + 0.820974996375349 * as.factor(Zipcode)45208 + 0.752137980891339 * as.factor(Zipcode)4520
9 + -0.187701153444633 * as.factor(Zipcode)45211 + 0.248165048342641 * as.factor(Zipcode)45212 + 0.488692591105007 * as.f
actor(Zipcode)45213 + 0.400738276500507 * as.factor(Zipcode)45214 + 0.440458042815413 * as.factor(Zipcode)45215 + 0.20664
6350722009 * as.factor(Zipcode)45216 + -0.370832686508302 * as.factor(Zipcode)45217 + 0.0831535778361033 * as.factor(Zipc
ode)45218 + 0.110166112677986 * as.factor(Zipcode)45219 + 0.620266205312194 * as.factor(Zipcode)45220 + -0.13463548067737
8 * as.factor(Zipcode)45223 + -0.269568756047999 * as.factor(Zipcode)45224 + -1.02682315245416 * as.factor(Zipcode)45225
+ 0.605945847401494 * as.factor(Zipcode)45226 + 0.365037465091295 * as.factor(Zipcode)45227 + 0.0321944458098237 * as.fac
tor(Zipcode)45229 + 0.0234427877573101 * as.factor(Zipcode)45230 + -0.22711018364659 * as.factor(Zipcode)45231 + 0.826261
708817784 * as.factor(Zipcode)45232 + -0.0840815007366839 * as.factor(Zipcode)45233 + 0.283973104495782 * as.factor(Zipco
de)45236 + -0.736493811239495 * as.factor(Zipcode)45237 + -0.398909012237115 * as.factor(Zipcode)45238 + -0.2538992644760
39 * as.factor(Zipcode)45239 + 0.009878969718263 * as.factor(Zipcode)45240 + -0.0186566101073018 * as.factor(Zipcode)4524
1 + 0.0892588961273523 * as.factor(Zipcode)45242 + 0.845583050745831 * as.factor(Zipcode)45243 + 0.0896649208434195 * as.
factor(Zipcode)45244 + 0.0707752956744468 * as.factor(Zipcode)45245 + -0.219987294863167 * as.factor(Zipcode)45246 + 0.49
9129065466392 * as.factor(Zipcode)45247 + 0.287439584691759 * as.factor(Zipcode)45248 + 0.53082649825849 * as.factor(Zipc
```

```
summary(model_final1)
```

as described above.

```
##
## Call:
## lm(formula = log(Soldprice) ~ as.factor(Zipcode) + bathrooms +
##     Squaredfeet + I(Age^2))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.16491 -0.19221  0.00611  0.20552  1.88680
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             1.144e+01  4.658e-01  24.560  < 2e-16 ***
## as.factor(Zipcode)45202  8.937e-01  4.788e-01   1.867 0.062799 .
## as.factor(Zipcode)45203  2.191e-01  4.855e-01   0.451 0.652035
## as.factor(Zipcode)45205 -5.235e-01  4.917e-01  -1.065 0.287708
## as.factor(Zipcode)45206  5.256e-01  4.830e-01   1.088 0.277303
```