Introduction

Our analysis provides an overview of the real estate market within Ames Iowa. Our analysis has two sections regarding the association between sales price and livable area within the target market of Century 21 Ames and can be visualized with an R Shiny dashboard at your convenience. The second section discusses the general market of Ames, Iowa, and provides four multiple linear regression models to better understand what factors influence sales prices.

Data Description

This data set was retrieved from Kaggle[1], features 80 columns and 1460 lines of data in the training set, and an additional 1459 lines in the test set. Descriptions of the variables as well as the fields we used can be accessed via the GitHub link for the project[2].

Analysis Question 1:

Restatement of Problem

Century 21 Ames has commissioned a project to better understand the relationship between the sales prices of homes and the livable square footage in their target real estate market of Northern Ames (NAmes), Edwards, and Brookside (BrkSide) neighborhoods.

Build and Fit the Model

Median{logSalesPrice|logSqFt, Neighborhood} = 8.007 + 0.520*logSqFt + (0.486*NAmes) – (2.094 * BrkSide) – (0.047 * NAmes) + (0.300 * BrkSide)

We have insufficient evidence to conclude there is a significant difference in estimated median price per square foot between Edwards and Northern Ames neighborhoods (p-value = .5203).

Brookside Model:
Median{logSalesPrice|logSqFt, Neighborhood = BrkSide} = 8.007-2.094 + (.520+.300)*logSqFt

Edwards and NAmes Model:
Median{logSalesPrice|logSqFt, Neighborhood = Edwards or NAmes} = 8.007 + 0.520*logSqFt

Checking Assumptions

Residual Plots

[1] https://www.kaggle.com/c/house-prices-advanced-regression-techniques/overview

[2] https://github.com/tskunz/MSDS6371_House_Regression_Project/blob/main/house-prices-advanced-regression-techniques/data_description.txt

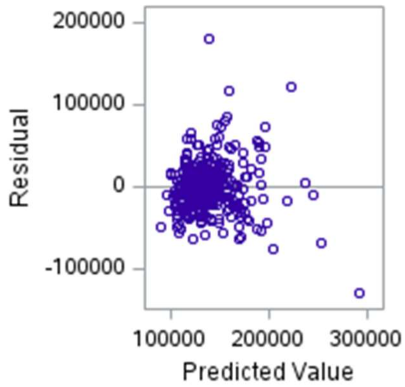


Figure 1 – Original Data

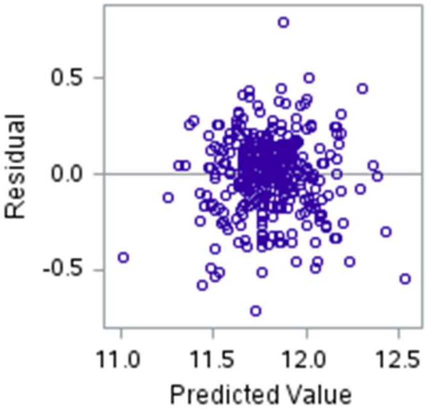
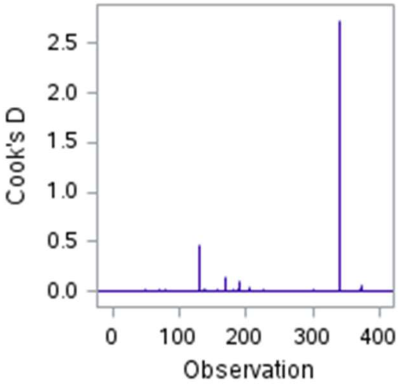


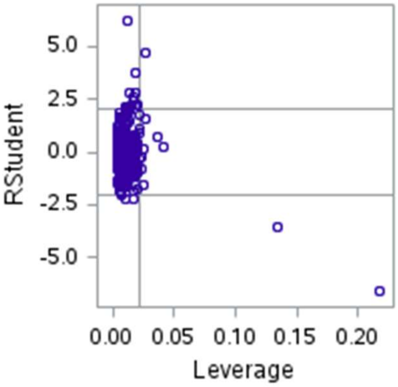
Figure 2 – Log transformed Data

Prior to the log transformation, we see a strong cluster, with outliers. After transforming the data, we still see a strong cluster, but the residuals visually appear to be more normally distributed. The cluster can be explained by the histogram of the residual points which shows most observations cluster around the median.

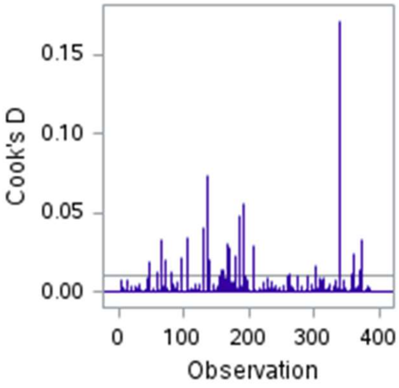
Influential point analysis (Cook's D and Leverage)

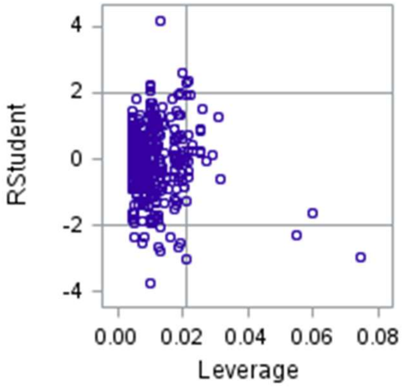
Original Data:





Transformed Data:

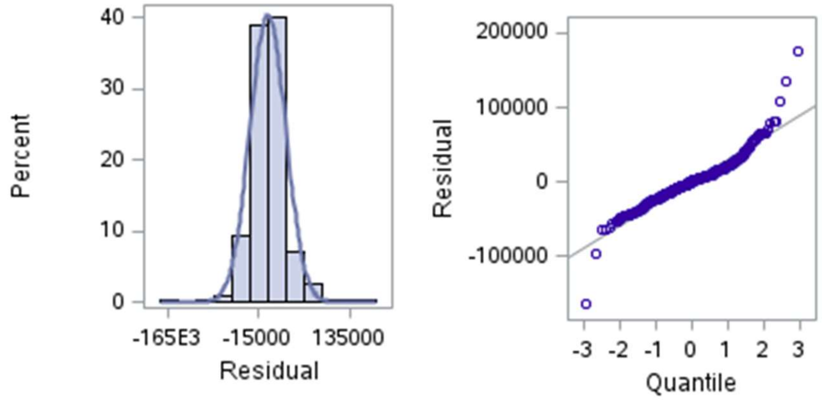




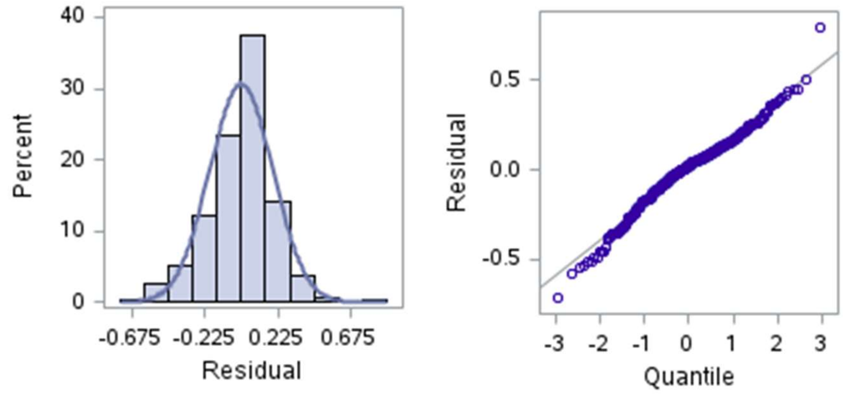
Prior to the transformation, we have several points where which have a much higher Cook's D. These points will have a much higher influence on the plots relative models than even the highest point in the log transformed data. After the transformation, we still have a few high residual – high leverage points, but will proceed with caution in this analysis.

The high leverage points were homes in Edwards Neighborhood which were sold as "partially completed"; therefore, the cost of the purchase is likely below what the expected sales price would be had it been completed. Further analysis and exploration is recommended for homes sold outside of normal conditions.

Normality:
Original Data:



Log Transformed Data:



Prior to the transformation, most of the data is concentrated around the median with some heavy outliers. Following the log transformation, the data more closely approximates a normal distribution. Additionally, the data set is sufficiently large for the Central Limit Theorem to apply to satisfy the assumption for normality.

Linear Trend and Standard Deviations:
Based on the Q-Q plots, the log transformed data has more evenly distributed data with less clustering, and fewer high leveraged outliers. The data also visually appears to have a linear trend.

Independence:
Real estate sales and appraisal prices are based on comparisons of similar properties in the similar geographical area, shared community amenities (parks, schools, shopping malls, etc.), and many in the area may be built by the same builder. We will be mindful that the industry may have underlying independence concerns while proceeding with caution in this analysis.

Analysis of the Model:

ß0: The intercept in this model provides an estimate of for the cost $ 3001.90 or log(8.007) of 0 square feet of livable area at the reference neighborhood (Edwards, as well as NAmes as there is not enough evidence to suggest they do not follow the same model) . This point is extrapolation as there were no recorded sales of undeveloped land with 0 square feet of livable area. Further research would be needed to determine the cost to purchase undeveloped land, and the intercept would not be an appropriate approximation.

ß1: The adjustment to the intercept for the Brookside neighborhood. The intercept for Brookside neighborhood would be $369.81, or an average cost of log(-2.094) less than that of the reference model.

ß2: The slope of the reference model provides the associated change (log(0.520)) in sales price for each incremental a one unit increase to the livable square feet. The increase for 100 square feet would be associated with $168.20 in the reference (Edwards) neighborhood.

ß3: The adjustment to the slope for Brookside neighborhood (log(-0.047)). The increase of 100 square feet in the Brookside neighborhood would be associated with a $160.50 increase to sales price.

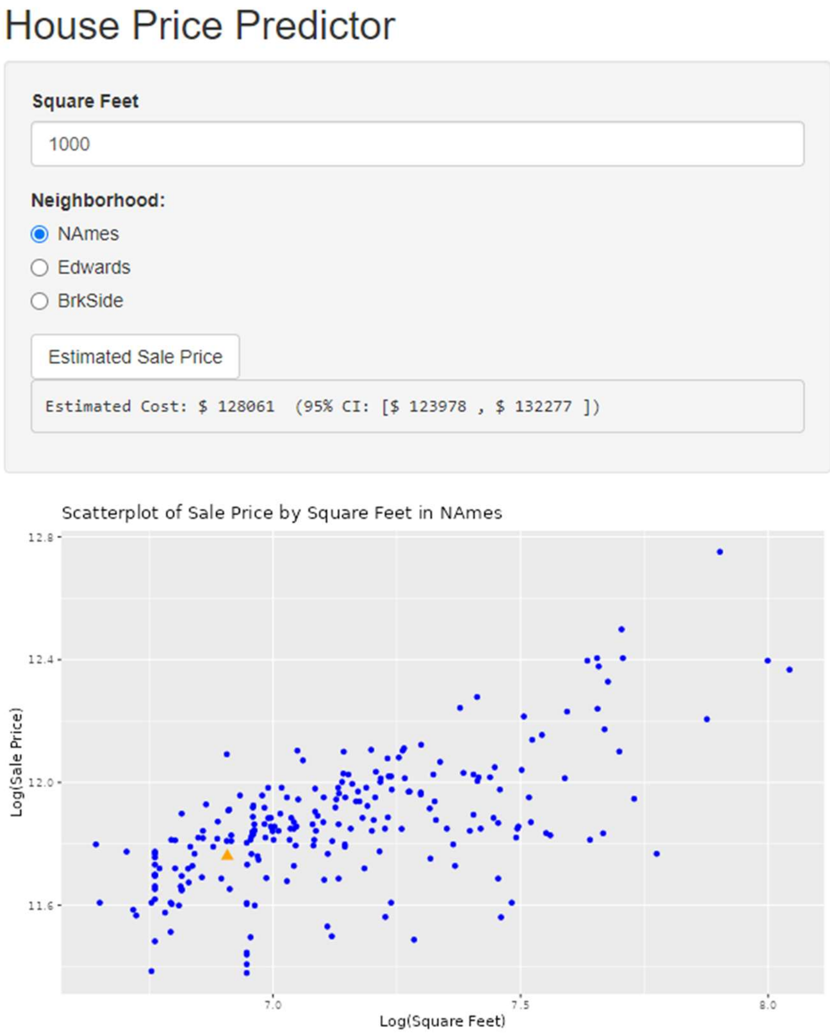
Conclusion

A doubling of Livable Square Feet in Brookside is associated with a 76.5% (2^.82; 95% Confidence Interval: [55.95%, 99.85%]) multiplicative increase in the estimated median Sales Price. A doubling of Livable Square Feet in Northern Ames or Edwards Neighborhoods is associated with a 43.4% (2^0.52; 95% Confidence Interval: [26.7%, 62.3%]) multiplicative increase in the estimated median Sales Price.

R Shiny: Price v. Living Area Chart

We created an R Shiny Dashboard for your agents to share with their clients to help them better understand the median sale price within a 95% confidence interval for a given total of livable space. This will assist the agents and clients better understand the market value for each of the three neighborhoods.

The app can be accessed via the following link: https://tkunz.shinyapps.io/House-Price-Predictor/ and the only two inputs needed are the estimated square feet of the livable area and the neighborhood of interest. After inputting these inputs and pressing the button, the model will estimate the median selling price for the given neighborhood and square footage. The estimated point will also be graphed as an orange triangle, to represent the point visually.



House Price Predictor

Square Feet

1000

Neighborhood:

● NAmes
○ Edwards
○ BrkSide

```
Estimated Cost: $ 128061  (95% CI: [$ 123978 , $ 132277 ])
```

Analysis Question 2

Restatement of Problem

The purpose of the second analysis was to create various prediction models to estimate Sales Prices for homes in Ames, Iowa. Specifically, we determined the best single predictor, as well as various other multi-linear regression models (MLRs) to best predict Sales Prices.

Candidate Models:

Before creating any of the models, it was decided that a healthy approach to the large dataset would begin with wrangling the provided data. After an initial analysis of the data, the following major characteristics were observed:

- “Normal” Sale Conditions accounted for 1,198 of 1,460 records (82.1%) in the training set, and 1,204 of 1,459 records (82.5%) in the test set,
- Many columns (E.g., “ExterQual”) represented categorical data that could be readily converted to numerical data according to a predetermined scale (1 = Excellent, 5 = Poor),
- One column, “CentralAir”, contained Boolean data that could be readily converted to numerical 0/1 data.

Based on these observations, this exercise primarily focused on the homes being sold under “Normal” Sale Conditions, following the domain assumption that abnormal sales could negatively affect the performance of any models due to outlying conditions. Additionally, each of the columns with readily convertible data were converted to numerical data for usage by the model.

SLR

$$\log{(SalePrice)} = \beta_0 + \ \beta_1 OverallQual$$[3]

Using a Forward and Backward Stepwise Regression (FBSR), the single greatest predictor (Adj. R^2 = 0.63) of Sale Price was determined to be the “Overall Qual[ity]” of the property. This fact was consistent with initial expectations. However, the fact that this column only contained ten levels did present an opportunity for an overly broad approach. This concern was addressed and accepted as the team understood MLR models would help establish a more narrow fit for the data.

MLR 1

$$\log{(SalePrice)} = \beta_0 + \beta_1 GrLivArea + \beta_2 FullBath$$

For the primary MLR model, the Above-Ground Living Area and Full Bathroom count were selected due to their strength as generally assumed predictors.

MLR 2

$$(Too\ long\ to\ display, see\ "Appendix - MLR2")$$

The second MLR model was chosen with minimal researcher bias, and was instead determined by selecting for the highest Adjusted R^2 score after a FBSR on all predictors, allowing for binary interactions. This model produced an excessive number of parameters; however, as demonstrated further in the comparison section, performed the best on both the training and testing datasets.

MLR 3

$$\begin{aligned}\log{(SalePrice)} \\ &= \beta_0 + \beta_1 OverallQual + \beta_2 GrLivArea + \beta_3 (OverallQual * GrLivArea) \\ &+ \beta_4 (OverallQual * BsmtFinSF1) + \beta_5 (GrLivArea * BsmtFinSF1) \\ &+ \beta_6 (OverallQual * TotalBsmtSF) + \beta_7 (GrLivArea * TotalBsmtSF) \\ &+ \beta_8 YearRemodAdd + \beta_9 (GrLivArea * YearRemodAdd)\end{aligned}$$

[3] For increased readability, all parameter estimates are included in the Appendix

Finally, after assessing the performance of MLR-2, a more parsimonious model was generated by a FBSR on the five strongest individual predictor, allowing for binary interactions. Although this model produced a promising Adjusted R^2 score, it still did not perform as well as MLR-2.

Checking Assumptions

Residual Plots

(See Appendix Residual Plots)
Although the right tail of the distribution begins to follow a more exponential trend, the vast majority of the data falls neatly within a linear model.

Influential point analysis (Cook's D and Leverage)

(See Appendix Cook's D and Leverage)
Before adjusting the data, there were a few data points that had high leverage and were strong outliers. These were removed in effort to reduce both the CV PRESS and Adjusted R^2 scores in addition to the log transformation performed in AQ1, however, this resulted in little effect on model performance.

Comparing Competing Models

PREDICTIVE MODELS	ADJUSTED R2	CV PRESS	KAGGLE SCORE
SIMPLE LINEAR REGRESSION	0.6518	55.25	0.47774
MLR-1	0.5508	71.75	0.28490
MLR-2	0.9051	11.34	0.13348
MLR-3	0.8876	6.36e11	0.17277

Conclusion: A short summary of the analysis.

Based on the above results, the best model for predictive analysis used 38 interaction parameters. This model had the best results on all three observed metrics, despite lacking any parsimony. Despite this model performing the best on training and set data, this team would recommend the third model which controlled for number of parameters, as this model is both easier to explain, and less likely to be an over-fit for the existing data.

Appendix

GitHub:

https://dhlaurel.github.io/ames.pdf
https://tskunz.github.io/

Analysis 1 SAS Code:

```
/* Import training data */

proc import out = work.train
   datafile= "/home/u63538552/sasuser.v94/train.csv"
   dbms = csv replace;
   getnames=yes;
   datarow=2;
   run;

/* Create a new data set to filter to the 3 Neighborhoods of interest "NAmes",
"Edwards", or "BrkSide".
  Perform log transformation of the sale price and living area. */

data house;
   set work.train;
   where Neighborhood in ("NAmes", "Edwards", "BrkSide"); /* Filter observations
based on neighborhoods of interest */
   logSalePrice = log(SalePrice);
   logGrLivArea = log(GrLivArea);
run;

/* Create a scatterplot of SalePrice against GrLivArea */

proc sgplot data=house;
   title 'Scatterplot of SalePrice and GrLivArea';
   scatter x=GrLivArea y=SalePrice;
   xaxis label='GrLivArea';
   yaxis label='SalePrice';
run;

/* Create a scatterplot of logSalePrice against logGrLivArea */

proc sgplot data=house;
   title 'Scatterplot of LogSalePrice and logGrLivArea';
   scatter x=logGrLivArea y=logSalePrice;
   xaxis label='logGrLivArea';
   yaxis label='logSalePrice';
```

```
run;

/* Use proc glm to generate linear regression analysis data for the untransformed data
using a common slope.*/

proc glm data=house plots=all;
   class Neighborhood;
   model SalePrice = GrLivArea Neighborhood / solution;
run;

/* Use proc glm to generate linear regression analysis data for the untransformed data
with unique slopes.*/
proc glm data=house plots=all;
   class Neighborhood;
   model SalePrice = GrLivArea*Neighborhood / solution;
run;

/* Use proc glm to generate linear regression analysis data for the transformed data
with equal slopes.*/
proc glm data=house plots=all;
   class Neighborhood; /* Specifies Neighborhood as a categorical variable */
   model logSalePrice = logGrLivArea Neighborhood / solution; /* Specifies the model */
run;

/* Use proc glm to generate linear regression analysis data for the transformed data
with unique slopes.*/
proc glm data=house plots=all;
   class Neighborhood; /* Specifies Neighborhood as a categorical variable */
   model logSalePrice = logGrLivArea*Neighborhood / solution; /* Specifies the model
with interaction */
run;
```

RShiny App Code:

UI:

```
# ui.R
 #load libraries
library(shiny)
library(ggplot2)

#set up the ui for the app
ui = fluidPage(
  titlePanel("House Price Predictor"), #app title
  sidebarLayout( #set up the side bar
  sidebarPanel(
   numericInput("sqft", "Square Feet", value = 0), # free numeric input box
   radioButtons("neighborhood", "Neighborhood:", #radio button to toggle between
neighborhoods
          choices = c("NAmes", "Edwards", "BrkSide"),
          selected = "NAmes"),  # Initial selected value
   actionButton("predict_button", "Estimated Sale Price"), # button to run the code to
display the output
   verbatimTextOutput("prediction_output") # output the results from the server code
  ),
  mainPanel(
   plotOutput("regression_plot") #plot the scatterplot from the server code
  )
 )
)
```

Server:

```
    # Load in libraries
    library(shiny)
    library(tidyverse)

    # Load in data set
    house =
    read.csv("https://raw.githubusercontent.com/tskunz/MSDS6371_House_Regression_Pr
    oject/main/house-prices-advanced-regression-techniques/train.csv")

    # filter data and log transform
    C21Ames = house %>% filter(Neighborhood %in% c("NAmes", "Edwards", "BrkSide"))
    C21Ames$logPrice = log(C21Ames$SalePrice)
    C21Ames$logSqFt = log(C21Ames$GrLivArea)

    # Different Slope Model with log transformation
    fitDifferentSlope = lm(logPrice ~ logSqFt * Neighborhood, data = C21Ames)

# Connect to RShiny
    shinyServer(function(input, output) {

    # Reactive expression for the Scatter plot
     output$regression_plot = renderPlot({
      filtered = C21Ames %>% filter(Neighborhood == input$neighborhood) # Using the
    original log model
      predicted_data = data.frame(logSqFt = log(input$sqft), logPrice =
    predict(fitDifferentSlope, newdata = data.frame(logSqFt = log(input$sqft),
    Neighborhood = input$neighborhood)), Neighborhood = "Predicted") #Create a
    separate data frame with the inputs from the app form to allow plotting a separate
    point for the estimated price
      ggplot() +
        geom_point(data = filtered, aes(x = logSqFt, y = logPrice, color = Neighborhood)) + #
    Scatter plot of the log transformed data
        geom_point(data = predicted_data, aes(x = logSqFt, y = logPrice), color = "orange",
    size = 3, shape = 17) + #plot the estimated point
        ggtitle(paste("Scatterplot of Sale Price by Square Feet in", input$neighborhood)) +
    #dynamic title to reflect which neighborhood we are observing
        xlab("Log(Square Feet)") + # name of the x axis
        ylab("Log(Sale Price)") + # name y axis
        scale_color_manual(values = c("NAmes" = "blue", "Edwards" = "blue", "BrkSide" =
    "blue", "Predicted" = "orange")) + # color the values
        theme(legend.position = "none")  # Remove the legend
     })


     # Reactive expression for the prediction using the user inputs from the form
     output$prediction_output = renderText({
      req(input$predict_button)
      new_data = data.frame(
        logSqFt = log(input$sqft), # convert the user input into log data to allow to combine
    into the dynamic graph
        Neighborhood = input$neighborhood # pull the response from the neighborhood
      )
      prediction = round(exp(predict(fitDifferentSlope, newdata = new_data, interval =
    "confidence")[1]))  # Extracting the point estimate and converting out of a log number
    for interpretability
      lower_bound = round(exp(predict(fitDifferentSlope, newdata = new_data, interval =
    "confidence")[2]))  # Extracting the lower bound of the interval and converting out of a
    log number for interpretability
      upper_bound = round(exp(predict(fitDifferentSlope, newdata = new_data, interval =
    "confidence")[3]))  # Extracting the upper bound of the interval and converting out of a
    log number for interpretability

      paste("Estimated Cost: $", prediction, " (95% CI: [$", lower_bound, ", $",
    upper_bound,"])") # display the estimated sales price for the provided square feet and
    neighborhood
     })
    })
```

Analysis 2 R Code:

```
### PROJECT ###

# Setup
```

```
#ames_data = read.csv(file.choose())
#ames_test_kaggle = read.csv(file.choose())
head(ames_data)

# Wrangling

## Check for normal only
library(dplyr)
ames_normal = ames_data[ames_data$SaleCondition == 'Normal',]
ames_adj = ames_data
ames_adj = ames_adj %>% select(-SaleCondition)
ames_adj[ames_adj == 'Ex'] = 1
ames_adj[ames_adj == 'Gd'] = 2
ames_adj[ames_adj == 'TA'] = 3
ames_adj[ames_adj == 'Fa'] = 4
ames_adj[ames_adj == 'Po'] = 5

ames_adj$CentralAir = ifelse(ames_adj$CentralAir == 'Y', 1, 0)
ames_adj$PavedDrive = ifelse(ames_adj$PavedDrive == 'Y', 1, 0)
ames_adj$LotShape = ifelse(ames_adj$LotShape == 'Reg', 1, 0)
ames_adj$Street = ifelse(ames_adj$Street == 'Pave', 1, 0)
ames_adj$LandContour = ifelse(ames_adj$LandContour == 'Lvl', 1, 0)
ames_adj$Exterior1st = ifelse(ames_adj$Exterior1st == 'AsbShng', 1, 0)
ames_adj$Exterior2nd = ifelse(ames_adj$Exterior2nd == 'AsbShng', 1, 0)
ames_adj$Functional = ifelse(ames_adj$Functional == 'Sev' | ames_adj$Functional == 'Sal', 1, 0)

ames_adj$ExterCond = as.numeric(ames_adj$ExterCond)
ames_adj$ExterQual = as.numeric(ames_adj$ExterQual)
ames_adj$HeatingQC = as.numeric(ames_adj$HeatingQC)
ames_adj$CentralAir = as.numeric(ames_adj$CentralAir)
ames_adj$KitchenQual = as.numeric(ames_adj$KitchenQual)
ames_adj$PavedDrive = as.numeric(ames_adj$PavedDrive)
ames_adj$LotShape = as.numeric(ames_adj$LotShape)
ames_adj$Street = as.numeric(ames_adj$Street)
ames_adj$LandContour = as.numeric(ames_adj$LandContour)
ames_adj$Exterior1st = as.numeric(ames_adj$Exterior1st)
ames_adj$Exterior2nd = as.numeric(ames_adj$Exterior2nd)
ames_adj$Functional = as.numeric(ames_adj$Functional)

ames_adj$LotFrontage[is.na(ames_adj$LotFrontage)] = mean(ames_adj$LotFrontage, na.rm=TRUE)
ames_adj$MasVnrArea[is.na(ames_adj$MasVnrArea)] = mean(ames_adj$MasVnrArea, na.rm=TRUE)
ames_adj$GarageYrBlt    [is.na(ames_adj$GarageYrBlt    )] = mean(ames_adj$GarageYrBlt    , na.rm=TRUE)
ames_adj$LotFrontage[is.na(ames_adj$LotFrontage)] = mean(ames_adj$LotFrontage, na.rm=TRUE)
ames_adj$LotFrontage[is.na(ames_adj$LotFrontage)] = mean(ames_adj$LotFrontage, na.rm=TRUE)
ames_adj$LotFrontage[is.na(ames_adj$LotFrontage)] = mean(ames_adj$LotFrontage, na.rm=TRUE)
ames_adj$LotFrontage[is.na(ames_adj$LotFrontage)] = mean(ames_adj$LotFrontage, na.rm=TRUE)

ames_adj$LogSalePrice = log(ames_adj$SalePrice)


factor_i = sapply(ames_adj, is.factor)
numeric_i = sapply(ames_adj, is.numeric)
```

```
## For Normal sales, only one Functional category
ames_adj = ames_adj %>% select(-c(Functional, ))

## Export to csv for SAS
write.csv(ames_adj, file.choose(), row.names=FALSE)

ames_log = ames_log %>% select(-c(Functional, ))

which(numeric_i)
ames_data_numeric = ames_adj[, numeric_i]

## Count unique values for each variable:
sapply(lapply(ames_data_numeric, unique), length)
lapply(ames_data_numeric, unique)

## Log transforms

ames_log = ames_data
ames_log$LogSalePrice = log(ames_data$SalePrice)
ames_log$LogOverallQual = log(ames_data$OverallQual)
ames_log = ames_log %>% select(-c(SalePrice, ))
numeric_i = sapply(ames_log, is.numeric)


ames_log_numeric = ames_log[, numeric_i]

#-----------

# Part 1. Best single predictor
fit = lm(LogSalePrice ~ OverallQual, ames_log_numeric)
summary(fit)

fit2 = lm(LogSalePrice ~ GrLivArea, ames_log_numeric)
summary(fit2)



ggplot(ames_data, aes(x=LogOverallQual, y=log(SalePrice))) +
  geom_point() +
  geom_smooth(method='lm')

ggplot(ames_data, aes(x=OverallQual, y=SalePrice)) +
  geom_point() +
  geom_smooth(method='lm')


## Check for interactions
fit = lm(LogSalePrice ~., data=ames_log_numeric)

## Stepwise
step_aic = ols_step_both_aic(fit, details=TRUE)
step_adjr = ols_step_both_adj_r2(fit, details=TRUE)
```

```
#-----------

# Part 2. MLR with GrLivArea + FullBath

fit = lm(SalePrice ~ GrLivArea + FullBath, ames_sorted)
summary(fit)


# Split data into train and test
set.seed(1)

sample = sample(c(TRUE, FALSE), nrow(ames_sorted), replace=TRUE, prob=c(0.7, 0.3))
ames_train = ames_sorted[sample, ]
ames_test = ames_sorted[!sample, ]

# Fit model to training data
fit = lm(SalePrice ~ GrLivArea + FullBath, ames_train)

# Generate predictions based on model
predictions = predict(fit, ames_test)

# Compare predicted value to actual
ames_compare = data.frame(Predicted=predictions, Actual=ames_test$SalePrice)
ames_compare$ID = 1:nrow(ames_compare)
ames_compare$Diff = ames_compare$Predicted - ames_compare$Actual

# Plot the differences
ames_plot = data.frame(x=rep(1:nrow(ames_compare), 2),
                       value=c(ames_compare$Predicted, ames_compare$Actual),
                       variable=c(rep('Predicted', nrow(ames_compare)), rep('Actual', nrow(ames_compare)))
                       )
ggplot(ames_plot, aes(x=x, y=value)) +
  geom_line(aes(color=variable))


#-------------

# Part 3. Second attempt in R
library(olsrr)
library(tidyverse)
library(caret)
library(GGally)

head(ames_data)

## Count unique values for each variable:
sapply(lapply(ames_data, unique), length)

#ggpairs(ames_data_numeric)

## Check for interactions
```

```
fit = lm(SalePrice ~., data=ames_data_numeric)

## Stepwise
step_aic = ols_step_both_aic(fit, details=TRUE)
step_adjr = ols_step_both_adj_r2(fit, details=TRUE)

## Train Control
train_control<- trainControl(method="LOOCV")
model <- train(SalePrice ~ GrLivArea + FullBath, data=ames_data_numeric, trControl=train_control, method="lm")
model

nrow(ames_data_numeric[complete.cases(ames_data_numeric),])
sapply(ames_data_numeric, function(x)any(is.na(x)))

nrow(na.exclude(ames_data_numeric))

model <- train(SalePrice ~ LotArea*LandContour + LandContour*OverallQual + YearBuilt*YearRemodAdd +
LotArea*ExterQual + OverallCond*ExterQual + ExterQual*BsmtFinSF1 + LotShape*BsmtUnfSF +
BsmtFinSF2*BsmtUnfSF + OverallQual*TotalBsmtSF + OverallCond*TotalBsmtSF + ExterQual*TotalBsmtSF +
BsmtFinSF1*TotalBsmtSF + BsmtUnfSF*TotalBsmtSF + MSSubClass*X2ndFlrSF + MasVnrArea*X2ndFlrSF +
TotalBsmtSF*X2ndFlrSF + Street*GrLivArea + OverallQual*GrLivArea + OverallQual*BsmtFullBath +
MSSubClass*BedroomAbvGr + BsmtFullBath*BedroomAbvGr + LandContour*KitchenQual +
GrLivArea*KitchenQual + BsmtFullBath*TotRmsAbvGrd + LotArea*Fireplaces + OverallCond*Fireplaces +
BsmtFullBath*Fireplaces + FullBath*Fireplaces + BedroomAbvGr*Fireplaces + LotArea*GarageCars +
Fireplaces*GarageCars + FullBath*GarageArea + CentralAir*PavedDrive + EnclosedPorch*X3SsnPorch +
BsmtUnfSF*ScreenPorch + X2ndFlrSF*PoolArea + YearRemodAdd*YrSold + KitchenAbvGr*OverallQual,
data=ames_data_numeric,trControl=train_control, method="lm")
model

predictions = predict(model, ames_adj)
summary(predictions)
ames_adj$PredictedSalePrice = predictions

compare_df = data.frame(SalePrice = c(ames_adj$SalePrice, ames_adj$PredictedSalePrice),
Variable=c(rep('Actual', nrow(ames_adj)), rep('Predicted', nrow(ames_adj))),ID = rep(rank(ames_adj$SalePrice), 2))



ggplot(compare_df, aes(x=ID, y=SalePrice)) +
  geom_point(aes(color=Variable), alpha=0.3)




# Kaggle Submission

## Setup (Copied from ames_adj):
ames_test_kaggle[ames_test_kaggle == 'Ex'] = 1
ames_test_kaggle[ames_test_kaggle == 'Gd'] = 2
ames_test_kaggle[ames_test_kaggle == 'TA'] = 3
ames_test_kaggle[ames_test_kaggle == 'Fa'] = 4
ames_test_kaggle[ames_test_kaggle == 'Po'] = 5
```

```
ames_test_kaggle$CentralAir = ifelse(ames_test_kaggle$CentralAir == 'Y', 1, 0)
ames_test_kaggle$PavedDrive = ifelse(ames_test_kaggle$PavedDrive == 'Y', 1, 0)
ames_test_kaggle$LotShape = ifelse(ames_test_kaggle$LotShape == 'Reg', 1, 0)
ames_test_kaggle$Street = ifelse(ames_test_kaggle$Street == 'Pave', 1, 0)
ames_test_kaggle$LandContour = ifelse(ames_test_kaggle$LandContour == 'Lvl', 1, 0)
ames_test_kaggle$Exterior1st = ifelse(ames_test_kaggle$Exterior1st == 'AsbShng', 1, 0)
ames_test_kaggle$Exterior2nd = ifelse(ames_test_kaggle$Exterior2nd == 'AsbShng', 1, 0)
ames_test_kaggle$Functional = ifelse(ames_test_kaggle$Functional == 'Sev' | ames_test_kaggle$Functional ==
'Sal', 1, 0)

ames_test_kaggle$ExterCond = as.numeric(ames_test_kaggle$ExterCond)
ames_test_kaggle$ExterQual = as.numeric(ames_test_kaggle$ExterQual)
ames_test_kaggle$HeatingQC = as.numeric(ames_test_kaggle$HeatingQC)
ames_test_kaggle$CentralAir = as.numeric(ames_test_kaggle$CentralAir)
ames_test_kaggle$KitchenQual = as.numeric(ames_test_kaggle$KitchenQual)
ames_test_kaggle$PavedDrive = as.numeric(ames_test_kaggle$PavedDrive)
ames_test_kaggle$LotShape = as.numeric(ames_test_kaggle$LotShape)
ames_test_kaggle$Street = as.numeric(ames_test_kaggle$Street)
ames_test_kaggle$LandContour = as.numeric(ames_test_kaggle$LandContour)
ames_test_kaggle$Exterior1st = as.numeric(ames_test_kaggle$Exterior1st)
ames_test_kaggle$Exterior2nd = as.numeric(ames_test_kaggle$Exterior2nd)
ames_test_kaggle$Functional = as.numeric(ames_test_kaggle$Functional)

ames_test_kaggle$LotFrontage[is.na(ames_test_kaggle$LotFrontage)] = mean(ames_test_kaggle$LotFrontage,
na.rm=TRUE)
ames_test_kaggle$MasVnrArea[is.na(ames_test_kaggle$MasVnrArea)] = mean(ames_test_kaggle$MasVnrArea,
na.rm=TRUE)
ames_test_kaggle$GarageYrBlt    [is.na(ames_test_kaggle$GarageYrBlt    )] = mean(ames_test_kaggle$GarageYrBlt
, na.rm=TRUE)
ames_test_kaggle$TotalBsmtSF[is.na(ames_test_kaggle$TotalBsmtSF)] = mean(ames_test_kaggle$TotalBsmtSF,
na.rm=TRUE)
ames_test_kaggle$GarageArea[is.na(ames_test_kaggle$GarageArea)] = mean(ames_test_kaggle$GarageArea,
na.rm=TRUE)
ames_test_kaggle$BsmtUnfSF[is.na(ames_test_kaggle$BsmtUnfSF)] = mean(ames_test_kaggle$BsmtUnfSF,
na.rm=TRUE)
ames_test_kaggle$BsmtFullBath[is.na(ames_test_kaggle$BsmtFullBath)] = mean(ames_test_kaggle$BsmtFullBath,
na.rm=TRUE)
ames_test_kaggle$BsmtFinSF2[is.na(ames_test_kaggle$BsmtFinSF2)] = mean(ames_test_kaggle$BsmtFinSF2,
na.rm=TRUE)
ames_test_kaggle$BsmtFinSF1[is.na(ames_test_kaggle$BsmtFinSF1)] = mean(ames_test_kaggle$BsmtFinSF1,
na.rm=TRUE)
ames_test_kaggle$KitchenQual[is.na(ames_test_kaggle$KitchenQual)] = mean(ames_test_kaggle$KitchenQual,
na.rm=TRUE)
ames_test_kaggle$GarageCars[is.na(ames_test_kaggle$GarageCars)] = mean(ames_test_kaggle$GarageCars,
na.rm=TRUE)
ames_test_kaggle$LotFrontage[is.na(ames_test_kaggle$LotFrontage)] = mean(ames_test_kaggle$LotFrontage,
na.rm=TRUE)
ames_test_kaggle$LotFrontage[is.na(ames_test_kaggle$LotFrontage)] = mean(ames_test_kaggle$LotFrontage,
na.rm=TRUE)
ames_test_kaggle$LotFrontage[is.na(ames_test_kaggle$LotFrontage)] = mean(ames_test_kaggle$LotFrontage,
na.rm=TRUE)
ames_test_kaggle$LotFrontage[is.na(ames_test_kaggle$LotFrontage)] = mean(ames_test_kaggle$LotFrontage,
na.rm=TRUE)
```

```
ames_test_kaggle$LotFrontage[is.na(ames_test_kaggle$LotFrontage)] = mean(ames_test_kaggle$LotFrontage,
na.rm=TRUE)

sapply(ames_test_kaggle, function(x) sum(is.na(x)))
ames_test_kaggle[complete.cases(ames_test_kaggle),]

## 1. SLR

model <- train(SalePrice ~ OverallQual, data=ames_data_numeric, trControl=train_control, method="lm")
model

kaggle_predictions = predict(model, newdata=ames_test_kaggle)
kaggle_predictions

ames_test_kaggle$Predictions = kaggle_predictions

kaggle_submission = data.frame(Id=ames_test_kaggle$Id, SalePrice=ames_test_kaggle$Predictions)
write.csv(kaggle_submission, file.choose(), row.names=FALSE)

## 2. MLR
model <- train(SalePrice ~ GrLivArea + FullBath, data=ames_data_numeric, trControl=train_control, method="lm")
model

kaggle_predictions = predict(model, newdata=ames_test_kaggle)
kaggle_predictions

ames_test_kaggle$Predictions = kaggle_predictions

kaggle_submission = data.frame(Id=ames_test_kaggle$Id, SalePrice=ames_test_kaggle$Predictions)
write.csv(kaggle_submission, file.choose(), row.names=FALSE)


## 3. Custom
model <- train(LogSalePrice ~ LotArea*LandContour + LandContour*OverallQual + YearBuilt*YearRemodAdd +
LotArea*ExterQual + OverallCond*ExterQual + ExterQual*BsmtFinSF1 + LotShape*BsmtUnfSF +
BsmtFinSF2*BsmtUnfSF + OverallQual*TotalBsmtSF + OverallCond*TotalBsmtSF + ExterQual*TotalBsmtSF +
BsmtFinSF1*TotalBsmtSF + BsmtUnfSF*TotalBsmtSF + MSSubClass*X2ndFlrSF + MasVnrArea*X2ndFlrSF +
TotalBsmtSF*X2ndFlrSF + Street*GrLivArea + OverallQual*GrLivArea + OverallQual*BsmtFullBath +
MSSubClass*BedroomAbvGr + BsmtFullBath*BedroomAbvGr + LandContour*KitchenQual +
GrLivArea*KitchenQual + BsmtFullBath*TotRmsAbvGrd + LotArea*Fireplaces + OverallCond*Fireplaces +
BsmtFullBath*Fireplaces + FullBath*Fireplaces + BedroomAbvGr*Fireplaces + LotArea*GarageCars +
Fireplaces*GarageCars + FullBath*GarageArea + CentralAir*PavedDrive + EnclosedPorch*X3SsnPorch +
BsmtUnfSF*ScreenPorch + X2ndFlrSF*PoolArea + YearRemodAdd*YrSold + KitchenAbvGr*OverallQual,
data=ames_data_numeric,trControl=train_control, method="lm")
model

kaggle_predictions = predict(model, newdata=ames_test_kaggle)
kaggle_predictions

ames_test_kaggle$Predictions = exp(kaggle_predictions)

kaggle_submission = data.frame(Id=ames_test_kaggle$Id, SalePrice=ames_test_kaggle$Predictions)
write.csv(kaggle_submission, file.choose(), row.names=FALSE)
```

```
## 4. Custom 2
model <- train(SalePrice ~ OverallQual + GrLivArea + OverallQual*GrLivArea + OverallQual*BsmtFinSF1 +
GrLivArea*BsmtFinSF1 + OverallQual*TotalBsmtSF + GrLivArea*TotalBsmtSF + YearRemodAdd +
GrLivArea*YearRemodAdd, data=ames_data_numeric, trControl=train_control, method="lm")
model

kaggle_predictions = predict(model, newdata=ames_test_kaggle)
kaggle_predictions

ames_test_kaggle$Predictions = kaggle_predictions

kaggle_submission = data.frame(Id=ames_test_kaggle$Id, SalePrice=ames_test_kaggle$Predictions)
write.csv(kaggle_submission, file.choose(), row.names=FALSE)
```

Analysis 2 SAS Code:

```
FILENAME REFFILE '/home/u63732424/sasuser.v94/ames_adj.csv';

PROC IMPORT REPLACE DATAFILE=REFFILE DBMS=CSV OUT=ames;
        GETNAMES=YES;
run;

DATA ames_abr;
        SET ames;
        if _n_ = 564 then delete;
        if _n_ = 278 then delete;
        LogSalePrice = log(SalePrice);
run;

proc glmselect data=ames;
        model SalePrice=GrLivArea FullBath / selection=Forward(stop=SL SLE=0.5)
                stats=adjrsq CVDETAILS;
run;

proc glmselect data=ames;
        model SalePrice=GrLivArea FullBath / selection=Forward(stop=CV)
                cvmethod=random(5) stats=adjrsq CVDETAILS;
run;

proc glmselect data=ames;
        model SalePrice=MSSubClass LotArea OverallQual OverallCond YearBuilt
                YearRemodAdd MasVnrArea BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF
                LowQualFinSF GrLivArea BsmtFullBath BsmtHalfBath FullBath HalfBath
                BedroomAbvGr KitchenAbvGr TotRmsAbvGrd Fireplaces GarageYrBlt
GarageCars
                GarageArea WoodDeckSF OpenPorchSF EnclosedPorch ScreenPorch PoolArea
MiscVal
                MoSold YrSold / selection=Forward(stop=SL SLE=0.1) stats=adjrsq
CVDETAILS;
run;

proc glmselect data=ames;
        model SalePrice=MSSubClass | LotArea | OverallQual | OverallCond | YearBuilt
```

```
| YearRemodAdd | MasVnrArea | BsmtFinSF1 | BsmtFinSF2 | BsmtUnfSF | TotalBsmtSF |
LowQualFinSF | GrLivArea | BsmtFullBath | BsmtHalfBath | FullBath | HalfBath |
BedroomAbvGr | KitchenAbvGr | TotRmsAbvGrd | Fireplaces | GarageYrBlt | GarageCars |
GarageArea | WoodDeckSF | OpenPorchSF | EnclosedPorch | ScreenPorch | PoolArea |
MiscVal | MoSold | YrSold@2
                / selection=Forward(stop=SL SLE=0.2) cvmethod=random stats=adjrsq
CVDETAILS;
run;

proc glmselect data=ames;
        model SalePrice=MSSubClass | LotArea | OverallQual | OverallCond | YearBuilt
| YearRemodAdd | MasVnrArea | BsmtFinSF1 | BsmtFinSF2 | BsmtUnfSF | TotalBsmtSF |
LowQualFinSF | GrLivArea | BsmtFullBath | BsmtHalfBath | FullBath | HalfBath |
BedroomAbvGr | KitchenAbvGr | TotRmsAbvGrd | Fireplaces | GarageYrBlt | GarageCars |
GarageArea | WoodDeckSF | OpenPorchSF | EnclosedPorch | ScreenPorch | PoolArea |
MiscVal | MoSold | YrSold@2
                / selection=Forward(stop=CV) cvmethod=random(5) stats=adjrsq
CVDETAILS;
run;

proc reg data=ames plots(label)=(CooksD all);
        model SalePrice=MSSubClass LotArea Street LotShape LandContour OverallQual
                OverallCond YearBuilt YearRemodAdd Exterior1st Exterior2nd
MasVnrArea
                ExterQual ExterCond BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF
HeatingQC
                CentralAir X1stFlrSF X2ndFlrSF LowQualFinSF GrLivArea BsmtFullBath
                BsmtHalfBath FullBath HalfBath BedroomAbvGr KitchenAbvGr KitchenQual
                TotRmsAbvGrd Fireplaces GarageYrBlt GarageCars GarageArea PavedDrive
                WoodDeckSF OpenPorchSF EnclosedPorch X3SsnPorch ScreenPorch PoolArea
MiscVal
                MoSold YrSold LogOverallQual / VIF;
        run;

proc glmselect data=ames;
        model SalePrice=MSSubClass LotArea Street LotShape LandContour OverallQual
                OverallCond YearBuilt YearRemodAdd Exterior1st Exterior2nd
MasVnrArea
                ExterQual ExterCond BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF
HeatingQC
                CentralAir X1stFlrSF X2ndFlrSF LowQualFinSF GrLivArea BsmtFullBath
                BsmtHalfBath FullBath HalfBath BedroomAbvGr KitchenAbvGr KitchenQual
                TotRmsAbvGrd Fireplaces GarageYrBlt GarageCars GarageArea PavedDrive
                WoodDeckSF OpenPorchSF EnclosedPorch X3SsnPorch ScreenPorch PoolArea
MiscVal
                MoSold YrSold LogOverallQual / selection=Forward(stop=SL SLE=0.05)
                stats=adjrsq CVDETAILS;
run;

proc glmselect data=ames;
        model SalePrice=MSSubClass LotArea Street LotShape LandContour OverallQual
                OverallCond YearBuilt YearRemodAdd Exterior1st Exterior2nd
MasVnrArea
                ExterQual ExterCond BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF
HeatingQC
```

```
                CentralAir X1stFlrSF X2ndFlrSF LowQualFinSF GrLivArea BsmtFullBath
                BsmtHalfBath FullBath HalfBath BedroomAbvGr KitchenAbvGr KitchenQual
                TotRmsAbvGrd Fireplaces GarageYrBlt GarageCars GarageArea PavedDrive
                WoodDeckSF OpenPorchSF EnclosedPorch X3SsnPorch ScreenPorch PoolArea
MiscVal
                MoSold YrSold LogOverallQual / selection=Forward(stop=SL SLE=0.05)
                stats=adjrsq CVDETAILS;
run;


proc glmselect data=ames;
        model SalePrice= OverallQual GrLivArea BsmtFinSF1 TotalBsmtSF YearRemodAdd
LotArea / selection=Forward(stop=CV) cvmethod=random(5)
                stats=adjrsq CVDETAILS;
run;

proc reg data=ames plots(label)=(CooksD all);
        model SalePrice= OverallQual GrLivArea BsmtFinSF1 TotalBsmtSF YearRemodAdd
LotArea / VIF;
        run;

proc glmselect data=ames;
        model SalePrice=MSSubClass | LotArea | LandContour | Street | LotShape |
OverallQual | OverallCond | YearBuilt | YearRemodAdd | Exterior1st | Exterior2nd |
MasVnrArea | ExterQual | ExterCond | BsmtFinSF1 | BsmtFinSF2 | BsmtUnfSF |
TotalBsmtSF | HeatingQC | CentralAir | X1stFlrSF | X2ndFlrSF | LowQualFinSF |
GrLivArea | BsmtFullBath | BsmtHalfBath | FullBath | HalfBath | BedroomAbvGr |
KitchenAbvGr | KitchenQual | TotRmsAbvGrd | Fireplaces | GarageYrBlt | GarageCars |
GarageArea | PavedDrive | WoodDeckSF | OpenPorchSF | EnclosedPorch | X3SsnPorch |
ScreenPorch | PoolArea | MiscVal | MoSold | YrSold | LogOverallQual@2
                / selection=Forward(stop=CV) cvmethod=random(5) stats=adjrsq
CVDETAILS;
run;

proc reg data=ames_abr plots(label)=(CooksD all);
        model SalePrice= OverallQual GrLivArea BsmtFinSF1 TotalBsmtSF YearRemodAdd
LotArea / VIF;
        run;

proc glmselect data=ames_abr;
        model SalePrice= OverallQual GrLivArea BsmtFinSF1 TotalBsmtSF YearRemodAdd
LotArea / selection=Forward(stop=CV) cvmethod=random(5)
                stats=adjrsq CVDETAILS;
run;

/* SLR */
proc glmselect data=ames_abr;
        model LogSalePrice= OverallQual / selection=Forward(stop=CV)
cvmethod=random(5)
                stats=adjrsq CVDETAILS;
run;

/* MLR-1 */
proc glmselect data=ames_abr;
        model LogSalePrice=GrLivArea FullBath / selection=Forward(stop=CV)
```

```
cvmethod=random(5)
                stats=adjrsq CVDETAILS;
run;

/* MLR-2 */

proc glmselect data=ames;
        model LogSalePrice=MSSubClass | LotArea | LandContour | Street | LotShape |
OverallQual | OverallCond | YearBuilt | YearRemodAdd | Exterior1st | Exterior2nd |
MasVnrArea | ExterQual | ExterCond | BsmtFinSF1 | BsmtFinSF2 | BsmtUnfSF |
TotalBsmtSF | HeatingQC | CentralAir | X1stFlrSF | X2ndFlrSF | LowQualFinSF |
GrLivArea | BsmtFullBath | BsmtHalfBath | FullBath | HalfBath | BedroomAbvGr |
KitchenAbvGr | KitchenQual | TotRmsAbvGrd | Fireplaces | GarageYrBlt | GarageCars |
GarageArea | PavedDrive | WoodDeckSF | OpenPorchSF | EnclosedPorch | X3SsnPorch |
ScreenPorch | PoolArea | MiscVal | MoSold | YrSold | LogOverallQual@2
                / selection=Forward(stop=CV) cvmethod=random(5) stats=adjrsq
CVDETAILS;
run;

/* MLR-3 */
proc glmselect data=ames_abr;
        model LogSalePrice= OverallQual | GrLivArea | BsmtFinSF1 | TotalBsmtSF |
YearRemodAdd@2 / selection=Forward(stop=CV) cvmethod=random(5)
                stats=adjrsq CVDETAILS;
run;


proc reg data=ames plots(label)=(CooksD all);
        model SalePrice= OverallQual GrLivArea BsmtFinSF1 TotalBsmtSF YearRemodAdd
LotArea / VIF;
run;

proc reg data=ames_abr plots(label)=(CooksD all);
        model LogSalePrice = OverallQual GrLivArea BsmtFinSF1 TotalBsmtSF
YearRemodAdd LotArea / VIF;
run;
```

Model Estimates

SLR

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 10.58444    0.02727  388.18   <2e-16 ***
OverallQual  0.23603    0.00436   54.14   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2303 on 1458 degrees of freedom
Multiple R-squared:  0.6678,   Adjusted R-squared:  0.6676
F-statistic:  2931 on 1 and 1458 DF,  p-value: < 2.2e-16
```

MLR-1

```
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.111e+01  2.385e-02  465.96   <2e-16 ***
GrLivArea   4.112e-04  1.758e-05   23.39   <2e-16 ***
FullBath    1.842e-01  1.677e-02   10.98   <2e-16 ***
```

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

Residual standard error: 0.274 on 1457 degrees of freedom
Multiple R-squared:  0.5302,	Adjusted R-squared:  0.5296 
F-statistic: 822.2 on 2 and 1457 DF,  p-value: < 2.2e-16
```

MLR-2

Formula:

$$
\begin{aligned}
\log(SalePrice) \\
&= \beta_0 \ + \ \beta_1(LotArea * LandContour) + \beta_2(LandContour * OverallQual) \\
&+ \beta_3(YearBuilt * YearRemodAdd) + \beta_4(LotArea * ExterQual) \\
&+ \beta_5(OverallCond * ExterQual) + \beta_6(ExterQual * BsmtFinSF1) \\
&+ \beta_7(LotShape * BsmtUnfSF) + \beta_8(BsmtFinSF2 * BsmtUnfSF) \\
&+ \beta_9(OverallQual * TotalBsmtSF) + \beta_{10}(OverallCond * TotalBsmtSF) \\
&+ \beta_{11}(ExterQual * TotalBsmtSF) + \beta_{12}(BsmtFinSF1 * TotalBsmtSF) \\
&+ \beta_{13}(BsmtUnfSF * TotalBsmtSF) + \beta_{14}(MSSubClass * X2ndFlrSF) \\
&+ \beta_{15}(MasVnrArea * X2ndFlrSF) + \beta_{16}(TotalBsmtSF * X2ndFlrSF) \\
&+ \beta_{17}(Street * GrLivArea) + \beta_{18}(OverallQual * GrLivArea) \\
&+ \beta_{19}(OverallQual * BsmtFullBath) + \beta_{20}(MSSubClass * BedroomAbvGr) \\
&+ \beta_{21}(BsmtFullBath * BedroomAbvGr) + \beta_{22}(LandContour \\
&* KitchenQual) + \beta_{23}(GrLivArea * KitchenQual) + \beta_{24}(BsmtFullBath \\
&* TotRmsAbvGrd) + \beta_{25}(LotArea * Fireplaces) + \beta_{26}(OverallCond \\
&* Fireplaces) + \beta_{27}(BsmtFullBath * Fireplaces) + \beta_{28}(FullBath \\
&* Fireplaces) + \beta_{29}(BedroomAbvGr * Fireplaces) + \beta_{30}(LotArea \\
&* GarageCars) + \beta_{31}(Fireplaces * GarageCars) + \beta_{32}(FullBath \\
&* GarageArea) + \beta_{33}(CentralAir * PavedDrive) + \beta_{34}(EnclosedPorch \\
&* X3SsnPorch) + \beta_{35}(BsmtUnfSF * ScreenPorch) + \beta_{36}(X2ndFlrSF \\
&* PoolArea) + \beta_{37}(YearRemodAdd * YrSold) + \ \beta_{38}(KitchenAbvGr \\
&* OverallQual)
\end{aligned}
$$

```
                              Estimate Std. Error t value Pr(>|t|)    
(Intercept)                  1.428e+03  4.797e+02   2.978 0.002955 ** 
LotArea                      1.194e-06  4.083e-06   0.293 0.769943    
LandContour                  9.464e-02  9.459e-02   1.000 0.317275    
OverallQual                  1.212e-01  2.402e-02   5.048 5.05e-07 ***
YearBuilt                    1.335e-03  1.562e-02   0.085 0.931886    
YearRemodAdd                -7.117e-01  2.418e-01  -2.944 0.003294 ** 
ExterQual                   -6.671e-02  4.675e-02  -1.427 0.153800    
OverallCond                  1.288e-02  2.132e-02   0.604 0.546058    
BsmtFinSF1                   2.968e-04  1.192e-04   2.489 0.012931 *  
LotShape                     1.428e-02  1.185e-02   1.206 0.228160    
BsmtUnfSF                    1.709e-04  1.276e-04   1.340 0.180593    
BsmtFinSF2                   1.007e-04  1.242e-04   0.810 0.417925    
TotalBsmtSF                         NA         NA      NA       NA    
MSSubClass                  -5.119e-04  2.807e-04  -1.824 0.068344 .  
X2ndFlrSF                    1.684e-04  3.742e-05   4.500 7.37e-06 ***
MasVnrArea                  -2.680e-05  3.150e-05  -0.851 0.395075    
Street                       6.666e-02  1.741e-01   0.383 0.701883    
GrLivArea                    3.748e-04  1.632e-04   2.297 0.021786 *  
BsmtFullBath                 3.023e-02  4.312e-02   0.701 0.483294    
BedroomAbvGr                -5.340e-03  1.215e-02  -0.440 0.660280    
KitchenQual                  5.401e-02  2.933e-02   1.841 0.065784 .  
TotRmsAbvGrd                 4.782e-03  5.261e-03   0.909 0.363571    
Fireplaces                   6.960e-02  4.163e-02   1.672 0.094829 .  
```

```
FullBath                     4.279e-02  1.927e-02   2.221 0.026514 *  
GarageCars                   4.964e-02  1.445e-02   3.435 0.000610 ***
GarageArea                   8.785e-05  6.378e-05   1.377 0.168588    
CentralAir                   8.307e-02  2.554e-02   3.252 0.001172 ** 
PavedDrive                   6.111e-02  2.665e-02   2.293 0.022006 *  
EnclosedPorch                8.399e-05  6.086e-05   1.380 0.167772    
X3SsnPorch                   1.080e-04  1.132e-04   0.954 0.340274    
ScreenPorch                 -4.116e-05  1.051e-04  -0.391 0.695532    
PoolArea                     1.150e-04  1.262e-04   0.911 0.362401    
YrSold                      -7.086e-01  2.388e-01  -2.967 0.003063 ** 
KitchenAbvGr                 2.032e-01  8.692e-02   2.338 0.019527 *  
`LotArea:LandContour`        5.961e-06  1.420e-06   4.199 2.86e-05 ***
`LandContour:OverallQual`   -1.719e-02  9.011e-03  -1.908 0.056600 .  
`YearBuilt:YearRemodAdd`     6.449e-07  7.849e-06   0.082 0.934526    
`LotArea:ExterQual`          2.664e-06  8.332e-07   3.198 0.001415 ** 
`ExterQual:OverallCond`      9.062e-03  6.080e-03   1.490 0.136355    
`ExterQual:BsmtFinSF1`      -8.377e-06  1.779e-05  -0.471 0.637809    
`LotShape:BsmtUnfSF`        -3.417e-05  1.561e-05  -2.189 0.028736 *  
`BsmtUnfSF:BsmtFinSF2`      -1.477e-07  7.423e-08  -1.990 0.046837 *  
`OverallQual:TotalBsmtSF`    3.398e-05  1.071e-05   3.172 0.001548 ** 
`OverallCond:TotalBsmtSF`    6.870e-06  9.572e-06   0.718 0.473085    
`ExterQual:TotalBsmtSF`     -2.915e-05  2.449e-05  -1.190 0.234080    
`BsmtFinSF1:TotalBsmtSF`    -1.189e-07  9.658e-09 -12.313  < 2e-16 ***
`BsmtUnfSF:TotalBsmtSF`     -8.038e-08  2.128e-08  -3.777 0.000166 ***
`MSSubClass:X2ndFlrSF`      -7.050e-07  2.850e-07  -2.474 0.013497 *  
`X2ndFlrSF:MasVnrArea`       7.587e-08  3.902e-08   1.945 0.052035 .  
`TotalBsmtSF:X2ndFlrSF`     -9.952e-08  2.629e-08  -3.785 0.000160 ***
`Street:GrLivArea`           8.254e-05  1.485e-04   0.556 0.578399    
`OverallQual:GrLivArea`     -2.226e-05  7.612e-06  -2.924 0.003515 ** 
`OverallQual:BsmtFullBath`   7.457e-03  6.960e-03   1.071 0.284140    
`MSSubClass:BedroomAbvGr`    1.482e-04  1.114e-04   1.331 0.183510    
`BsmtFullBath:BedroomAbvGr` -1.232e-02  1.031e-02  -1.195 0.232199    
`LandContour:KitchenQual`   -2.919e-02  1.951e-02  -1.496 0.134867    
`GrLivArea:KitchenQual`     -3.434e-05  1.339e-05  -2.565 0.010415 *  
`BsmtFullBath:TotRmsAbvGrd`  1.873e-03  6.286e-03   0.298 0.765771    
`LotArea:Fireplaces`        -3.225e-07  8.347e-07  -0.386 0.699238    
`OverallCond:Fireplaces`     5.325e-03  5.063e-03   1.052 0.293086    
`BsmtFullBath:Fireplaces`   -4.121e-02  1.143e-02  -3.605 0.000324 ***
`Fireplaces:FullBath`       -3.596e-02  1.221e-02  -2.946 0.003277 ** 
`BedroomAbvGr:Fireplaces`   -1.095e-02  7.508e-03  -1.458 0.145153    
`LotArea:GarageCars`        -2.741e-06  1.105e-06  -2.480 0.013266 *  
`Fireplaces:GarageCars`      2.726e-02  9.788e-03   2.785 0.005418 ** 
`FullBath:GarageArea`       -5.797e-06  3.324e-05  -0.174 0.861576    
`CentralAir:PavedDrive`     -3.663e-02  3.080e-02  -1.189 0.234565    
`EnclosedPorch:X3SsnPorch`  -2.093e-05  1.078e-05  -1.942 0.052344 .  
`BsmtUnfSF:ScreenPorch`      4.334e-07  1.506e-07   2.878 0.004069 ** 
`X2ndFlrSF:PoolArea`         5.685e-08  1.274e-07   0.446 0.655611    
`YearRemodAdd:YrSold`        3.543e-04  1.204e-04   2.943 0.003300 ** 
`OverallQual:KitchenAbvGr`  -5.167e-02  1.719e-02  -3.005 0.002701 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1231 on 1389 degrees of freedom
Multiple R-squared:  0.9096,	Adjusted R-squared:  0.9051 
F-statistic: 199.7 on 70 and 1389 DF,  p-value: < 2.2e-16
```

MLR-3

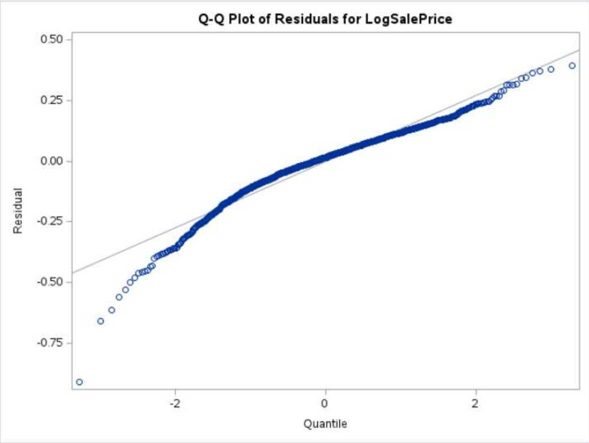
```
                            Estimate Std. Error t value Pr(>|t|)    
(Intercept)                8.457e+00  1.516e+00   5.579 2.88e-08 ***
OverallQual                7.341e-02  1.230e-02   5.970 2.99e-09 ***
GrLivArea                 -2.840e-03  1.041e-03  -2.727 0.006464 ** 
BsmtFinSF1                 3.605e-04  5.014e-05   7.190 1.04e-12 ***
TotalBsmtSF                2.046e-04  5.447e-05   3.757 0.000179 ***
```

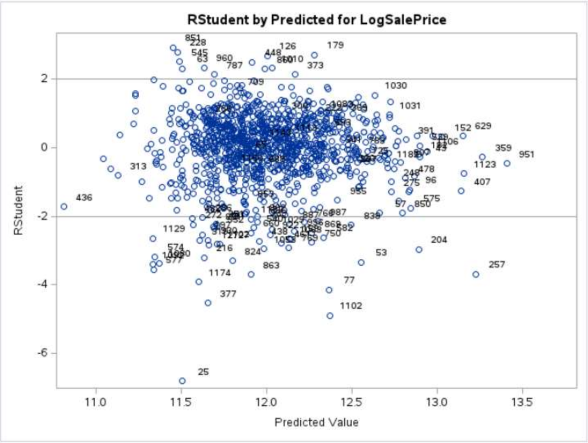
```
YearRemodAdd                  1.079e-03  7.788e-04   1.386 0.166071    
`OverallQual:GrLivArea`    4.352e-06  6.890e-06   0.632 0.527766    
`OverallQual:BsmtFinSF1`  -2.812e-05  8.755e-06  -3.212 0.001346 ** 
`GrLivArea:BsmtFinSF1`    -1.535e-08  2.275e-08  -0.675 0.500041    
`OverallQual:TotalBsmtSF`  3.813e-05  8.854e-06   4.306 1.77e-05 ***
`GrLivArea:TotalBsmtSF`   -1.743e-07  2.428e-08  -7.180 1.11e-12 ***
`GrLivArea:YearRemodAdd`   1.658e-06  5.344e-07   3.103 0.001952 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1559 on 1448 degrees of freedom
Multiple R-squared:  0.8489,	Adjusted R-squared:  0.8477 
F-statistic: 739.3 on 11 and 1448 DF,  p-value: < 2.2e-16
```

Figures

Analysis Question 2

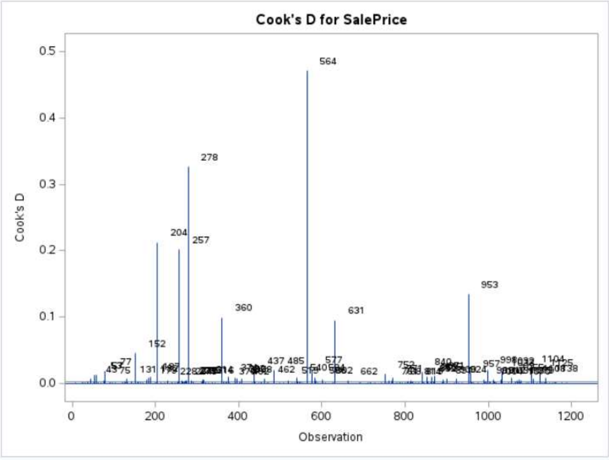
Residual Plots

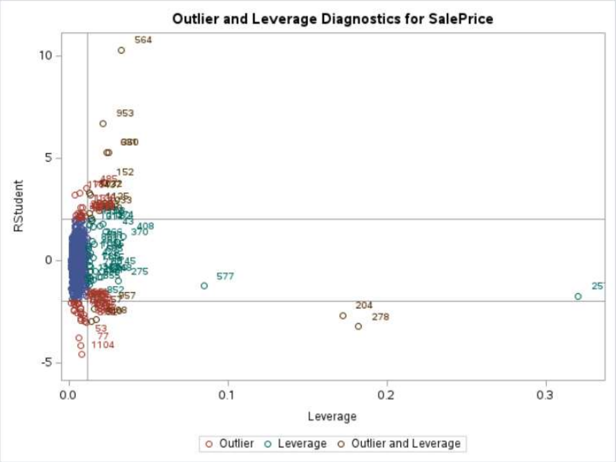




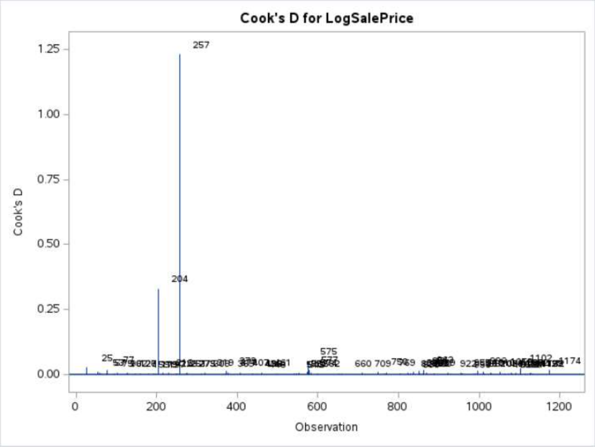
Cook’s D and Leverage

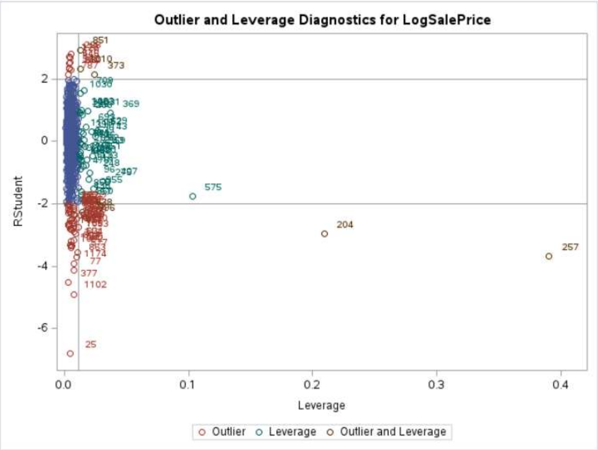
Before Removal/Transformation:



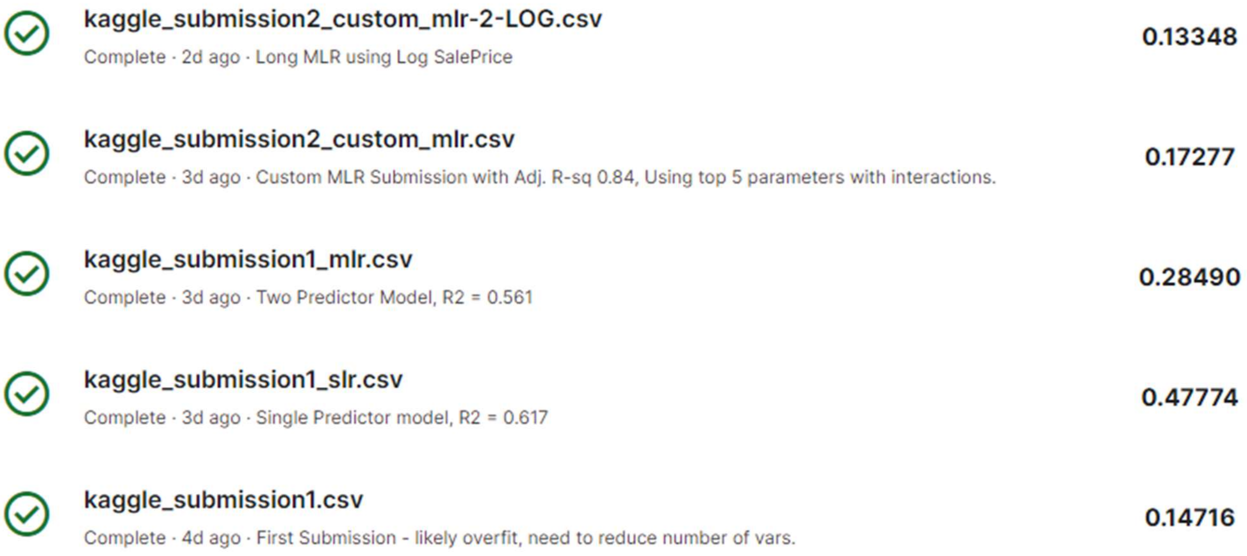


After Transformation/Removal:





Kaggle Results



kaggle_submission2_custom_mlr-2-LOG.csv
Complete · 2d ago · Long MLR using Log SalePrice
0.13348

kaggle_submission2_custom_mlr.csv
Complete · 3d ago · Custom MLR Submission with Adj. R-sq 0.84, Using top 5 parameters with interactions.
0.17277

kaggle_submission1_mlr.csv
Complete · 3d ago · Two Predictor Model, R2 = 0.561
0.28490

kaggle_submission1_slr.csv
Complete · 3d ago · Single Predictor model, R2 = 0.617
0.47774

kaggle_submission1.csv
Complete · 4d ago · First Submission - likely overfit, need to reduce number of vars.
0.14716