Ames Housing Project

Sijia Liang, Yuqing Weng

3/27/2018

1 Introduction

This AmesHousing dataset contains about 80 explanatory variables to describe almost every aspect of the house. This dataset describes 2340 residential property sales in Ames, Iowa between 2006 and 2010.

Our goal is to build a predictive model on sale prices of houses using multiple linear regression. Objective of this project is to identify the most important features and best regression model.Less than 15 attributes were used as input in the best fitted model.

2 Exploratory Data Analysis

We discovered that there are some numerical features such as following:

1. Square footage (1stFlrSF, GarageArea)
2. Time related features (when the home was built or sold)
3. Room and amenties data (bathrooms)
4. Condition and quality: (rated from 1–10)

Some categorical features such as following:

1. Neighborhood
2. External quality
3. Zoning

```
# load all packages
library(knitr)
library(ggplot2)
library(plyr)
library(dplyr)
library(corrplot)
library(caret)
library(gridExtra)
library(scales)
library(Rmisc)
library(ggrepel)
library(psych)
library(gdata)
library(tidyverse)
library(stringr)
library(lubridate)
library(graphics)
library(randomForest)
```

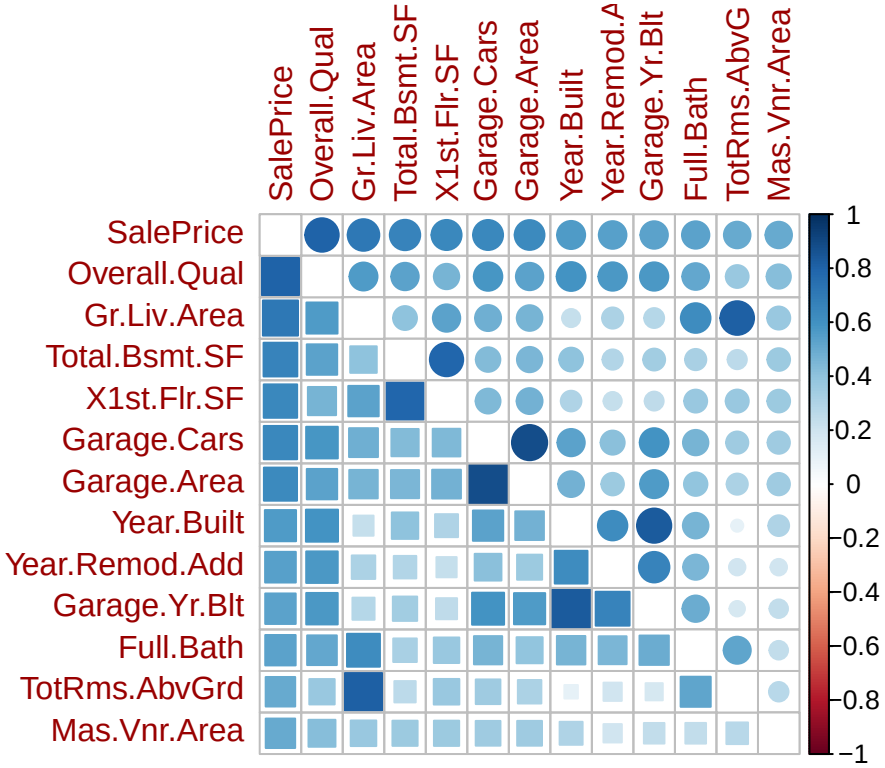
```
# read data using read.csv
train = read.csv(file="/Users/sijialiang/Desktop/Eclipse/MultiVarStats/AmesHousing_Training.csv")
```

Lets first take a look at a correlation plot on all numerical variables.

```
# create a variable named numVars that contains all numerical variables in the dateset
numVars <- which(sapply(train, is.numeric))

# assign the value to numVarNames for further use
numVarNames <- names(numVars)

# print out the numbers of numerical variables
cat('There exist', length(numVars), 'numeric variables in the Ames housing dataset.')
```

```
## There exist 39 numeric variables in the Ames housing dataset.
```

```
train_numVars <- train[, numVars]

# correlations of all numeric variables
cor_numVars <- cor(train_numVars, use="pairwise.complete.obs")

# set it as a decreasing order on correlations with our y
cor_sorted <- as.matrix(sort(cor_numVars[,'SalePrice'], decreasing = TRUE))

# show only high correlations and assign the value to cor_high for use of furture plot
cor_high <- names(which(apply(cor_sorted, 1, function(x) abs(x)>0.5)))
cor_numVars <- cor_numVars[cor_high, cor_high]

# plot using corrplot.mixed() function from 'corrplot' package
corrplot.mixed(cor_numVars, lower = "square", upper = "circle", tl.col="#990000", tl.pos = "lt")
```

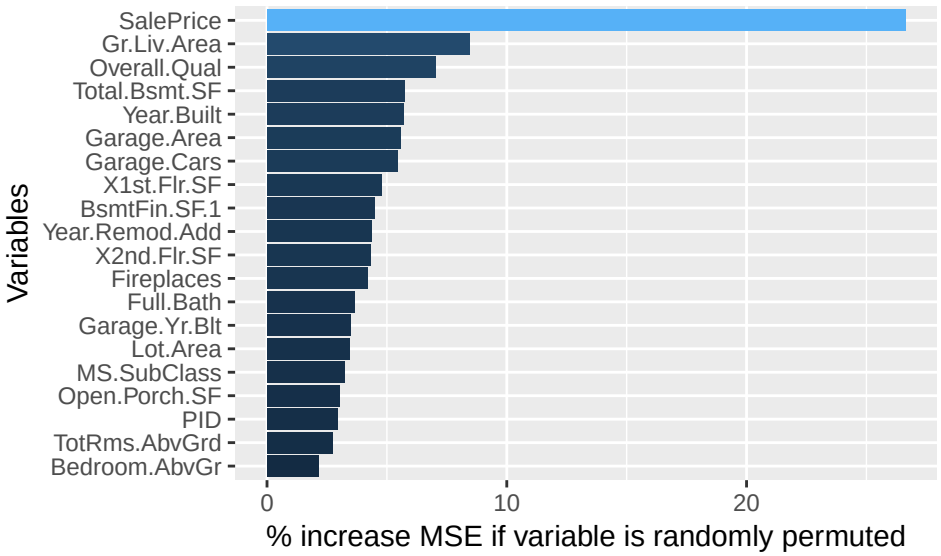


From the correlation plot, we determined that predictors like Overall quality, Ground living area, Total basement square feet and etc have the highest correlation with Sale Price, which make sense.

For multicollinearity, we determined that both GarageCars and GarageArea are highly correlated,same for GrLivArea and TotRmsAbvGrd, and YearBuilt and GarageYrBlt. Due to the reason that these variables are highly correlated and share the similar traits of the housing prices, therefore we decided to use only one of them in later regression model.

3 Data Preprocessing and Random Forest

```
# find numeric columns
train_numeric=select_if(train, is.numeric)

# find NA in columns
#kable(sort(colSums(is.na(train_numeric)),decreasing = T))

# replace NA with 0
train_numeric[is.na(train_numeric)] <- 0

# find character columns
train_char=select_if(train, is.character)

# find NA in columns
#kable(sort(colSums(is.na(train_char)),decreasing = T))

# replace NA with 0
train_char[is.na(train_char)] <- 0

# convert to factor
train_char[] <- lapply(train_char, factor)

# combine numerical and char
train_new=cbind(train_numeric,train_char)

# take a look
#kable(head(train_new), caption = "Training Data Set")

# set random seed, lets say this year
set.seed(2018)

# quick and dirty randomForest model with 88 small number of trees
quick_randomForest <- randomForest(x=train_new[1:2340,-79], y=train_new$SalePrice[1:2340], ntree=88,impo

# set importance on the quick randomForest
importance_randomForest <- importance(quick_randomForest)
importance_DF <- data.frame(Variables = row.names(importance_randomForest), MSE = importance_randomFores

# set in decending order display
importance_DF <- importance_DF[order(importance_DF$MSE, decreasing = TRUE),]

# plot important variables using ggplot from 'ggplot2' package
ggplot(importance_DF[1:20,], aes(x=reorder(Variables, MSE), y=MSE, fill=MSE)) + geom_bar(stat = 'identi
```



Clearly Overall.Qual, Year.Built, Gr.Liv.Area, X1st.Flr.SF, Garage.Area, Year.Remod.Add, should be picked as variables. Then we randomly select the rest of categorical and numerical variables on our preference such as MS.SubClass, Lot.Area.

4 Feature Selection and Visualization

Based on the correlation plot, randomForest and some common sense, we decided the following subset of the variables as predicators:

- Overall Cond - Overall condition rating
- Year Built - Original construction date
- X1st.Flr.SF - First floor square feet
- Bedroom Abv Gr - Number of bedrooms above basement level
- Gr Liv Area - Above grade (ground) living area square feet
- Kitchen Abv Gr - Number of kitchens above grade
- Pool Area - Pool area in square feet
- Lot Area - Lot size in square feet
- BsmtFin.SF.1
- Neighboorhood
- Year Remod Add - Remodel data
- Garage Qual - Garage quality
- MS.SubClass - categorical variable

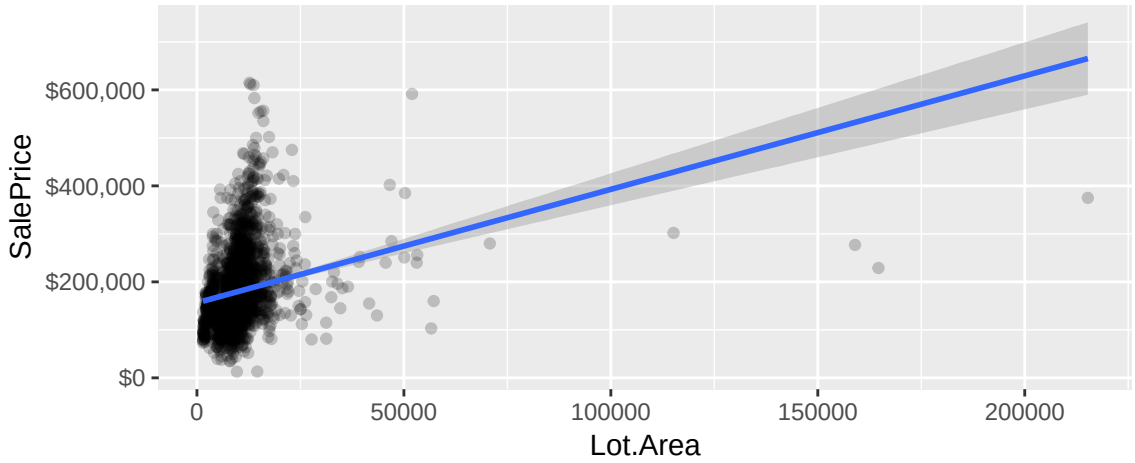
Due to multicollinearity and low correlation between SalePrice, we eliminated the following variables that we initially believed it would be the potential predictors:

- Tot Rms Abv Grd - Total rooms above grade (does not include bathrooms)
- Garage Cars - Size of garage in car capacity
- Pool Area - Pool area in square feet

- Bsmt.Unf.SF - Basement square feet
- X2nd.Flr.SF - Second floor square feet

After take a look at the scatter plot of the Lot Area, we decided to take a log on Lot.Area.

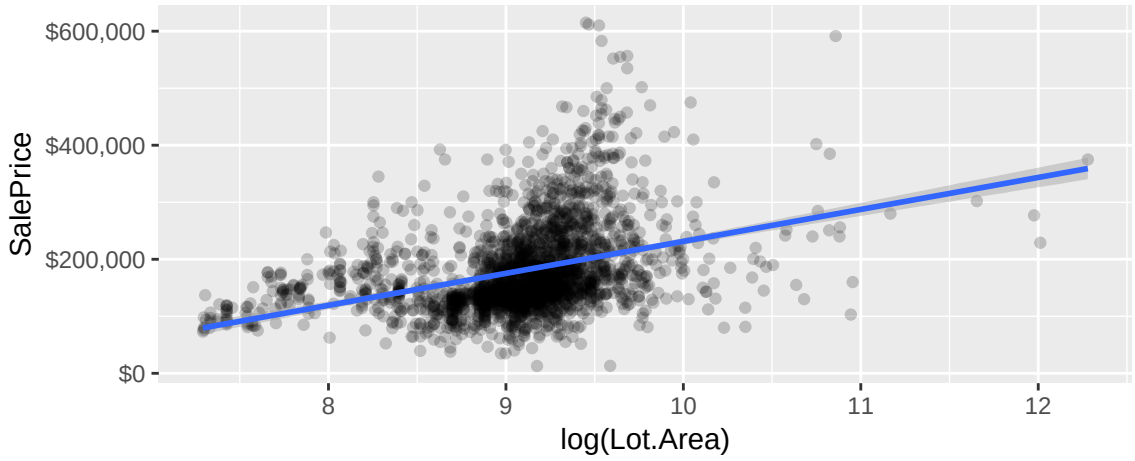
```
# Lot Area
ggplot(train, aes(x = Lot.Area, y = SalePrice)) +
  geom_point(alpha = 0.2) +
  geom_smooth(method = "lm") +
  scale_y_continuous(labels = dollar)
```




```
# distribution of lot area
summary(train$Lot.Area)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1470    7407    9355   10140   11475  215245
```

```
# ln Lot Area
ggplot(train, aes(x = log(Lot.Area), y = SalePrice)) +
  geom_point(alpha = 0.2) +
  geom_smooth(method = "lm") +
  scale_y_continuous(labels = dollar)
```



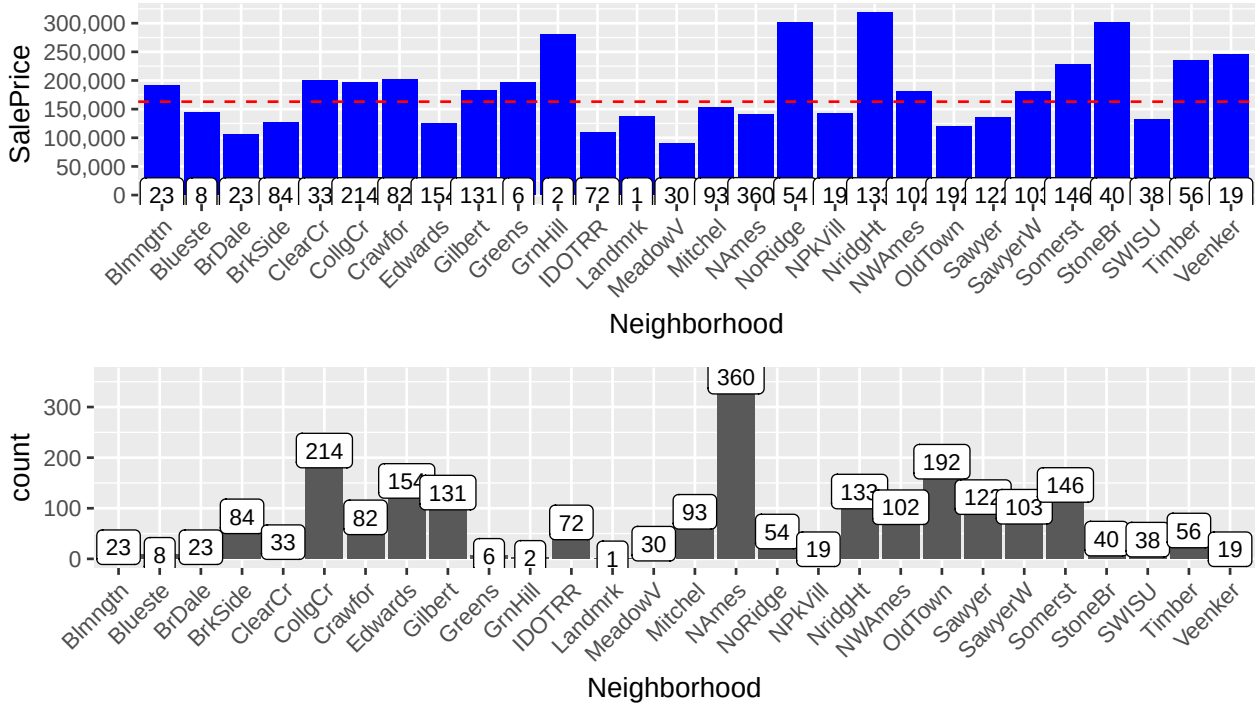
There seems to be some very high end outliers influencing the scatter plot of sale price and lot area–some lot areas are much larger than most. For this reason, we look at the natural log of lot area instead. We also add a variable to represent the natural log of lot area.

Next, we visualize an important categorical variable Neighborhood because usually neighborhood is an important concern when people choosing apartment.

Th first graph shows the median SalePrice by Neighorhood. The frequency (number of houses) of each Neighborhood in the train set is shown in the labels.

The second graph below shows the frequencies across all data.

```
# using ggplot to visualize the median SalesPrice by Neighboorhood
# Note that the dashed line is median SalePrice
n1 <- ggplot(train[!is.na(train$SalePrice),], aes(x=Neighborhood, y=SalePrice)) +
        geom_bar(stat='summary', fun.y = "median", fill='blue') +
        theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
        scale_y_continuous(breaks= seq(0, 800000, by=50000), labels = comma) +
        geom_label(stat = "count", aes(label = ..count.., y = ..count..), size=3) +
        geom_hline(yintercept=163000, linetype="dashed", color = "red")

# frenquencies plot across all data
n2 <- ggplot(data=train, aes(x=Neighborhood)) +
        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))

# arrange multiple grobs on a page, using 'gridExtra' package
grid.arrange(n1, n2)
```



5 Data Cleaning

Next, we would like to construct our selected variable in to one subset.

```
# using  the subset() function to select out variables from ames huge dataset
train <- subset(train, select=c(SalePrice, Lot.Area, Neighborhood, Garage.Area, X1st.Flr.SF, Kitchen.Ab


# take a look at what we have in the new train variable by using head() function
#head(train)

# is there any missing values in our data?
missingValue <- which(colSums(is.na(train)) > 0)

# sort the missing value column in decreasing order
sort(colSums(sapply(train[missingValue], is.na)), decreasing = TRUE)
```

```
## Lot.Frontage  Garage.Area BsmtFin.SF.1
##          402            1            1
```

```
# print out numbers of missingValue column in the selected variables dataset
cat('There exist', length(missingValue), 'columns with missing values in the new train dataset.')
```

```
## There exist 3 columns with missing values in the new train dataset.
```

As we can see there are only 3 column has missingValue, and they are Lot.Frontage, Garage.Qual and BsmtFin.SF.1. We need to fix this.

As NAs mean 'No Garage' for character variables, and the missingValue of Garage.Area is a a house built on 1923, we decided to replace it with zero. Also we will replace 0 for BsmtFin.SF.1 as well. Lot frontage is the linear feet of street connected to property. Although Lot frontage looks like related to neighborhood, bue we do not have any missingValue in neighborhood.

```
# find the NA of Garage.Area
kable(train[is.na(train$Garage.Area), c('Garage.Area','Year.Built')])
```

	Garage.Area	Year.Built
1785	NA	1923

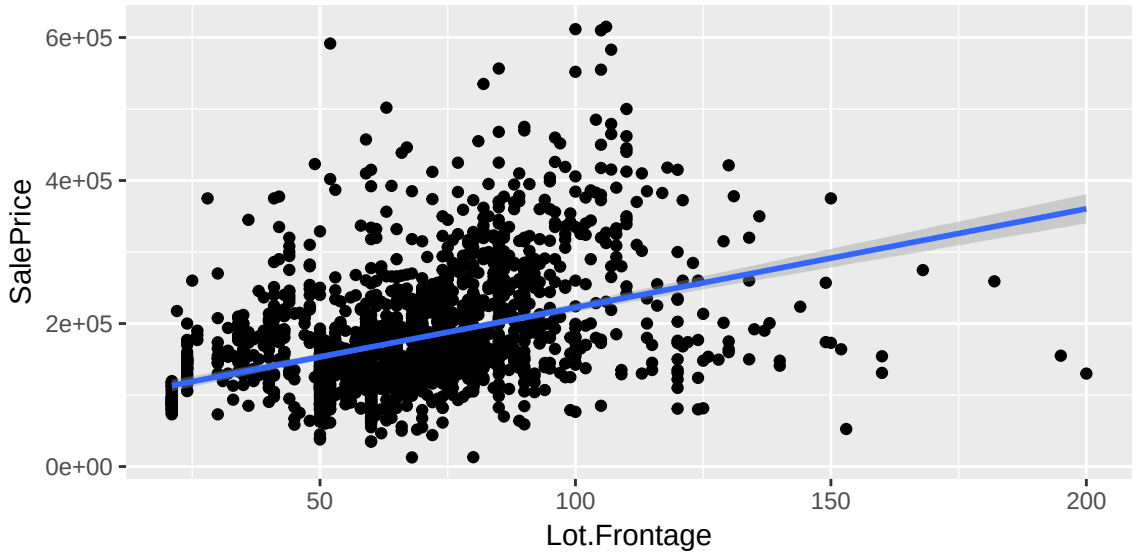
```
# it was built on 1923, maybe its too old, we will replace it with zero
train$Garage.Area[1785] <- 0

kable(train[is.na(train$BsmtFin.SF.1), c('BsmtFin.SF.1','Year.Built')])
```

	BsmtFin.SF.1	Year.Built
1073	NA	1946

```
train$BsmtFin.SF.1[1073] <- 0

# summary Lot.Frontage
summary(train$Lot.Frontage)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   21.00   58.25   68.00   68.91   80.00  200.00     402
```

```
# okay NA's 402, see which ones are missing lot frontage data.
#index <- which(is.na(train$Lot.Frontage))

# look at the head
#head(train[index,])

# plot Lot.Frontage
ggplot(train, aes(x = Lot.Frontage, y = SalePrice)) + geom_point() + geom_smooth(method = "lm")
```




```
# fix MSSubClass categorical variable
train$MS.SubClass <- as.factor(train$MS.SubClass)
 #revalue for better readability
train$MS.SubClass<-revalue(train$MS.SubClass, c('20'='1 story 1946+', '30'='1 story 1945-', '40'='1 sto
```

```
## The following `from` values were not present in `x`: 150
```

```
#str(train$MS.SubClass)
```

6 Linear Model

$$SalePrice_t = \beta_0 + \beta_1 LotArea + \beta_2 Neighborhood + \beta_3 GarageArea + ... + \epsilon$$

```
# fit model without MSSubClass categorical variable
fit1 <- lm(SalePrice ~ I(log(Lot.Area)) + Neighborhood + Garage.Area + X1st.Flr.SF+ Kitchen.AbvGr + Gr.

# fit model with final 14 varibles
fit2 <- lm(SalePrice ~ I(log(Lot.Area)) + Neighborhood + Garage.Area + X1st.Flr.SF+ Kitchen.AbvGr + Gr.

# checking AIC, pick the model with lower AIC
AIC(fit1, fit2)
```

```
##      df      AIC
## fit1 39 44963.84
## fit2 53 44858.79
```

```r
# fit2 has lowest AIC, highest R^2, we picked this as final regression model
summary(fit2)
```

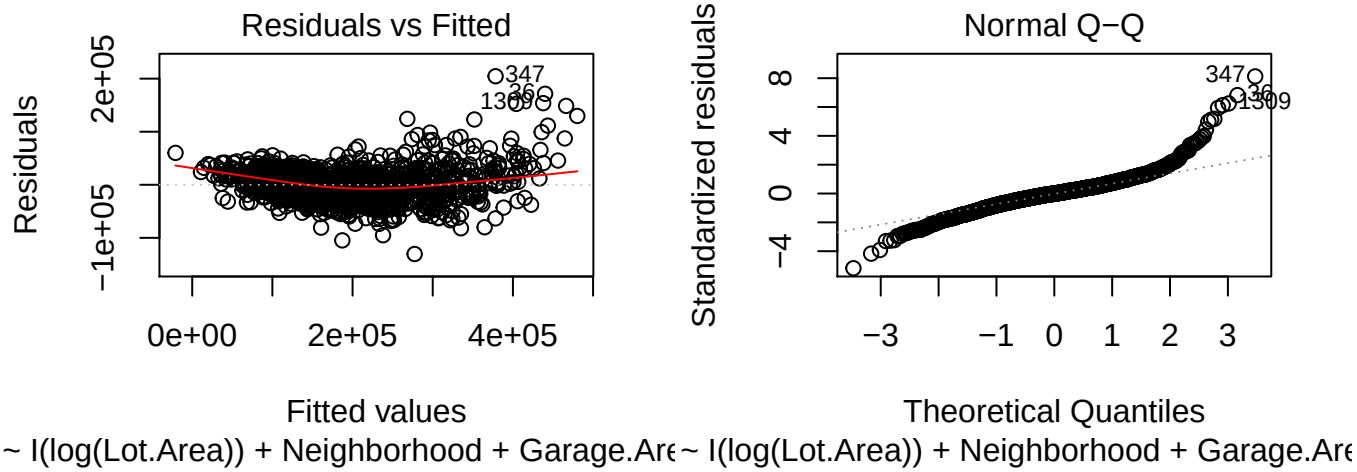
```
##
## Call:
## lm(formula = SalePrice ~ I(log(Lot.Area)) + Neighborhood + Garage.Area +
##     X1st.Flr.SF + Kitchen.AbvGr + Gr.Liv.Area + Bedroom.AbvGr +
##     Year.Remod.Add + Year.Built + Overall.Cond + Overall.Qual +
##     Lot.Frontage + BsmtFin.SF.1 + MS.SubClass, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -130450  -12597    -116   11448  204538
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   -1.605e+06  1.226e+05 -13.097  < 2e-16 ***
## I(log(Lot.Area))               2.014e+04  2.342e+03   8.600  < 2e-16 ***
## NeighborhoodBlueste            7.387e+03  1.129e+04   0.654 0.513020
## NeighborhoodBrDale             8.018e+03  9.377e+03   0.855 0.392626
## NeighborhoodBrkSide           -1.473e+04  7.840e+03  -1.879 0.060418 .
## NeighborhoodClearCr           -2.998e+04  9.915e+03  -3.024 0.002529 **
## NeighborhoodCollgCr           -2.981e+04  7.030e+03  -4.240 2.34e-05 ***
## NeighborhoodCrawfor           -1.195e+02  7.699e+03  -0.016 0.987619
## NeighborhoodEdwards           -2.427e+04  7.364e+03  -3.296 0.000999 ***
## NeighborhoodGilbert           -3.082e+04  7.387e+03  -4.172 3.15e-05 ***
## NeighborhoodGreens            -2.091e+03  1.314e+04  -0.159 0.873610
## NeighborhoodIDOTRR            -2.409e+04  7.945e+03  -3.033 0.002457 **
## NeighborhoodMeadowV            1.356e+04  9.164e+03   1.480 0.139015
## NeighborhoodMitchel           -3.752e+04  7.538e+03  -4.978 7.01e-07 ***
## NeighborhoodNAmes             -2.748e+04  7.159e+03  -3.838 0.000128 ***
## NeighborhoodNoRidge           -3.273e+03  7.977e+03  -0.410 0.681622
## NeighborhoodNPkVill            5.771e+03  9.116e+03   0.633 0.526773
## NeighborhoodNridgHt            2.470e+04  6.983e+03   3.538 0.000414 ***
## NeighborhoodNWAmes            -4.499e+04  7.615e+03  -5.907 4.11e-09 ***
## NeighborhoodOldTown           -2.207e+04  7.558e+03  -2.919 0.003548 **
## NeighborhoodSawyer            -2.699e+04  7.621e+03  -3.541 0.000408 ***
## NeighborhoodSawyerW           -3.549e+04  7.250e+03  -4.895 1.07e-06 ***
## NeighborhoodSomerst           -5.665e+03  6.982e+03  -0.811 0.417295
## NeighborhoodStoneBr            3.913e+04  7.780e+03   5.029 5.40e-07 ***
## NeighborhoodSWISU             -2.629e+04  8.435e+03  -3.117 0.001857 **
## NeighborhoodTimber            -1.621e+04  7.856e+03  -2.063 0.039242 *
## NeighborhoodVeenker           -2.296e+04  1.008e+04  -2.278 0.022865 *
## Garage.Area                    1.923e+01  3.709e+00   5.186 2.38e-07 ***
## X1st.Flr.SF                    2.009e+01  4.138e+00   4.854 1.31e-06 ***
## Kitchen.AbvGr                 -1.554e+03  5.060e+03  -0.307 0.758840
## Gr.Liv.Area                    6.052e+01  3.710e+00  16.312  < 2e-16 ***
## Bedroom.AbvGr                 -7.308e+03  1.041e+03  -7.022 3.04e-12 ***
## Year.Remod.Add                 1.444e+02  4.296e+01   3.360 0.000794 ***
## Year.Built                     5.658e+02  5.602e+01  10.100  < 2e-16 ***
## Overall.Cond                   5.094e+03  6.700e+02   7.603 4.52e-14 ***
## Overall.Qual                   1.377e+04  7.719e+02  17.833  < 2e-16 ***
## Lot.Frontage                  -4.439e+01  3.989e+01  -1.113 0.265908
## BsmtFin.SF.1                   3.107e+01  1.585e+00  19.598  < 2e-16 ***
```

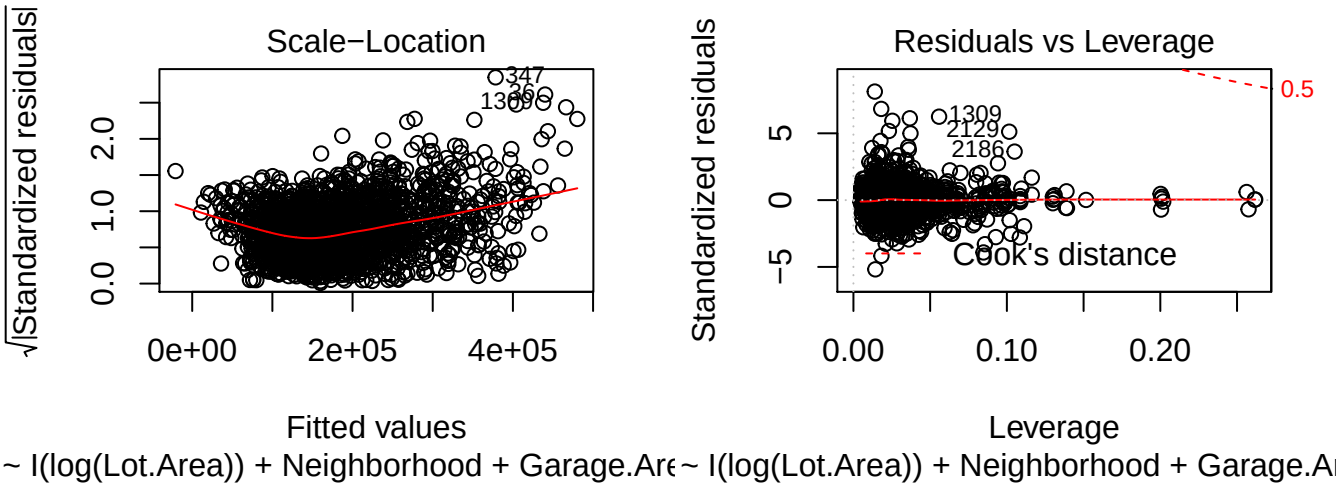
```
## MS.SubClass1 story 1945-         1.375e+04  3.721e+03   3.694 0.000227 ***
## MS.SubClass1 story unf attic    -3.337e+03  1.311e+04  -0.255 0.799042
## MS.SubClass1,5 story unf         1.173e+04  7.287e+03   1.610 0.107557
## MS.SubClass1,5 story fin         1.860e+03  3.285e+03   0.566 0.571327
## MS.SubClass2 story 1946+         1.450e+03  3.622e+03   0.400 0.688919
## MS.SubClass2 story 1945-         4.685e+03  4.574e+03   1.024 0.305765
## MS.SubClass2,5 story all ages    4.931e+03  8.391e+03   0.588 0.556812
## MS.SubClasssplit/multi level    -4.961e+03  3.340e+03  -1.485 0.137638
## MS.SubClasssplit foyer           3.465e+03  5.210e+03   0.665 0.506041
## MS.SubClassduplex all style/age -1.570e+04  5.349e+03  -2.935 0.003380 **
## MS.SubClass1 story PUD 1946+    -3.128e+04  3.401e+03  -9.196  < 2e-16 ***
## MS.SubClass2 story PUD 1946+    -2.724e+04  5.356e+03  -5.086 4.03e-07 ***
## MS.SubClassPUD multilevel       -2.024e+04  8.792e+03  -2.302 0.021440 *
## MS.SubClass2 family conversion  -7.847e+03  4.925e+03  -1.593 0.111251
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25360 on 1886 degrees of freedom
##   (402 observations deleted due to missingness)
## Multiple R-squared:  0.9056, Adjusted R-squared:  0.9031
## F-statistic: 354.8 on 51 and 1886 DF,  p-value: < 2.2e-16
```

while Neighborhood and MSSubClass catergorical data, it automatically changed to dummy, so this is just two variable.

7 Diagonistic Plots

```
plot(fit2)
```





- The first plot indicates the relationship between predictor variables and an outcome variable is approximate linear.
- The QQ-plot looks like we don't have to be concerned too much since it looks normally distributed.
- The cook's distance plot helps us to find influential cases if any. Not all outliers are influential in linear regression analysis. It looks like none of the outliers in our model are influential.

Conclusion

Our final multiple linear model contains 14 variables with R^2 0.9031.

Test the equation

$$\hat{Y} = -1594000 + 20190x_1 - 31050x_2 + 19.3x_3 + 20.09x_4 - 1521x_5 + 60.5x_6$$
$$-7292x_7 + 144.2x_8 + 566.5x_9 + 5094x_{10} + 13780x_{11} - 3644x_{12} + 3.103x_{13} + 1460x_{14}$$

$$\hat{Y} = -1594000 + (20190 * 9.534595) + (-31050) + (19.3 * 482) + (20.09 * 928) + (-1521) + (60.5 * 1629)$$

$$+(-7292*3)+(144.2*1998)+(566.5*1997)+(5094*5)+(13780*5)-(3644*4.304065)+(3.103*791)+1460$$

We did calculate it BY HAND using a calculator, use the training data from the original spreadsheet and manually calculate all transformations andinteractions with our calculator!

Predicted value = $172569.7

Actual value = $189900