# Ames, Iowa housing analysis for Century 21 Ames

Researchers:

Max Pagan - AMaxpagan.github.io

Christian Castro - CDCastr0.github.io

## Introduction:

The purpose of this analysis is to understand the housing market of Ames, Iowa, and determine a model that accurately predicts the sale price of homes in the area for Century 21 Ames (which is a real estate company). The accuracy of each model is judged by the Root-Mean-Squared-Error (RMSE) for predictions vs the actual sale price, which means that the model with the least error is more accurate. These models are crafted using some of the 79 variables to choose from and proceed in one of two ways: 1) Predicting the prices of homes in NAmes, Edwards and BrkSide neighborhoods through finding the relationship between square footage of the home (in 100 sq.ft. increments) and location. This model will provide estimates for the client as well as a confidence interval for the estimates. 2) Predicting the sale prices of homes in all of Ames, doing so through 3 competing models: a simple linear regression model, a multiple linear regression model using sale price, living area, and bathroom count, alongside a third model using lot area, living area, land contour, and land slope. Each model is verified using R^2, CV Press, and Kaggle Score.

## Data Description:

The data comes from the Kaggle website where it is part of an ongoing competition to predict the sale value of houses in the Ames, Iowa area. The data consists of four files: train.csv, test.csv, data_description.txt, and sample_submission.csv. With 79 variables contained in the train dataset, it would be an understatement to say that listing and describing all of them is excessive. The most relevant variables are: “Sale Price” or a property’s sale price in dollars (to be predicted), “neighborhood” or the physical location within the city of Ames, “GrLivArea” or the above ground living area in square feet, “FullBath” or full bathrooms above ground, “LotArea” or lot size in square feet, “LandContour” or flatness of the property, and “LandSlope” or the slope of the property. The training data has 1460 different homes to study as well as an additional 1459 to predict in the test file. For further information, the data can be found on the kaggle website.

Analysis Question 1:

In this analysis, we ventured to find a model that predicted the sale price of homes in the neighborhoods of NAmes, Edwards, and BrkSide using simple linear regression. The way to find this is through understanding the relationship between a couple factors: square footage above ground and neighborhood location of the house. We sought to provide information like average sale price by square footage in increments of 100 sq. ft., and confidence intervals for the estimates while addressing anomalies in the data.

The model consists of:

SalePrice ~ GrLivArea * Neighborhood

As a start, it's important to check the assumptions of simple linear regression. The assumptions are:

1. Linear relationships of the data

By plotting the data in figure 1 and figure 2, each of the neighborhoods have a clear linear relationship between living area and price.

2. Independence of the residuals

Figure 3 provides the rest of the data needed to address the assumptions. The residuals appear to be normally distributed and independent.

3. Homoscedasticity (constant variation of residuals)

Figure 3: "residuals vs leverage" and "Scale location" have features that would suggest a lack of homoscedasticity, requiring correction before evaluation.

4. Normality (residuals are normally distributed)

The "residuals vs fitted" chart lends evidence that the residuals are normally distributed, except for the higher values of the x-axis.

In order to proceed with SLR, the outlier will have to be handled. Because of the high Cook's D and leverage of a few extreme values in Figure 3, we concluded that the simplest and most effective solution would be to identify and remove values above a certain threshold (more info can be found within the comments of the codebook).

The results are shown in Figure 4 as the values are transformed and provide much more appropriate results towards the assumptions of SLR.

To compare the SLR model, we will see how it performs against Adjusted-R^2 and Internal Cross Validation Predicted Residual Error Sum of Squares.

Adjusted R-squared for the simple model was 0.3917

Adjusted R-squared for the complex model was 0.4400

CV PRESS for the simple model was 3.64 * 10^11

CV PRESS for the complex model was 3.41 * 10^11

With these scores in mind, it appears that the complex model is a more accurate model, meaning that the inclusion of the interaction between “GrLivArea” and “Neighborhood” improves the model’s ability to predict sales better than any one factor. The higher R-squared of the complex model suggests it is better suited to understand and predict the variance in sale prices.

The results of the linear regression analysis is that there is statistical evidence that the square footage of the living area of a house has a positive impact on the sale price of houses across the neighborhoods. There is also evidence that the price varies between neighborhoods. The baseline sale price (or intercept) of the model for BrkSide is $19,971. For each additional 100 sq.ft. , there is an associated increase in sale price of ~$8716. In the Edwards neighborhood, homes tend to start around $68,381 and are associated with a $2975 increase per 100 sq. ft. of living area. Within the N Ames neighborhood, home prices baseline at $54,704 and are associated with a ~$5432 increase in price per 100 sq.ft..

This model explains 44.74% of the variance in sale prices as indicated by the R^2 value. WIth an adjusted R^2 of 0.44, this indicates that the model is a good fit. The model provides evidence that as there is an increase in the sq.ft. of a house, there is an increase in the sale price of the house. It also provides evidence that the price differs significantly between neighborhoods.

Based on the model, the starting sale price for a home in the BrkSide neighborhood is $19,971.51, but with 95% confidence between $-4,314.21 to $44,257.24. In the Edwards neighborhood, starting sale prices are $68,381.59, but with 95% confidence between $40,913.67 to $95.849.51. For the NAmes neighborhood, sales begin at a baseline of $54,704.89, but with 95% confidence between $27,408.38 to $82,001.39. Each additional 100 square feet in living area is associated with an increase of $8716 with 95% confidence between $6793 to $10640.

The analysis suggests that both the living area and neighborhood significantly influence house prices in Ames, Iowa. The confidence intervals suggest a high degree of certainty about these effects. While there are differences between neighborhoods, these findings provide insights into the relationships between different variables and how they play into home sale prices. This information can be very valuable for real estate pricing strategies in Ames and we hope Century 21 considers our models.

RShiny:

At this link, you can find the RShiny app we have made that allows you to observe the linear relationship begtween GrLivArea and SalePrice for any neighborhood you choose, and you can enhance the scatterplot by adding a best fit line!

https://amaxpagan.shinyapps.io/RShinyApp/

Analysis 2:

For Analysis 2, we were asked to find the most effective predictive model for home sales prices in Ames, Iowa, encompassing all neighborhoods using Linear regression. We will be comparing three linear regression models to do this: first, we will use a simple linear regression model. Next, we will use a provided multiple linear regression model. Then, we will employ a final multiple linear regression model of our own design.

Linear Model 1

The first linear model we were instructed to create was a simple linear regression with one predictor variable predicting our dependent variable of SalePrice. We decided to perform a log transformation on the year the home was built, YearBuilt, and use that transformation as the independent variable.

The linear model equation can be expressed as:

$$SalePrice = \beta_0 + \beta_1 \times log(YearBuilt)$$

where: - $\beta_0$ is the intercept term, and - $\beta_1$ is the coefficient for $log(YearBuilt)$

$\beta_1$ was found to be 2685933.

We can interpret this model by saying that a 1% increase in the year a house was built is associated with an average increase in sale price of $26,859.

Observing the plots of figure 5, we can address the necessary assumptions of the linear model. One necessary assumption we must address for all models is the independence of each observation. Since each observed data point is its own property with a unique ID number, we know that this assumption will be met for this model and all models going forward. Next, we must address the appearance of a linear relationship between the two variables, which is sufficiently demonstrated in figure 13 in the appendix. The next assumption we will need to address is homoscedasticity. The Standardized Residuals vs Fitted values plot demonstrates a marginally acceptable level of homoscedasticity. We will have to proceed with caution if we continue to use this variable. The QQ plot of residuals demonstrates some degree of normality of the data, and the Residuals vs. leverage plot demonstrates some points with a higher Cook's distance than others, however, the highest Cook's Distance value being 0.08 means this is not particularly a concern. Overall, this model does a less-than-ideal job handling the necessary assumptions for simple linear regression, so we will proceed with caution.

Linear Model 2

For the second linear model, we used a provided outline for a multiple linear regression model. The linear model equation can be expressed as:

$$SalePrice = \beta_0 + \beta_1 \times GrLivArea + \beta_2 \times FullBath +$$

where: - $\beta_0$ is the intercept term, - $\beta_1$ is the coefficient for GrLivArea, and - $\beta_2$ is the coefficient for FullBath

Addressing the assumptions, the multiple linear regression fits the necessary assumptions of linear regression even better. The residuals show a stronger degree of normality and homoscedasticity than the previous model, though still slightly less than ideal. The Model appears to have residuals that slightly grow in variance as the fitted values increase. Additionally, we still have some values that appear to have individually slightly high cook's D values, however the highest of these have been reduced from 0.08 to approximately 0.06. The linear relationship is demonstrated more strongly here than in the previous model.

Linear Model 3

The final model we were asked to construct was an additional multiple linear regression model. In constructing this model, we first visually compared several of the available metrics as predictor variables for price in simple scatterplots. When we looked at LotArea, we found something rather interesting. As demonstrated in the scatterplots below (figures 9 through 12 in the Appendix), LotArea appears to have a steep linear relationship with SalePrice, save for some interesting exceptions. One value had an immense area of over 200,000 square feet, but was sold for much lower than other, much smaller properties. Visually, it appears as if there are two linear relationships within this scatterplot.

Immediately this had us questioning what it was about those occasional large properties that made them worth less than the most expensive homes. After visually comparing the data with many other variables, we determined that the best additional predictors to include were GrLivArea, in order to control for it because it seemed to be partially collinear with LotArea, as well as LandSlope and LandContour. As demonstrated by figures 9 through 12, LandSlope and LandContour appeared to be strong predictors for why the larger properties weren't worth as much as some smaller ones. The large, relatively inexpensive properties were almost all banked, meaning the property had a quick and significant rise from street grade to building. Many of them were also categorized as having 'severe' slope. By finding this apparent relationship between the slope/contour of the land and the sale price within the relationship of Lot Area to sale price, we continued with the construction of the model. After a first iteration yielded some values that were not statistically significant, we created a more robust second version.

Below is the output of the second iteration of our model. We determined that whether or not a property was 'Banked' was an incredibly significant predictor, As well as the interaction between LotArea and LandSlope. The output of running the model, with all predictors and interaction terms, is below. The linear model equation can be expressed as:

$$SalePrice = \beta_0 + \beta_1 \times GrLivArea + \beta_2 \times LotArea + \beta_3 \times Banked + \beta_4 \times (GrLivArea \times Moderate)$$

where: $\beta_0$ is the intercept term, $\beta_1$ is the coefficient for $GrLivArea$, $\beta_2$ is the coefficient for $LotArea$, $\beta_3$ is the coefficient for $Banked$, - $\beta_4$ is the coefficient for $GrLivArea \times Moderate\ Slope$, $\beta_5$ is the coefficient for $LotArea \times LandSlope$, $\beta_6$ is the coefficient for $GrLivArea \times Banked$, and $\beta_7$ is the coefficient for $LotArea \times Banked$.

Addressing the assumptions of this model, the first thing we can notice is that it is much better in meeting the normality of residuals assumption than the previous two models. Additionally, while it still is not perfect in this area, it has a higher degree of homoscedasticity, when visually comparing the residuals vs fitted values to the previous models. One thing to note, however, is the relatively high Cook's D value of some of these influential points. Some points have a Cook's D of approximately 0.4, which is much higher than the rest. We must again proceed with caution with this model.

After running all three linear regression models, we must find the Adjusted R^2, CV Press, and Kaggle scores of the models to see which one was the best. Our CV Press scores were calculated in SAS, and the code to find all fo these values will be in the appendix.

| Predictive Models | Adjusted R2 | CV PRESS | Kaggle Score |
|---|---|---|---|
| Simple Linear Regression | 0.2704971 | 153.891 | .33906 |
| Multiple Linear Regression | 0.5231282 | 4.4258e+12 | .28586 |
| Custom MLR Model | 0.5707977 | 4.0913e+12 | .28449 |

Above is the final table comparing the three models. The CV Press values for the first and third values seem exceptionally high. However, seeing as the Adjusted R^2 continues to increase as we go down the list, and the Kaggle score continues to decrease, our final recommendation is to use the third model, our formulated multiple linear regression model, in order to predict sale price in Ames. All in all, each model is not without its flaws, and more time must be spent to determine a truly ideal model, however, by carefully observing relationships, we can demonstrate that our ability to predict the sale price will only increase.

Appendix:

'''Max Pagan and Christian Castro

Analysis 1

ChatGPT was utilized for specific functions, error solving, and commenting for documentation'''

```
# dplyr for data manipulation
library(dplyr)
# ggplot2 for data visualization
library(ggplot2)
# car for diagnostic plots
library(car)
# boot for cross validation
library(boot)

# Load the dataset
data <- read.csv(choose.files())

# Focus on the three specified neighborhoods
filtered_data <- filter(data, Neighborhood %in% c("NAmes", "Edwards", "BrkSide"))

# Check for missing data in predictors and the response variable
sum(is.na(filtered_data$SalePrice))
sum(is.na(filtered_data$GrLivArea))
sum(is.na(filtered_data$Neighborhood))
filtered_data <- filtered_data %>%
  drop_na(SalePrice, GrLivArea, Neighborhood)
```

```
# Basic summary and structure
summary(filtered_data)
str(filtered_data)

# Plotting SalePrice against GrLivArea
ggplot(filtered_data, aes(x = GrLivArea, y = SalePrice)) +
  geom_point() +
  facet_wrap(~ Neighborhood) +
  labs(title = "SalePrice vs GrLivArea in Selected Neighborhoods",
          x = "Living Area (GrLivArea)", y = "Sale Price")

# SalePrice vs GrLivArea by Neighborhood
ggplot(filtered_data, aes(x = GrLivArea, y = SalePrice, color = Neighborhood)) +
  geom_point() +
  labs(title = "SalePrice vs GrLivArea in Selected Neighborhoods",
          x = "Living Area (GrLivArea)", y = "Sale Price") +
  theme_minimal()

# Linear regression model with interaction between GrLivArea and Neighborhood
model <- lm(SalePrice ~ GrLivArea * Neighborhood, data = filtered_data)

# Display the model summary
summary(model)

# Plotting diagnostic plots for the linear regression model
par(mfrow = c(2, 2)) # Setting up the plotting area for multiple plots
plot(model) # Base R diagnostic plots for linear models
```

```
# Generating the influence plot
influencePlot(model, main = "Influence Plot")

# Calculating Cook's distance
cooksD <- cooks.distance(model)

# Plotting Cook's distance
plot(cooksD, type = "h", main = "Cook's Distance", ylab = "Cook's distance")
abline(h = 4/(nrow(filtered_data)-length(coef(model))), col = "red")

# Fit your model
model <- lm(SalePrice ~ GrLivArea * Neighborhood, data = filtered_data)

# Calculate Cook's distance for each observation
cooks_d <- cooks.distance(model)

# Plot Cook's distance to identify potential outliers
plot(cooks_d, type="h", main="Cook's Distance", ylab="Cook's distance")
abline(h = 4 / length(cooks_d), col="red") # A common threshold is 4/n

# Calculate standardized residuals
std_residuals <- rstandard(model)

# Plot standardized residuals
plot(std_residuals, type="h", main="Standardized Residuals")
abline(h = c(-2, 2), col="red")
```

```
# Assuming you've decided that observations with Cook's distance > 4/n are outliers
threshold <- 4 / length(cooks_d)
outliers <- which(cooks_d > threshold)

# Remove outliers from the data
filtered_data_clean <- filtered_data[-outliers, ]

# Re-fit the model without outliers
model_clean <- lm(SalePrice ~ GrLivArea * Neighborhood, data = filtered_data_clean)

# Check the diagnostic plots for the new model
par(mfrow = c(2, 2))
plot(model_clean)

# Extracting and displaying model coefficients
coefficients <- coef(model)
conf_int <- confint(model)

# Display the coefficients and confidence intervals
print(coefficients)
print(conf_int)

# Fit the simpler model without interactions
simple_model <- lm(SalePrice ~ GrLivArea + Neighborhood, data = filtered_data)

# Fit the complex model with interactions (already done in your code)
complex_model <- model  # This is just for clarity, as you've already fitted this model
```

```
# Get Adjusted R-squared values
adj_r2_simple <- summary(simple_model)$adj.r.squared
adj_r2_complex <- summary(complex_model)$adj.r.squared

# Display Adjusted R-squared values
print(paste("Adjusted R-squared for simple model:", adj_r2_simple))
print(paste("Adjusted R-squared for complex model:", adj_r2_complex))

calc_cv_press <- function(model, data, folds = 10) {
  # Perform K-fold cross-validation and calculate PRESS
  cv_results <- cv.glm(data, model, K = folds)
  # cv.glm() returns NaN if there's an issue in prediction; handle this case
  if (any(is.nan(cv_results$delta))) {
        # Calculate PRESS manually if cv.glm() fails
        press <- sum((residuals(model) / (1 - hatvalues(model)))^2)
  } else {
        press <- sum(cv_results$delta)
  }
  return(press)
}

# Recalculate CV PRESS for each model
cv_press_simple <- calc_cv_press(simple_model, filtered_data)
cv_press_complex <- calc_cv_press(complex_model, filtered_data)

# Display CV PRESS values again
print(paste("CV PRESS for simple model:", cv_press_simple))
print(paste("CV PRESS for complex model:", cv_press_complex))
```


*Figure 1 - plotting the data*

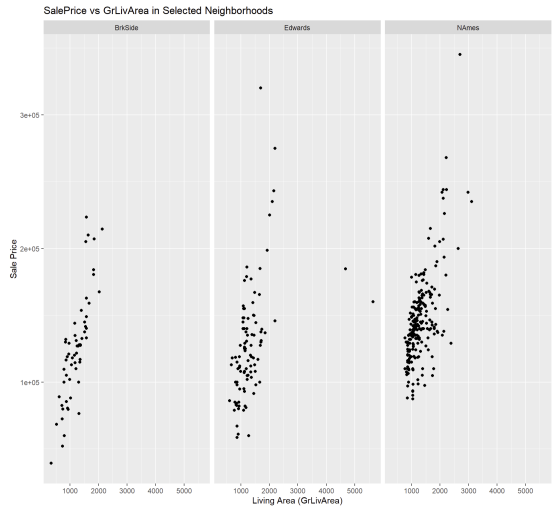


*Figure 2 - splitting data by neighborhood*

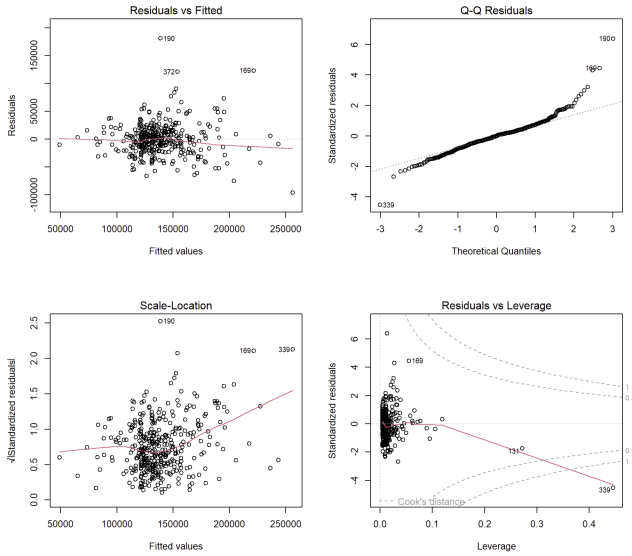


*Figure 3 - diagnostic graphs*

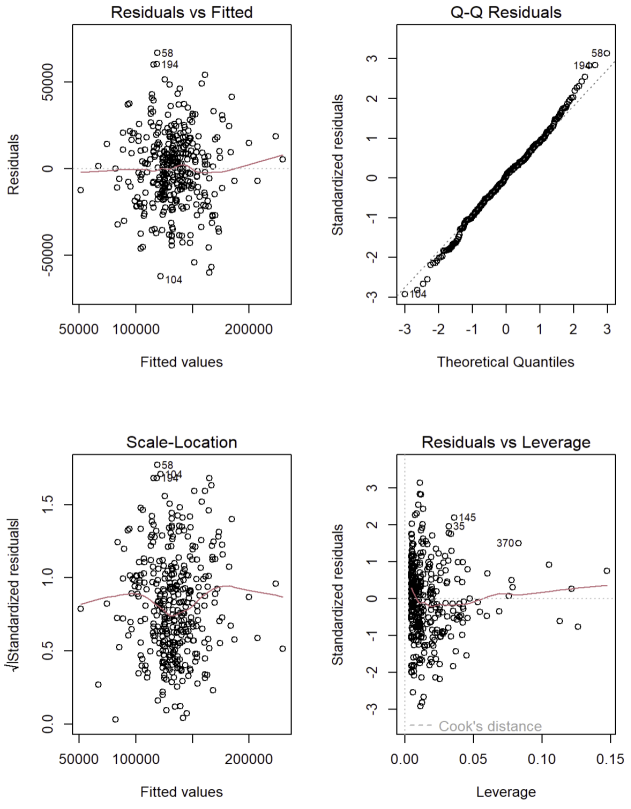


*Figure 4 - diagnostic graphs, post outlier handling*

```
train <- read.csv("~/Downloads/train.csv")
test <- read.csv("~/Downloads/test.csv")
library(ggplot2)

# Convert LandContour and LandSlope to factors
train$LandContour <- as.factor(train$LandContour)
train$LandSlope <- as.factor(train$LandSlope)
# Set reference levels for LandContour and LandSlope
train$LandContour <- relevel(train$LandContour, ref = "Lvl")
train$LandSlope <- relevel(train$LandSlope, ref = "Gtl")

#creating the simple linear model SalePrice vs Log(YearBuilt)
linear <- lm(SalePrice ~ log(YearBuilt), data = train)
linear

##
## Call:
## lm(formula = SalePrice ~ log(YearBuilt), data = train)
##
## Coefficients:
##       (Intercept)  log(YearBuilt)
##       -20195411       2685933

summary(linear)

##
## Call:
## lm(formula = SalePrice ~ log(YearBuilt), data = train)
##
## Residuals:
##       Min      1Q  Median      3Q       Max
## -143650  -40966  -15605   22501  542965
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -20195411       875247  -23.07   <2e-16 ***
## log(YearBuilt)   2685933     115372   23.28   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 67850 on 1458 degrees of freedom
## Multiple R-squared:  0.271,  Adjusted R-squared:  0.2705
## F-statistic:   542 on 1 and 1458 DF,  p-value: < 2.2e-16
```

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

#creating the given multiple regression model
given_model <- lm(SalePrice~GrLivArea + FullBath, data = train)
given_model

##
## Call:
## lm(formula = SalePrice ~ GrLivArea + FullBath, data = train)
##
## Coefficients:
## (Intercept)  GrLivArea        FullBath
##        3162.99           89.09   27311.09

summary(given_model)

##
## Call:
## lm(formula = SalePrice ~ GrLivArea + FullBath, data = train)
##
## Residuals:
##        Min      1Q  Median      3Q       Max
## -400438  -26191   -2027   21488  343260
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3162.993   4775.342   0.662    0.508
## GrLivArea    89.091            3.519  25.314  < 2e-16 ***
## FullBath     27311.090   3357.001   8.136  8.7e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 54860 on 1457 degrees of freedom
## Multiple R-squared:  0.5238, Adjusted R-squared:  0.5231
## F-statistic: 801.3 on 2 and 1457 DF,  p-value: < 2.2e-16

par(mfrow = c(2, 2))
plot(given_model)

#creating the first attempt of my model - GrLivArea, LotArea, LandContour, and LandSlope
my_model <- lm(SalePrice ~ GrLivArea + LotArea + LandSlope + LandSlope*GrLivArea +
LandSlope*LotArea + LandContour + LandContour*GrLivArea + LandContour*LotArea +
LandSlope*LandContour, data = train)
my_model
```

```
##
## Call:
## lm(formula = SalePrice ~ GrLivArea + LotArea + LandSlope + LandSlope *
##       GrLivArea + LandSlope * LotArea + LandContour + LandContour *
##       GrLivArea + LandContour * LotArea + LandSlope * LandContour,
##       data = train)
##
## Coefficients:
##                 (Intercept)                   GrLivArea
##                   -5.798e+03                   1.076e+02
##                     LotArea                LandSlopeMod
##                    2.514e+00                  -3.539e+04
##                 LandSlopeSev                LandContourBnk
##                    4.192e+04                   1.003e+05
##                 LandContourHLS                LandContourLow
##                    1.907e+04                  -3.524e+03
##       GrLivArea:LandSlopeMod       GrLivArea:LandSlopeSev
##                    3.285e+01                  -3.920e+01
##          LotArea:LandSlopeMod         LotArea:LandSlopeSev
##                   -2.766e+00                  -2.707e+00
##       GrLivArea:LandContourBnk     GrLivArea:LandContourHLS
##                   -7.794e+01                   8.817e+00
##       GrLivArea:LandContourLow     LotArea:LandContourBnk
##                   -2.587e+01                  -2.593e+00
##          LotArea:LandContourHLS       LotArea:LandContourLow
##                    9.315e-01                   1.045e+00
## LandSlopeMod:LandContourBnk  LandSlopeSev:LandContourBnk
##                    1.372e+04                   1.990e+05
## LandSlopeMod:LandContourHLS  LandSlopeSev:LandContourHLS
##                   -1.336e+04                  -8.760e+04
## LandSlopeMod:LandContourLow  LandSlopeSev:LandContourLow
##                    6.053e+04                   6.398e+04
```

```
summary(my_model)
```

```
##
## Call:
## lm(formula = SalePrice ~ GrLivArea + LotArea + LandSlope + LandSlope *
##       GrLivArea + LandSlope * LotArea + LandContour + LandContour *
##       GrLivArea + LandContour * LotArea + LandSlope * LandContour,
##       data = train)
##
## Residuals:
##       Min      1Q  Median      3Q     Max
```

```
## -189872  -28064    -225   22270  330592
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             -5.798e+03  4.829e+03  -1.201  0.23012
## GrLivArea                1.076e+02  3.164e+00  34.007  < 2e-16 ***
## LotArea                  2.514e+00  3.929e-01   6.400 2.10e-10 ***
## LandSlopeMod                -3.539e+04  2.776e+04  -1.275  0.20260
## LandSlopeSev                 4.192e+04  7.837e+04   0.535  0.59281
## LandContourBnk               1.003e+05  1.597e+04   6.278 4.54e-10 ***
## LandContourHLS               1.907e+04  2.734e+04   0.697  0.48564
## LandContourLow              -3.524e+03  3.904e+04  -0.090  0.92809
## GrLivArea:LandSlopeMod           3.285e+01  1.378e+01   2.384  0.01728 *
## GrLivArea:LandSlopeSev      -3.920e+01  5.582e+01  -0.702  0.48262
## LotArea:LandSlopeMod        -2.766e+00  1.059e+00  -2.611  0.00912 **
## LotArea:LandSlopeSev        -2.707e+00  1.041e+00  -2.600  0.00942 **
## GrLivArea:LandContourBnk         -7.794e+01  1.139e+01  -6.841 1.16e-11 ***
## GrLivArea:LandContourHLS         8.817e+00  1.671e+01   0.528  0.59777
## GrLivArea:LandContourLow         -2.587e+01  2.342e+01  -1.105  0.26951
## LotArea:LandContourBnk  -2.593e+00  1.016e+00  -2.552  0.01080 *
## LotArea:LandContourHLS           9.315e-01  1.117e+00   0.834  0.40430
## LotArea:LandContourLow           1.046e+00  1.054e+00   0.992  0.32153
## LandSlopeMod:LandContourBnk  1.372e+04  2.086e+04   0.658  0.51076
## LandSlopeSev:LandContourBnk  1.990e+05  1.020e+05   1.952  0.05116 .
## LandSlopeMod:LandContourHLS -1.336e+04  2.074e+04  -0.644  0.51965
## LandSlopeSev:LandContourHLS -8.760e+04  8.851e+04  -0.990  0.32246
## LandSlopeMod:LandContourLow  6.053e+04  2.638e+04   2.295  0.02190 *
## LandSlopeSev:LandContourLow  6.398e+04  6.228e+04   1.027  0.30450
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 51560 on 1436 degrees of freedom
## Multiple R-squared:  0.5854, Adjusted R-squared:  0.5788
## F-statistic: 88.16 on 23 and 1436 DF,  p-value: < 2.2e-16

par(mfrow = c(2, 2))
plot(my_model)

## Warning: not plotting observations with leverage one:
##   694, 1397

#editing my model down to only the significalnt variables
# Create the 'Banked' column
train$Banked <- ifelse(train$LandContour == "Bnk", 1, 0)
```

```
# Create the 'Low' column
train$Low <- ifelse(train$LandContour == "Low", 1, 0)

# Create the 'Moderate' column
train$Moderate <- ifelse(train$LandSlope == "Mod", 1, 0)


my_model_2 <- lm(SalePrice ~ GrLivArea + LotArea + Banked + GrLivArea:Moderate
+LotArea:LandSlope + GrLivArea:Banked + LotArea:Banked, data =train)
my_model_2

##
## Call:
## lm(formula = SalePrice ~ GrLivArea + LotArea + Banked + GrLivArea:Moderate +
##       LotArea:LandSlope + GrLivArea:Banked + LotArea:Banked, data = train)
##
## Coefficients:
##       (Intercept)              GrLivArea                LotArea
##       -5559.854                108.463                  2.440
##                Banked GrLivArea:Moderate  LotArea:LandSlopeMod
##       97961.266                16.388           -1.652
## LotArea:LandSlopeSev      GrLivArea:Banked      LotArea:Banked
##                -1.826           -84.619           -1.519

summary(my_model_2)

##
## Call:
## lm(formula = SalePrice ~ GrLivArea + LotArea + Banked + GrLivArea:Moderate +
##       LotArea:LandSlope + GrLivArea:Banked + LotArea:Banked, data = train)
##
## Residuals:
##       Min      1Q  Median      3Q     Max
## -191108  -28826    -816   22212  329282
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -5559.8544  4653.0269  -1.195  0.23233
## GrLivArea               108.4633     3.0820  35.193  < 2e-16 ***
## LotArea                   2.4404     0.3550   6.875 9.18e-12 ***
## Banked                97961.2656 14967.0199   6.545 8.22e-11 ***
## GrLivArea:Moderate       16.3881     6.2064   2.641  0.00837 **
## LotArea:LandSlopeMod     -1.6517     0.6107  -2.705  0.00692 **
## LotArea:LandSlopeSev     -1.8262     0.3564  -5.125 3.38e-07 ***
## GrLivArea:Banked         -84.6189    10.0158  -8.449  < 2e-16 ***
```

```
## LotArea:Banked                -1.5189          0.7526  -2.018  0.04375 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 52050 on 1451 degrees of freedom
## Multiple R-squared:  0.5732, Adjusted R-squared:  0.5708
## F-statistic: 243.5 on 8 and 1451 DF,  p-value: < 2.2e-16

par(mfrow = c(2, 2))
plot(my_model_2)

# Print Adjusted R^2
adjusted_r2_1 <- summary(linear)$adj.r.squared
cat("Adjusted R^2: SLR model - ", adjusted_r2_1, "\n")

## Adjusted R^2: SLR model -  0.2704971

adjusted_r2_2 <- summary(given_model)$adj.r.squared
cat("Adjusted R^2: provided Multiple Regression model - ", adjusted_r2_2, "\n")

## Adjusted R^2: provided Multiple Regression model -  0.5231282

adjusted_r2_3 <- summary(my_model_2)$adj.r.squared
cat("Adjusted R^2: My multiple regression model - ", adjusted_r2_3, "\n")

## Adjusted R^2: My multiple regression model -  0.5707977

#importing the CV press values from SAS:
# Internal CV Press
cv_press1 <- 153.89090
cat("Internal CV PRESS - SLR:", cv_press1, "\n")

## Internal CV PRESS - SLR: 153.8909

cv_press2 <- 4.425785e12
cat("Internal CV PRESS - provided multiple model:", cv_press2, "\n")

## Internal CV PRESS - provided multiple model: 4.425785e+12

cv_press3 <-    4.091315e12
cat("Internal CV PRESS - our model:", cv_press3, "\n")

## Internal CV PRESS - our model: 4.091315e+12

#In this code we are predicting SalePrice for test and creating a dataframe for it:

#First, we will use the SLR
# 1. Create an empty column in test called SalePrice (and any other necessary columns)
```

```
test$SalePrice <- NA

# 2. Populate the SalePrice column with the predicted values from the model
test$SalePrice <- predict(linear, newdata = test)

# 3. Output a new dataframe with SalePrice and Id
output_SLR <- data.frame(Id = test$Id, SalePrice = test$SalePrice)

#Next, we will use the given MLR
# 1. Create an empty column in test called SalePrice (and any other necessary columns)
test$SalePrice <- NA

# 2. Populate the SalePrice column with the predicted values from the model
test$SalePrice <- predict(given_model, newdata = test)

# 3. Output a new dataframe with SalePrice and Id
output_given <- data.frame(Id = test$Id, SalePrice = test$SalePrice)

#Finally, the model we came up with
# 1. Create an empty column in test called SalePrice (and any other necessary columns)
test$SalePrice <- NA
# Create the 'Banked' column
test$Banked <- ifelse(test$LandContour == "Bnk", 1, 0)

# Create the 'Low' column
test$Low <- ifelse(test$LandContour == "Low", 1, 0)

# Create the 'Moderate' column
test$Moderate <- ifelse(test$LandSlope == "Mod", 1, 0)

# 2. Populate the SalePrice column with the predicted values from the model
test$SalePrice <- predict(my_model_2, newdata = test)

# 3. Output a new dataframe with SalePrice and Id
output_my_model<- data.frame(Id = test$Id, SalePrice = test$SalePrice)

#outputting the data:
write.csv(output_SLR, file = "predicted_saleprice_SLR.csv", row.names = FALSE)
write.csv(output_given, file = "predicted_saleprice_given.csv", row.names = FALSE)
write.csv(output_my_model, file = "predicted_saleprice_our_model.csv", row.names = FALSE)

Kaggle_SLR <- 0.33906
cat("Kaggle Score - SLR:", Kaggle_SLR, "\n")
```

```
## Kaggle Score - SLR: 0.33906

Kaggle_given <- 0.28586
cat("Kaggle Score - given multiple regression:", Kaggle_SLR, "\n")

## Kaggle Score - given multiple regression: 0.33906

Kaggle_ours <- 0.28449
cat("Kaggle Score - our model:", Kaggle_SLR, "\n")

## Kaggle Score - our model: 0.33906

# Install and load necessary packages
#install.packages("knitr")
#install.packages("kableExtra")
#library(knitr)
#library(kableExtra)

# Assuming you have the necessary numbers in variables
#model_names <- c("Simple Linear Regression", "Multiple Linear Regression", "Our MLR Model")
#adjusted_r2 <- c(adjusted_r2_1, adjusted_r2_2, adjusted_r2_3)
#cv_press <- c(cv_press1, cv_press2, cv_press3)
#kaggle_score <- c(Kaggle_SLR, Kaggle_given, Kaggle_ours)

# Create a data frame
#results_df <- data.frame(Model = model_names, `Adjusted R2` = adjusted_r2, `CV PRESS` =
cv_press, `Kaggle Score` = kaggle_score)

# Print the table
#kable(results_df, format = "html") %>%
#  kable_styling()

train <- read.csv("~/Downloads/train.csv")
test <- read.csv("~/Downloads/test.csv")
library(ggplot2)

# Convert LandContour and LandSlope to factors
train$LandContour <- as.factor(train$LandContour)
train$LandSlope <- as.factor(train$LandSlope)
# Set reference levels for LandContour and LandSlope
train$LandContour <- relevel(train$LandContour, ref = "Lvl")
train$LandSlope <- relevel(train$LandSlope, ref = "Gtl")

ggplot(train, aes(x = LotArea, y = SalePrice, color = LandSlope)) +
  geom_point() +
  labs(title = "SalePrice vs lotArea by LandSlope",
```

```
        x = "LotArea",
        y = "SalePrice")

ggplot(train, aes(x = GrLivArea, y = SalePrice, color = LandSlope)) +
  geom_point() +
  labs(title = "SalePrice vs GrLivArea by LandSlope",
        x = "GrLivArea",
        y = "SalePrice")

ggplot(train, aes(x = LotArea, y = SalePrice, color = LandContour)) +
  geom_point() +
  labs(title = "SalePrice vs lotArea by LandContour",
        x = "LotArea",
        y = "SalePrice")

ggplot(train, aes(x = GrLivArea, y = SalePrice, color = LandContour)) +
  geom_point() +
  labs(title = "SalePrice vs GrLivArea by LandContour",
        x = "GrLivArea",
        y = "SalePrice")
```

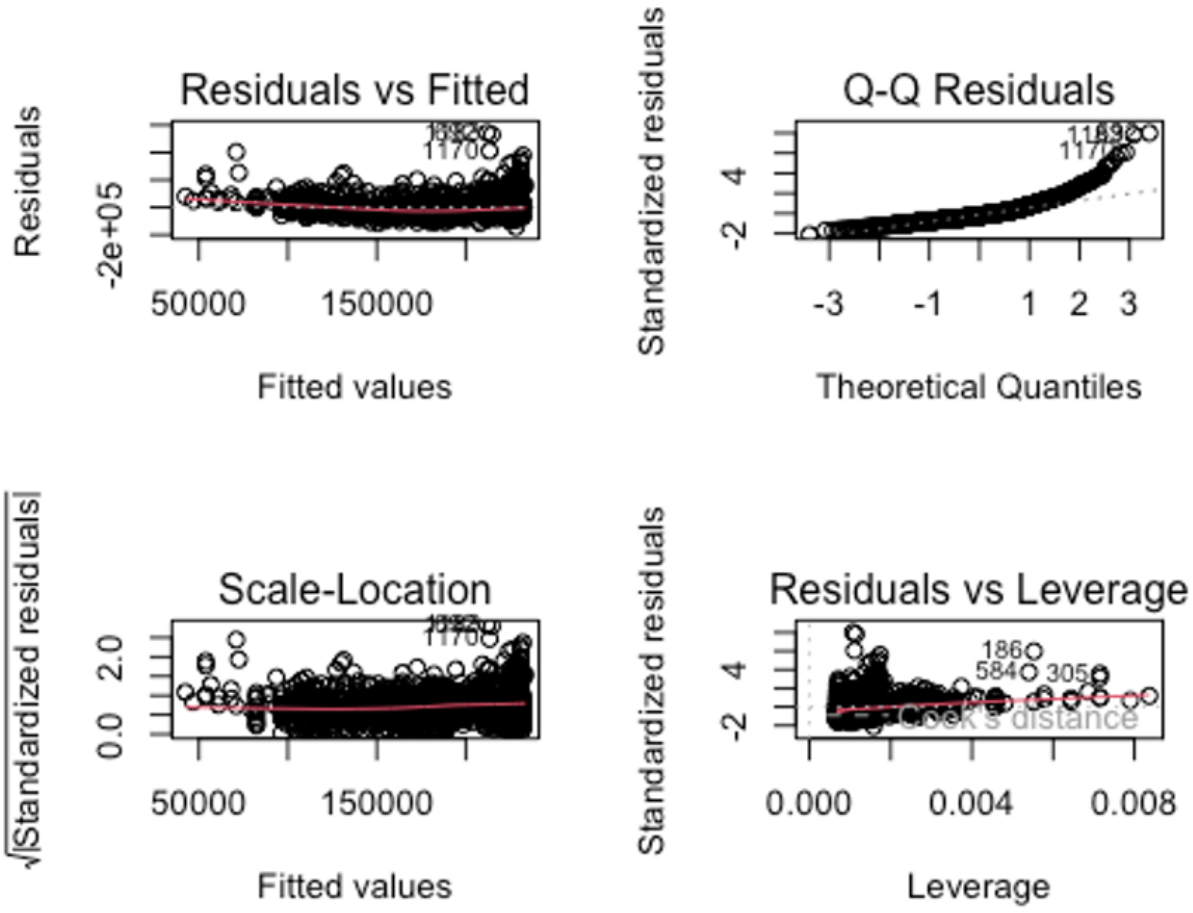


*Figure 5 - diagnostic graphs, SLR Analysis 2*

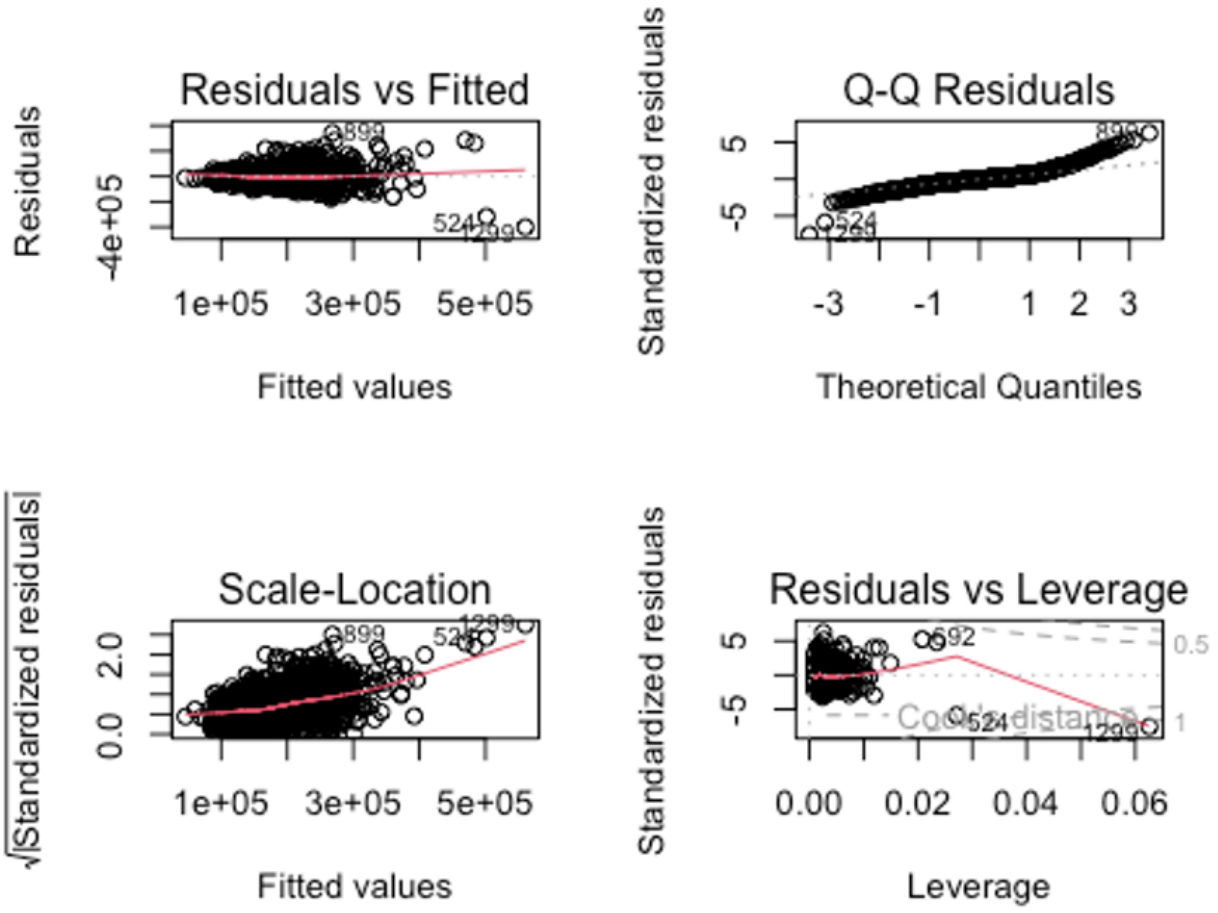


*Figure 6 - diagnostic graphs, provided MLR Analysis 2*

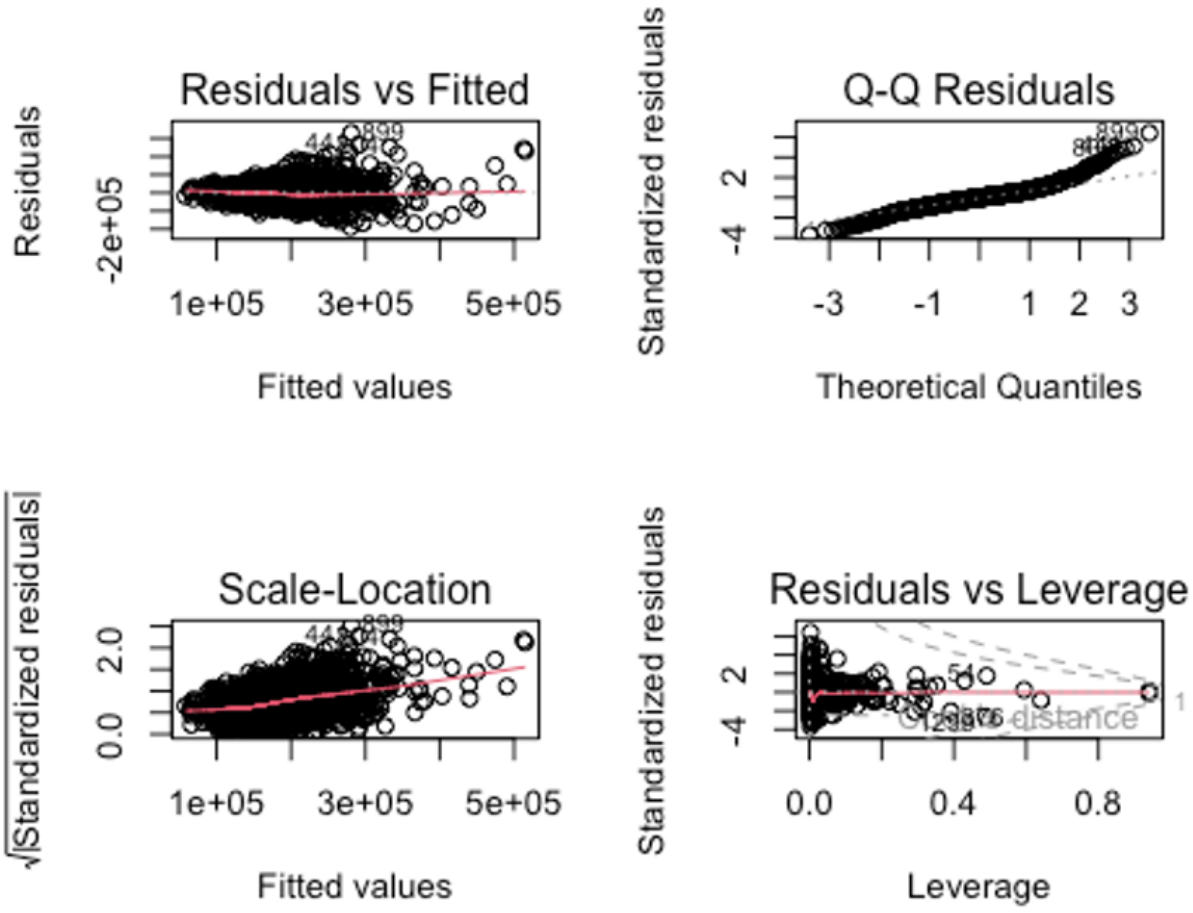


*Figure 7 - diagnostic graphs, Our MLR Analysis 2 attempt 1*

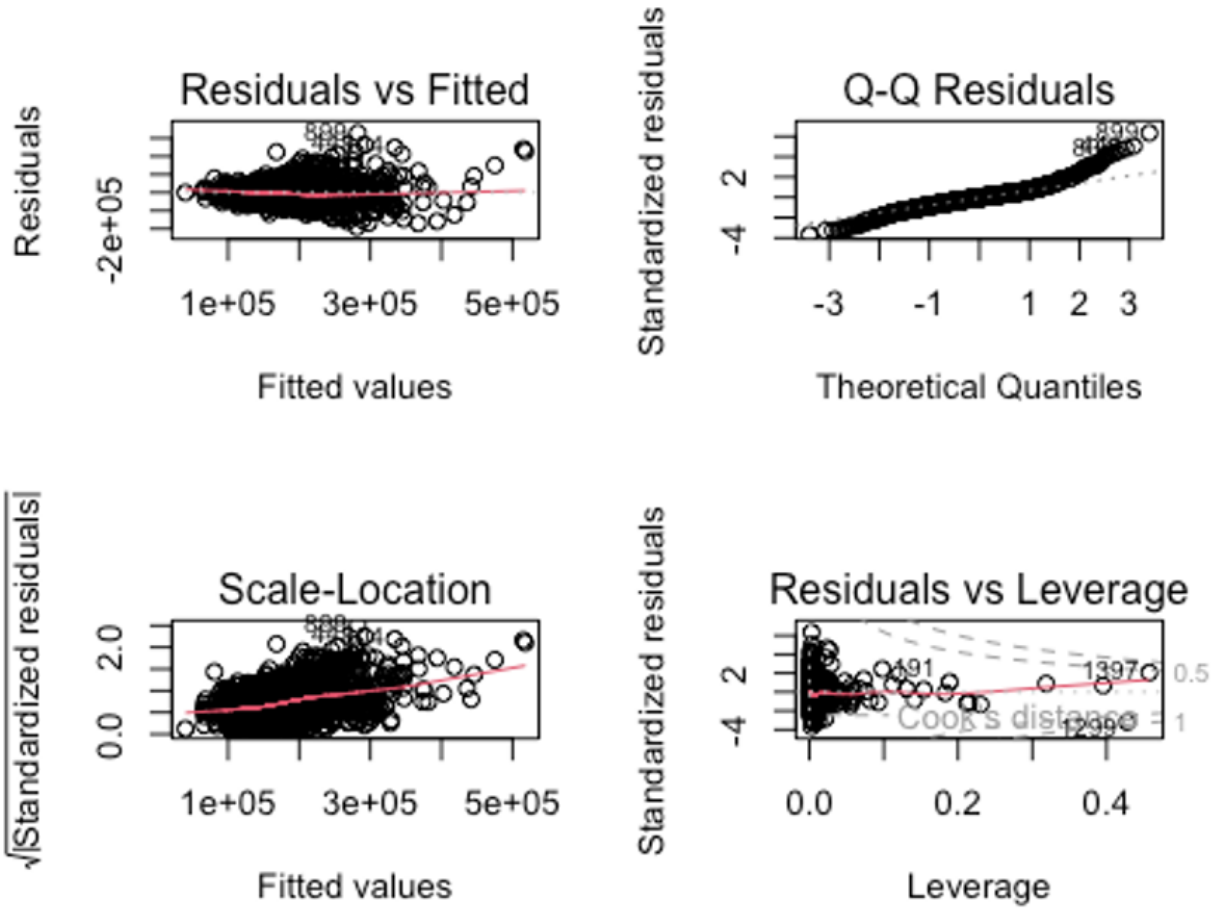


*Figure 8 - diagnostic graphs, Our MLR Analysis 2 attempt 2*

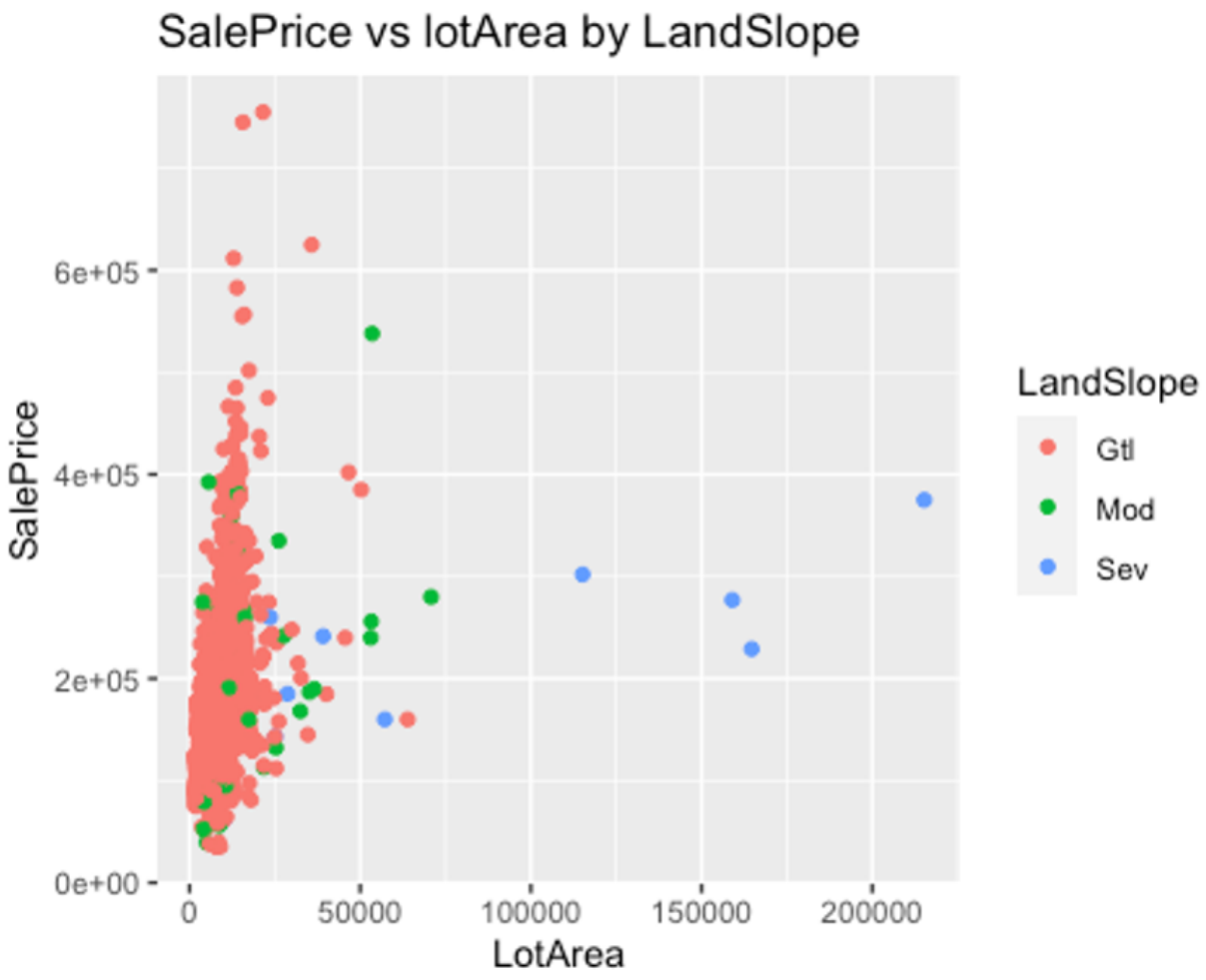


*Figure 9 - SalePrice vs LotArea, Color = LandSlope*

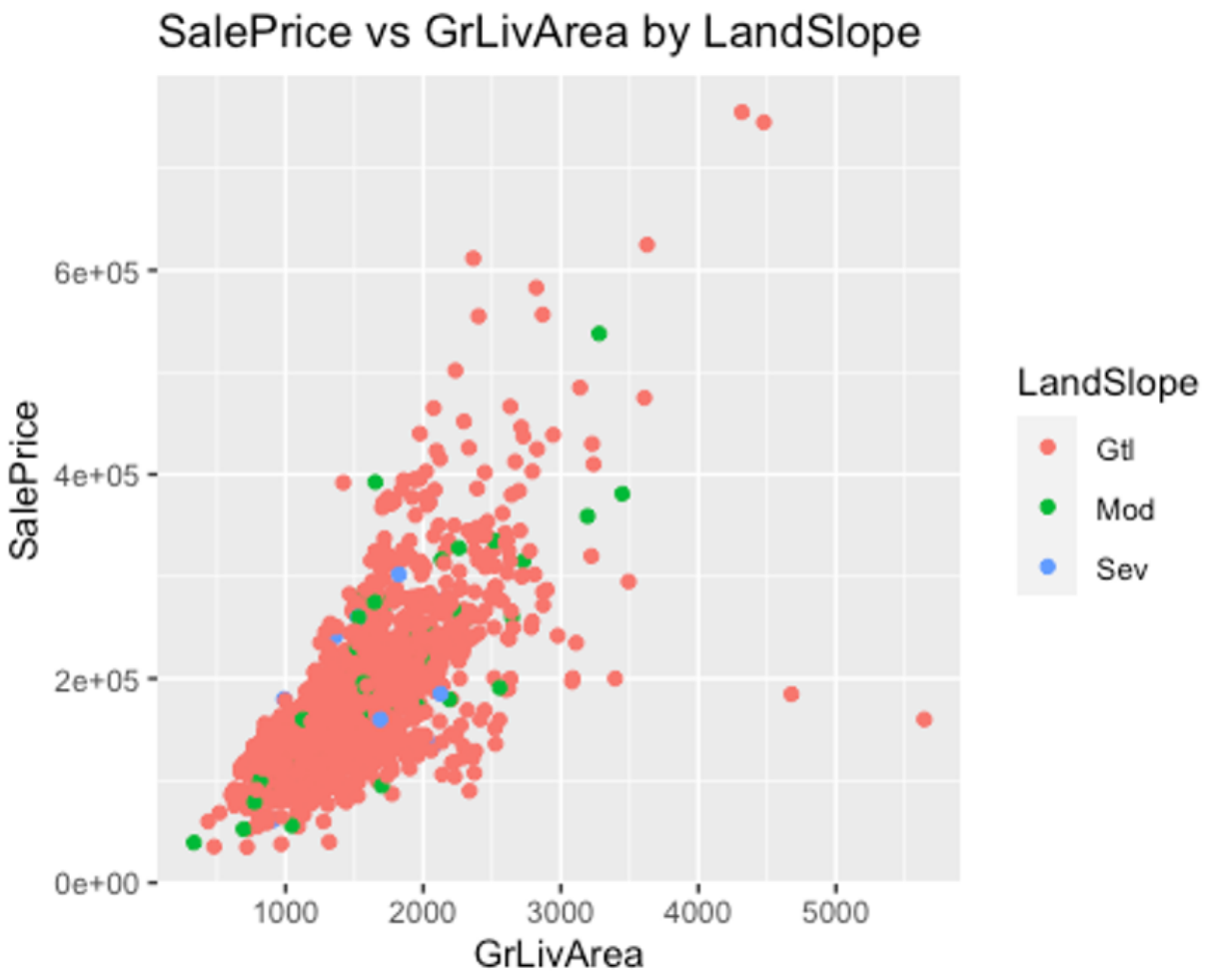


*Figure 10 - SalePrice vs GrLivArea, Color = LandSlope*

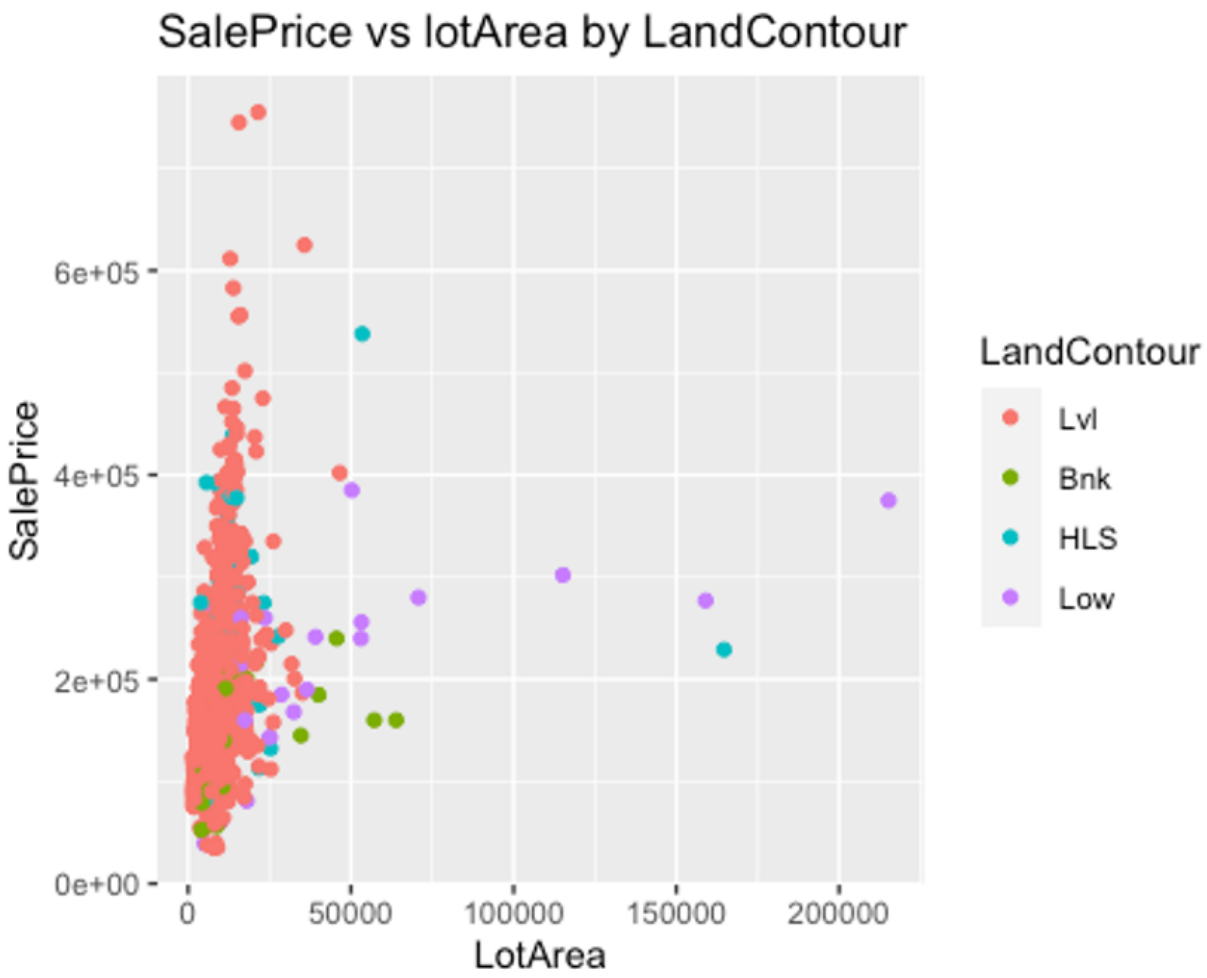


*Figure 11 - SalePrice vs LotArea, Color = LandContour*

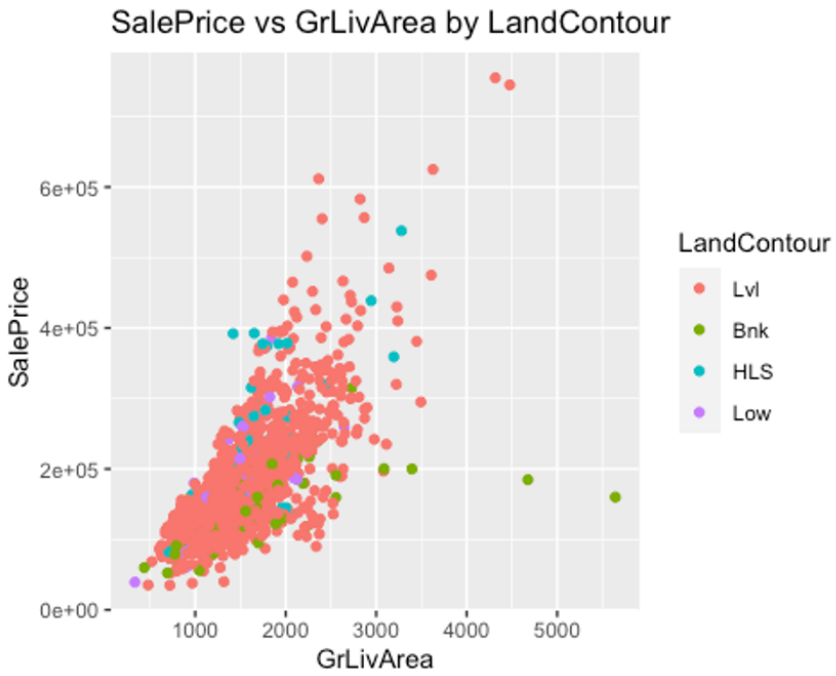


*Figure 12 - SalePrice vs GrLivArea, Color = LandContour*

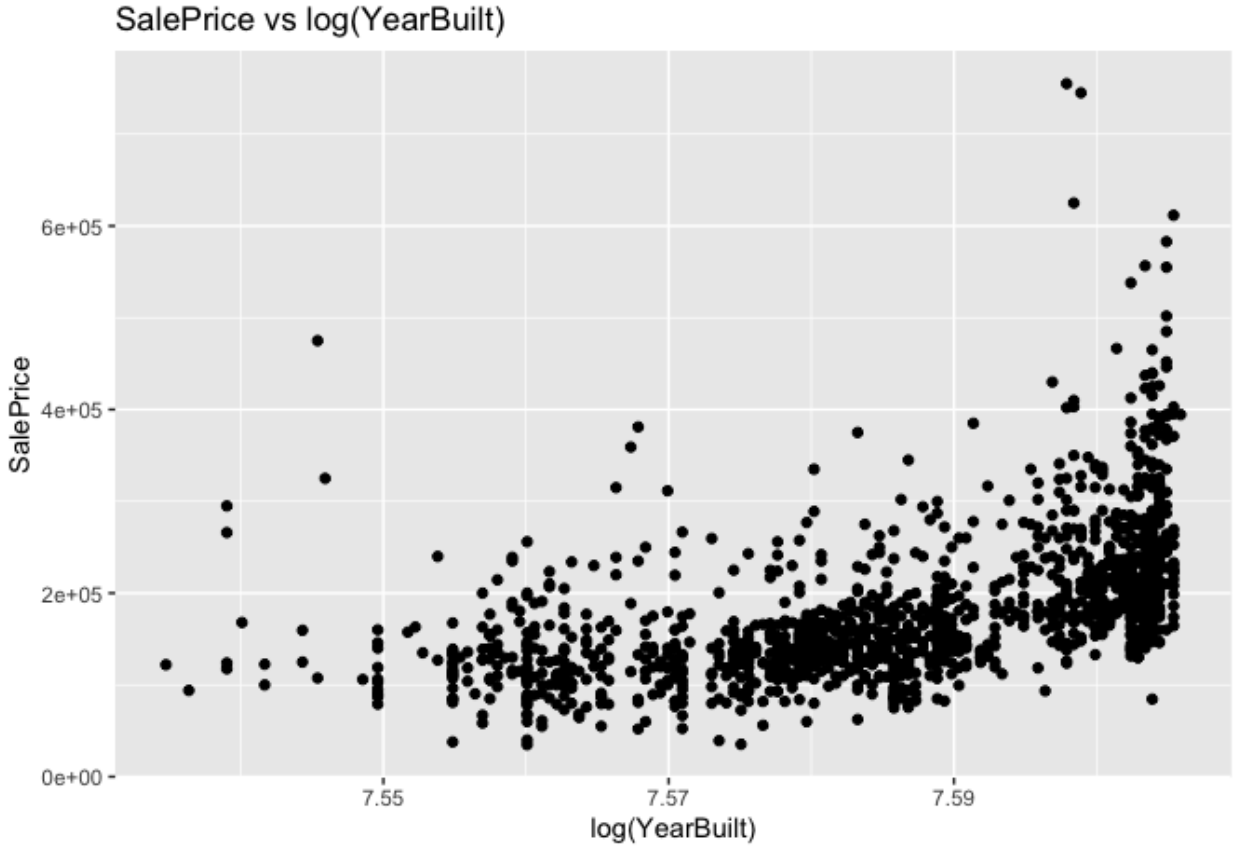

*Figure 13 - SalePrice vs log(YearBuilt)*

SAS CODE to find CV PRESS

```
/* Load the data */
proc import datafile='/home/u63115740/train.csv' out=train dbms=csv replace;
 getnames=yes;
run;

data train;
 set train;

 /* Add log(SalePrice) column */
 log_SalePrice = log(SalePrice);

 /* Add log(YearBuilt) column */
 log_YearBuilt = log(YearBuilt);
run;

proc glmselect data=train;
  model SalePrice = log_YearBuilt / selection=Forward(stop=CV) cvmethod = random(5) stats =
adjrsq;
run;

proc glmselect data=train;
  class FullBath; /* If FullBath is a categorical variable, specify it as a class variable */
```

```
  model SalePrice = GrLivArea FullBath / selection=Forward(stop=CV) cvmethod = random(5) stats
= adjrsq;
run;


data train;
 set train;


 /* Create the 'Banked' column */
 Banked = ifn(LandContour = "Bnk", 1, 0);


 /* Create the 'Low' column */
 Low = ifn(LandContour = "Low", 1, 0);


 /* Create the 'Moderate' column */
 Moderate = ifn(LandSlope = "Mod", 1, 0);
run;


proc glmselect data=train;
class LandSlope;
class LandContour;
  model SalePrice = GrLivArea LotArea Banked GrLivArea*Moderate
            LotArea*LandSlope GrLivArea*Banked LotArea*Banked / selection=Forward(stop=CV)
cvmethod = random(5) stats = adjrsq;
run;
```