MSDS 6372 Project 1:

Building Regression Models to Predict House Prices

Rajat Chandna

Jonathan Marin

Rene Pineda

Samira Zarandioon

June 16, 2018

# Introduction

# When buying a house, we look at a lot of features, its square feet area, number of bedrooms, bathrooms, frontyards & backyards, location, the look of the kitchen, location, etc. As it happens, the price itself is dependent on many factors. This project tries to understand the underlying factors that go into determined the sale price of houses. The goal of this project is to build a predictive model that can identify and capture the variance in the data as accurately as possible in order to use this data to predict future house prices.

# We are using a dataset of houses in Ames, Iowa. This dataset provides many features of a house that would give a better idea of what and how much do each of these features affect the final sale price. The dataset there are 80 explanatory variables describing every aspect of residential homes in Ames, Iowa such as Street, Neighborhood, LotShape, LandSlope, YearBuilt, FullBath, GarageCars, Fireplaces and Pool Quality, etc. for 2,930 homes.

# We are trying to answer 2 questions:

1. What are the important features and factors that impact house prices, and can we build a model that can accurately predict the price of a house? To answer this, we built many multiple variable regression models, compared their relative performance, and chose the model that did a better job at prediction. We also interpret this model and perform hypothesis testing on it.

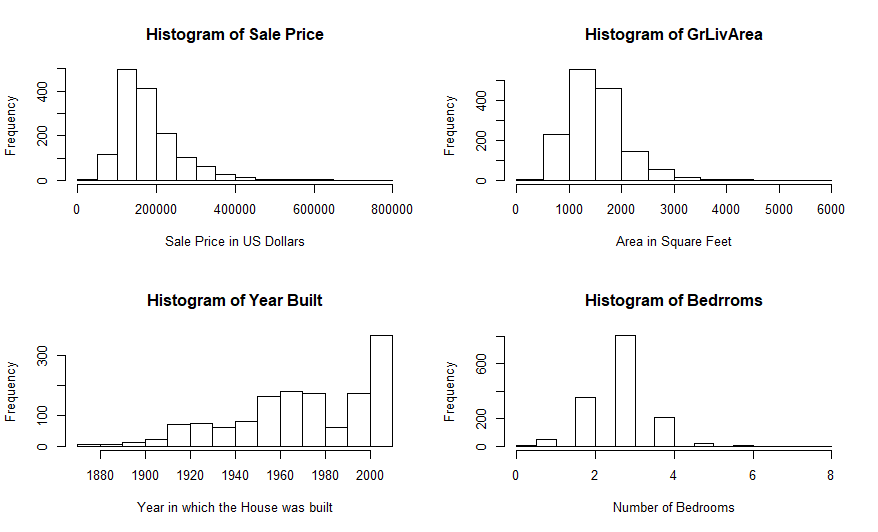
2. If we were to use only two important categorical variables to estimate and predict the price of a house, how can we assess such a model and estimate the influence of each of the two variables in house prices? To answer this, we chose the “Kitchen Quality” and “Neighborhood” variables, and perform two-way ANOVA analysis.

# Data Description

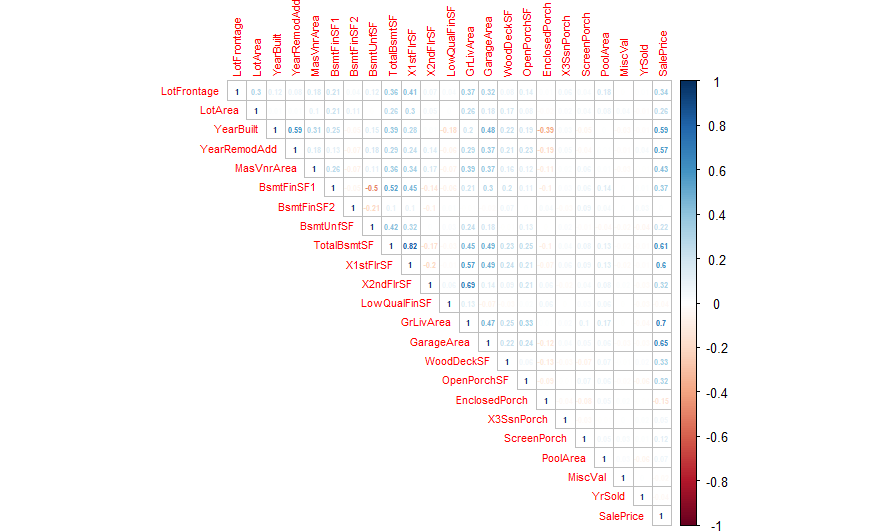
The data set (available at http://www.amstat.org/publications/jse/v19n3/decock/AmesHousing.txt) contains information from the Ames Assessor’s Office used in computing assessed values for individual residential properties sold in Ames, IA from 2006 to 2010. The dataset has 1460 observations of 80 variables related to the house, its features, and surroundings: 23 nominal, 23 ordinal, 14 discrete, and 20 continuous. A summary table of the variables can be found in the Appendix.

# Exploratory Analysis

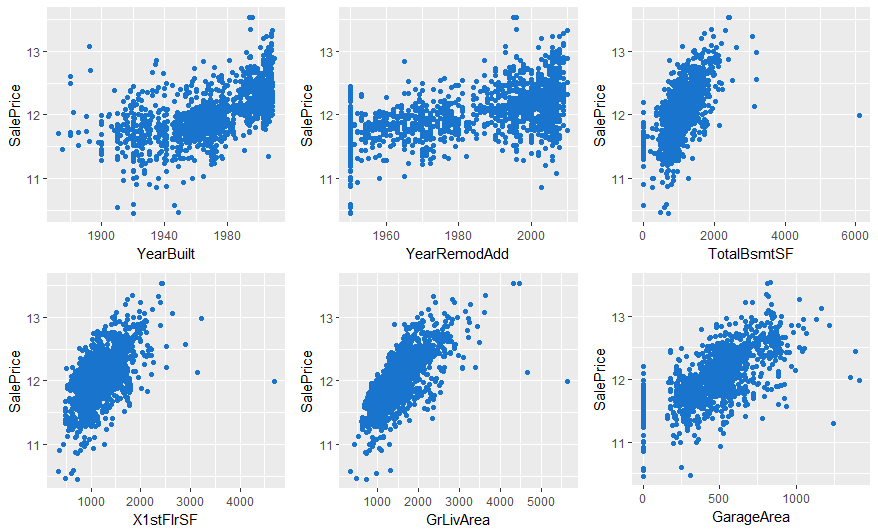
First, we’ll try to understand general features of the houses been sold. Doing some summary statistics and charts, we can draw the following conclusions: the median sale price is $163,000, most of the houses sold have an above the ground area between 1,000 and 2,000 sq ft, a large proportion of houses are fairly recent (built after the year 2000) and 3-bedroom houses are the most popular.



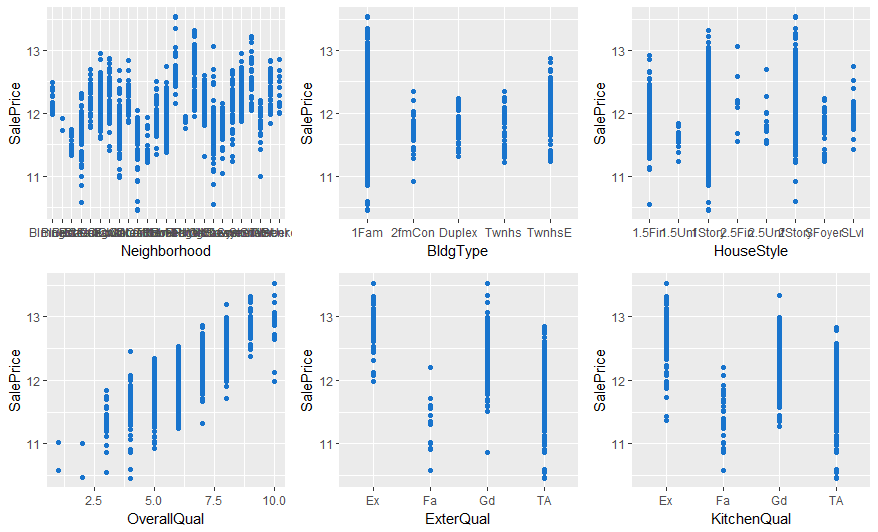
# We also want to understand which numeric variables are strongly correlated with the sale price, and whether multicollinearity may exist. For this, we create a correlation matrix as the one shown below:



# As expected, we find that variables related to the size of the home (TotalBSmtSF, X1stFlrSF, GrLivArea, Garage Area) are strongly, positively correlated with price, and at the same time these variables show moderate to strong correlations with each other. Another important correlation is related to the year that the home was build and remodeled (we also found that 42% of the houses were not remodeled, and that the average number of years before building and adding/remodeling is 13.6)



# We also want to understand which categorical variables can influence Sale Price. We present selected scatterplots below:



# Addressing Objective 1:

## Restatement of Problem

We want to create a model that accurately predicts the sale price of a house in Ames, Iowa, and identify which features of a home are more influential in determining its Sale Price.

The overall strategy will be:

1. Create a model with all features included, and perform residual, leverage and Cook’s D analysis on this model, assessing which observations with high leverage need to be removed

2. Create different models using widely adopted regression techniques (including interaction terms, forward, backwards and stepwise selection, and LASSO)

3. Check the model’s assumptions and perform influential point analysis

4. Assess and compare the models, selecting the one with the highest adjusted R-squared. criteria.

## Model Selection

We created 6 models:

1. An initial model that includes all variables in the dataset

2. A model including the following interaction terms: KitchenQual \* Neighborhood, SaleCondition \* Neighborhood, OverallCond \* OverallQual, GarageYrBlt \* GarageCars \* GarageArea.

2. A LASSO Model

3. Three models using automated variable selection techniques (forward, backwards and stepwise), using the p-value as criteria for including and removing variables.

## Comparing Competing Models

|  |  |
| --- | --- |
| **Predictive Models** | **Adjusted R2** |
| All features included | 0.9698447 |
| Model without interactions | 0.9698447 |
| Model with interactions | 0.9718073 |
| LASSO | 0.8768 |
| Forward | 0.9641 |
| Backward | 0.9648 |
| Stepwise | 0.9648 |

We select the model with the highest Adjusted R-Squared, the backwards selection model.

## Checking Assumptions

|  |  |
| --- | --- |
| Residuals Plot | QQ Plot |
| The residuals plot shows a pattern similar to a random cloud of plots. This provides evidence in favor of the assumption of normality. | The qq-plot confirms the assumption of normality of the residuals, because we don’t see large deviations from the theoretical quantiles of a normal distribution. |
| Studentized Residuals | Leverage and Cook’s D |
| The studentized residuals do not show high values that would create worries about outliers. This reinforces the conclusion that the assumptions of linear regression are met. | A common rule of thumb is that an observation with a value of Cook's D over 1.0 has too much influence. The chart shows that we do not have observations with high leverage (we removed them on a previous step). |

## Parameter Interpretation

The complete results of the selected linear regression model are shown in the appendix. We present below an example of coefficient interpretation:

| **Parameter** | **Estimate** | **95% Confidence Limits** | | **Interpretation** |
| --- | --- | --- | --- | --- |
| **GrLivArea** | 0.000219444 | 0.000197 | 0.000242 | An increase of 1 square feet in the Above Ground Living area increases the price by 0.0219%, or equivalently, increasing the area by 100 feet would increase the price of the house by 2.19%. A 95% confidence level of this estimate is between 0.0197% and 0.0242%. |

## Conclusion

After performing the data processing, removing influential observations, creating 6 models, comparing them and checking the assumptions; based on the adjusted R-squared we selected the backwards selection model as the best predictors of house prices.

## Appendix

Dataset description

|  |  |  |  |
| --- | --- | --- | --- |
| **Discrete Variables** | **Nominal Variables** | **Ordinal** | **Continuous** |
| Order: Observation number | MS SubClass: Identifies the type of dwelling involved in the sale. | Lot Shape: General shape of property | Lot Frontage: Linear feet of street connected to property |
| Year Built : Original construction date | PID: Parcel identification number - can be used with city web site for parcel review. | Utilities: Type of utilities available | Lot Area: Lot size in square feet |
| Year Remod/Add : Remodel date (same as construction date if no remodeling or additions) | Street: Type of road access to property | Land Slope: Slope of property | Mas Vnr Area: Masonry veneer area in square feet |
| Bsmt Full Bath : Basement full bathrooms | Alley: Type of alley access to property | Overall Qual: Rates the overall material and finish of the house | BsmtFin SF 1: Type 1 finished square feet |
| Bsmt Half Bath : Basement half bathrooms | Land Contour: Flatness of the property | Overall Cond: Rates the overall condition of the house | BsmtFin SF 2: Type 2 finished square feet |
| Full Bath : Full bathrooms above grade | Lot Config: Lot configuration | Exter Qual: Evaluates the quality of the material on the exterior | Bsmt Unf SF: Unfinished square feet of basement area |
| Half Bath : Half baths above grade | Neighborhood: Physical locations within Ames city limits (map available) | Exter Cond: Evaluates the present condition of the material on the exterior | Total Bsmt SF: Total square feet of basement area |
| Bedroom : Bedrooms above grade (does NOT include asement bedrooms) | MS Zoning: Identifies the general zoning classification of the sale. | Bsmt Qual: Evaluates the height of the basement | 1st Flr SF: First Floor square feet |
| Kitchen : Kitchens above grade | Condition 1: Proximity to various conditions | Bsmt Cond: Evaluates the general condition of the basement | 2nd Flr SF : Second floor square feet |
| TotRmsAbvGrd: Total rooms above grade (does not include bathrooms) | Condition 2: Proximity to various conditions (if more than one is present) | Bsmt Exposure : Refers to walkout or garden level walls | Low Qual Fin SF: Low quality finished square feet (all floors) |
| Fireplaces : Number of fireplaces | Bldg Type: Type of dwelling | BsmtFin Type 1 : Rating of basement finished area | Gr Liv Area: Above grade (ground) living area square feet |
| Garage Yr Blt : Year garage was built | House Style: Style of dwelling | BsmtFinType 2 : Rating of basement finished area (if multiple types) | Garage Area: Size of garage in square feet |
| Garage Cars : Size of garage in car capacity | Roof Style: Type of roof | HeatingQC: Heating quality and condition | Wood Deck SF: Wood deck area in square feet |
| Mo Sold : Month Sold (MM) | Roof Matl: Roof material | Electrical: Electrical system | Open Porch SF: Open porch area in square feet |
| Yr Sold : Year Sold (YYYY) | Exterior 1: Exterior covering on house | KitchenQual: Kitchen quality |  |
|  | Exterior 2: Exterior covering on house (if more than one material) | Functional: Home functionality (Assume typical unless deductions are warranted) | Enclosed Porch: Enclosed porch area in square feet |
|  | Mas Vnr Type: Masonry veneer type | FireplaceQu: Fireplace quality | 3-Ssn Porch: Three season porch area in square feet |
|  | Foundation: Type of foundation | Garage Finish : Interior finish of the garage | Screen Porch: Screen porch area in square feet |
|  | Heating : Type of heating | Garage Qual: Garage quality | Pool Area: Pool area in square feet |
|  | Central Air: Central air conditioning | Garage Cond: Garage condition | Misc Val: $Value of miscellaneous feature |
|  | Garage Type: Garage location | Paved Drive: Paved driveway | SalePrice: Sale price $$ |
|  | Misc Feature: Miscellaneous feature not covered in other categories | Pool QC: Pool quality |  |
|  | Sale Type: Type of sale | Fence: Fence quality |  |
|  | Sale Condition: Condition of sale |  |  |

Model coefficients and confidence intervals

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Estimate** | **tvalue** |  | **Pr(>|t|)** | **95% Confidence Interval** |  |
| (Intercept) | 5.011129102 | 9.251 | < | 2E-16 | 3.94942 | 6.072841471 |
| MSZoningFV | 0.403631756 | 7.715 |  | 2.36E-14 | 0.30109 | 0.506174298 |
| MSZoningRH | 0.359972682 | 5.948 |  | 3.4635E-09 | 0.24135 | 0.478590892 |
| MSZoningRL | 0.382505402 | 7.824 |  | 1.04E-14 | 0.28668 | 0.47832802 |
| MSZoningRM | 0.277273323 | 5.642 |  | 2.0498E-08 | 0.18095 | 0.373594876 |
| LotArea | 2.7266E-06 | 5.66 |  | 1.8543E-08 | 1.8E-06 | 3.67073E-06 |
| OverallQual | 0.082591441 | 17.41 | < | 2E-16 | 0.07329 | 0.091889034 |
| YearBuilt | 0.000518011 | 2.337 |  | 0.019582 | 8.4E-05 | 0.000952441 |
| YearRemodAdd | 0.002198544 | 9.194 | < | 2E-16 | 0.00173 | 0.002667255 |
| BsmtFinType1BLQ | -0.021113202 | -1.419 |  | 0.156141 | -0.0503 | 0.00805009 |
| BsmtFinType1GLQ | 0.010890701 | 0.847 |  | 0.397342 | -0.0143 | 0.036102721 |
| BsmtFinType1LwQ | -0.032329069 | -1.71 |  | 0.087537 | -0.0694 | 0.00473098 |
| BsmtFinType1None | -0.138579522 | -4.087 |  | 4.6223E-05 | -0.205 | -0.07212836 |
| BsmtFinType1Rec | -0.015689948 | -1.002 |  | 0.31632 | -0.0464 | 0.014987924 |
| BsmtFinType1Unf | -0.065413977 | -4.534 |  | 6.3144E-06 | -0.0937 | -0.03713488 |
| BsmtFinSF1 | -9.8721E-06 | -0.683 |  | 0.494877 | -4E-05 | 1.84679E-05 |
| TotalBsmtSF | 0.000033862 | 1.522 |  | 0.128255 | -1E-05 | 7.74698E-05 |
| CentralAirY | 0.096854817 | 5.499 |  | 4.574E-08 | 0.06233 | 0.131376614 |
| X1stFlrSF | 3.43894E-05 | 1.473 |  | 0.141028 | -1E-05 | 8.01532E-05 |
| GrLivArea | 0.000219444 | 19.15 | < | 2E-16 | 0.0002 | 0.000241903 |
| BsmtFullBath | 0.03550363 | 3.72 |  | 0.000207 | 0.0168 | 0.054208056 |
| FireplaceQuFa | -0.069661998 | -1.735 |  | 0.082941 | -0.1483 | 0.009025929 |
| FireplaceQuGd | -0.031833525 | -0.979 |  | 0.327828 | -0.0956 | 0.03190782 |
| FireplaceQuNone | -0.107902315 | -3.244 |  | 0.001209 | -0.1731 | -0.04270254 |
| FireplaceQuPo | -0.071685694 | -1.59 |  | 0.112051 | -0.16 | 0.016676284 |
| FireplaceQuTA | -0.072748968 | -2.207 |  | 0.027449 | -0.1373 | -0.00815644 |
| GarageTypeAttchd | 0.191318693 | 2.794 |  | 0.005279 | 0.05711 | 0.325524345 |
| GarageTypeBasment | 0.123820629 | 1.631 |  | 0.103153 | -0.025 | 0.272629086 |
| GarageTypeBuiltIn | 0.205515194 | 2.911 |  | 0.003661 | 0.06715 | 0.343881464 |
| GarageTypeCarPort | 0.07644253 | 0.94 |  | 0.347319 | -0.0829 | 0.235810416 |
| GarageTypeDetchd | 0.163567393 | 2.391 |  | 0.016955 | 0.02947 | 0.297667688 |
| GarageTypeNone | 0.134487297 | 1.841 |  | 0.065777 | -0.0087 | 0.277632371 |
| GarageCars | 0.062977127 | 5.388 |  | 8.4282E-08 | 0.04007 | 0.085887464 |
| GarageArea | 6.91836E-05 | 1.759 |  | 0.078739 | -8E-06 | 0.000146255 |

R Code (Problem #1):

---

title: "Project1"

author: "Jonathan Marin, Rajat Chandna, Rene Alvarenga, Samira Zarandioon"

date: "June 11, 2018"

output:

pdf\_document: default

html\_document: default

---

```{r setup, echo=FALSE, include = FALSE}

# install the required packages if needed

list.of.packages <- c("sqldf", "glmnet", "gfortran", "rgl", "CVST", "igraph", "recipes", "ggplot2", "caret", "forcats", "olsrr", "tidyr", "corrplot", "parallel", "doParallel")

new.packages <- list.of.packages[!(list.of.packages %in% installed.packages()[,"Package"])]

if(length(new.packages)) install.packages(new.packages, repos="http://cran.r-project.org")

```

```{r echo=FALSE, include = FALSE}

# load the required libraries

library(sqldf) # Used for manipulating the data frames using SQL

library(glmnet) # Used for...

library(caret) # Used for...

library(forcats) # Used for...

library(olsrr) # Used for creating the foward

library(MASS) # Used for backward, and stepwise models

library(tidyr) #Used for creating some of the plots

library(ggplot2) #Used for creating some of the plots

library(corrplot) #Used to create the correlation matrix

library(parallel) #Using to assign more cores and allow parallel processing

library(doParallel) #Using to assign more cores and allow parallel processing

```

```{r echo=FALSE, include = FALSE}

# Set seed for reproducibility

set.seed(0)

#Load the data

train <- read.csv("train.csv")

test <- read.csv("test.csv")

# Generate histogram of Sale Price

hist(train$SalePrice, main = "Historgram of Sale Price", xlab = "Sale Price in US$")

#Inspect the data

dim(train)

str(train)

```

```{r echo=FALSE, include = FALSE}

# Log transform the sales price

# Perform natural logarithm on response variable.

train$SalePrice <- log(train$SalePrice)

# Generate histogram of log of Sale Price

hist(train$SalePrice, main = "Historgram of natural log of Sale Price", xlab = "Natural log of Sale Price in US$")

```

```{r Data Preparation, echo=FALSE, include = FALSE, warning = FALSE}

#This shows us what is null

sapply(train, function(x) sum(is.na(x)))

#upon inspecton, the following variables have missing information:

#LotFrontage, Alley, MasVnrType, MasVnrArea, BsmtQual, BstCond, BsmtExposure, BsmtFinType1, BsmtFinType2,

#FireplaceQu, GarageType, GarageYrBlt, GarageFinish, GarageCond, Fence, MiscFeature, PoolQC

#Adding sale price to the test set for combining (using impossible value of -1 to distinguish between train and test data)

test$SalePrice <- -1

#Join the train and test sets for preprocessing

data <- rbind(train, test)

#Converting values to NULL in the following variables tha will not be used

data$MiscFeature <- NULL # Missing value in 96.4% of observations

data$Alley <- NULL # Missing value in 93.2% of observations

data$PoolQC <- NULL # Missing value in 99.7% of observations

#Variables with values as NA

NAFeatures = names(which(colSums(is.na(data))>0))

# use median imputation to handle missing LotFrontage data

data$LotFrontage[is.na(data$LotFrontage)] <- median(train$LotFrontage, na.rm = TRUE) #69

#If NA for GarageYrBlt, then set to YearBuilt of house

data$GarageYrBlt[is.na(data$GarageYrBlt)] <- as.integer(data$YearBuilt)

#Create vectors of Variables with missing observations and variables with zero values

missingObs = c("MSZoning", "MasVnrType", "Utilities", "Exterior1st", "Exterior2nd", "SaleType")

effZero = c("LotFrontage", "MasVnrArea", "BsmtFinSF1", "BsmtFinSF2", "BsmtUnfSF", "TotalBsmtSF","GarageCars", "GarageArea", "BsmtFullBath", "BsmtHalfBath")

# Get Effectively Absent category by excluding other categories from varsWithNA

effAbsent = NAFeatures[!NAFeatures %in% missingObs]

effAbsent = effAbsent[!effAbsent %in% effZero]

effAbsent = effAbsent[!effAbsent %in% c("Functional")]

# Function for replacing NAs in nominal and ordinal variables

replaceNAfactor = function(data.col, factorString){

char.col <- as.character(data.col)

char.col[which(is.na(data.col))] <- factorString

as.factor(char.col)

}

# Replace NAs with None in Effectively Absent category

for (i in 1:ncol(data)){

if(names(data[i]) %in% effAbsent){

data[,i] <- replaceNAfactor(data[,i], "None")}

}

# Replace NAs with MissingObs in Missing Observations category

for (i in 1:ncol(data)){

if(names(data[i]) %in% missingObs){

data[,i] <- replaceNAfactor(data[,i], "MissingObs")}

}

# Replace NAs with 0 in Effectively Zero category

for (i in 1:ncol(data)){

if(names(data[i]) %in% effZero)

data[is.na(data[,i]),i] <- 0

}

data$Functional <- replaceNAfactor(data$Functional, "Typ")

#Checking that we corrected for NAs

sapply(data, function(x) sum(is.na(x)))

#Resplitting train and test

xtrain <- as.data.frame(sqldf("select \* from data where SalePrice <> -1"))

ytrain <- sqldf("select SalePrice from data where SalePrice <> -1")

names(ytrain) <- c("SalePrice")

xtrain$SalePrice <- NULL

xtest <- sqldf("select \* from data where SalePrice = -1")

xtest$SalePrice <- NULL

```

```{r Corr Matrix and Scatterplots, echo=FALSE, include = FALSE}

#Create some histograms

par(mfrow=c(2,2))

hist(train$SalePrice, main = "Historgram of natural log of Sale Price", xlab = "Natural log of Sale Price in US$")

hist(train$GrLivArea, main = "Histogram of GrLivArea", xlab = "Area in Square Feet")

hist(train$YearBuilt, main = "Histogram of Year Built", xlab = "Year in which the House was built")

hist(train$BedroomAbvGr, main = "Histogram of Bedrroms", xlab = "Number of Bedrooms")

dev.off()

#Correlation matrix

#The following variables will be included in the matrix: Lot Frontage, Lot Area, Year Built, Year Remod/Add,

#Mas Vnr Area, Bsmt Fin SF 1, BsmtFin SF 2, Bsmt Unf SF, Total Bsmt SF, 1st Flr SF, 2nd Flr SF, Low Qual Fin SF,

#Gr Liv Area, Garage Area, Wood Deck Sf, Open Porch SF, Enclosed Porch, 3-Ssn Porch, Screen Porch, Pool Area,

#Misc Val, Yr Sold, Sale Price

#Choose only these variables:

numericvars <- xtrain[,c(4,5,19,20,26,34,36,37,38,43,44,45,46,62,66,67,68,69,70,71,73,75)]

nonnumericvars <- xtrain[,-c(4,5,19,20,26,34,36,37,38,43,44,45,46,62,66,67,68,69,70,71,73,75)]

corrvar <- cbind(numericvars,ytrain)

#Create the correlation matrix

N <- cor(corrvar, use = "complete.obs")

corrplot(N, method = "number", number.cex = 0.5, tl.cex = 0.7)

#Create the scatterplots

#Numeric variables:

p1 <- ggplot(data = corrvar) +

geom\_point(mapping = aes(x = corrvar[,3], y = SalePrice), color = "dodgerblue3" ) +

labs(x = (names(corrvar[3])))

p2 <- ggplot(data = corrvar) +

geom\_point(mapping = aes(x = corrvar[,4], y = SalePrice), color = "dodgerblue3" ) +

labs(x = (names(corrvar[4])))

p3 <- ggplot(data = corrvar) +

geom\_point(mapping = aes(x = corrvar[,9], y = SalePrice), color = "dodgerblue3" ) +

labs(x = (names(corrvar[9])))

p4 <- ggplot(data = corrvar) +

geom\_point(mapping = aes(x = corrvar[,10], y = SalePrice), color = "dodgerblue3" ) +

labs(x = (names(corrvar[10])))

p5 <- ggplot(data = corrvar) +

geom\_point(mapping = aes(x = corrvar[,13], y = SalePrice), color = "dodgerblue3" ) +

labs(x = (names(corrvar[13])))

p6 <- ggplot(data = corrvar) +

geom\_point(mapping = aes(x = corrvar[,14], y = SalePrice), color = "dodgerblue3" ) +

labs(x = (names(corrvar[14])))

gridExtra::grid.arrange(p1, p2,p3, p4,p5, p6, nrow = 2)

#Nonnumeric variables

nonnumericvars$SalePrice <- corrvar$SalePrice

p7 <- ggplot(data = nonnumericvars) +

geom\_point(mapping = aes(x = nonnumericvars[,10], y = SalePrice), color = "dodgerblue3" ) +

labs(x = (names(nonnumericvars[10])))

p8 <- ggplot(data = nonnumericvars) +

geom\_point(mapping = aes(x = nonnumericvars[,13], y = SalePrice), color = "dodgerblue3" ) +

labs(x = (names(nonnumericvars[13])))

p9 <- ggplot(data = nonnumericvars) +

geom\_point(mapping = aes(x = nonnumericvars[,14], y = SalePrice), color = "dodgerblue3" ) +

labs(x = (names(nonnumericvars[14])))

p10 <- ggplot(data = nonnumericvars) +

geom\_point(mapping = aes(x = nonnumericvars[,15], y = SalePrice), color = "dodgerblue3" ) +

labs(x = (names(nonnumericvars[15])))

p11 <- ggplot(data = nonnumericvars) +

geom\_point(mapping = aes(x = nonnumericvars[,22], y = SalePrice), color = "dodgerblue3" ) +

labs(x = (names(nonnumericvars[22])))

p12 <- ggplot(data = nonnumericvars) +

geom\_point(mapping = aes(x = nonnumericvars[,40], y = SalePrice), color = "dodgerblue3" ) +

labs(x = (names(nonnumericvars[40])))

gridExtra::grid.arrange(p7, p8,p9, p10,p11, p12, nrow = 2)

```

##Regression Models

So far we've created four regression models

\* Initial model with all variables: modelfit1

\* LASSO model

\* Forward selection model using p-values as criteria: modelfoward

\* Backward selection model using p-values as criteria: modelbackward

\* Stepwise selection model using p-values as criteria: modelstepwise

=======

```{r Regression models, echo=FALSE, include = FALSE, warning= FALSE}

#Running an initial regression model with all the data

modelfit\_all\_fields <- lm(ytrain$SalePrice ~ ., data = xtrain[,-1])

plot(modelfit\_all\_fields)

#Adjusted R-squared: 0.9337166

print("modelfit\_all\_fields: ")

summary(modelfit\_all\_fields)$adj.r.squared

# Testing model interactions from problem 2 analysis

modelfit\_without\_interactions <- lm(ytrain$SalePrice ~ MSSubClass + MSZoning + LotFrontage + LotArea + Street + LotShape + LandContour + Utilities + LotConfig + LandSlope + Neighborhood + Condition1 + Condition2 + BldgType + HouseStyle + OverallQual + OverallCond + YearBuilt + YearRemodAdd + RoofStyle + RoofMatl + Exterior1st + Exterior2nd + MasVnrType + MasVnrArea + ExterQual + ExterCond + Foundation + BsmtQual + BsmtCond + BsmtExposure + BsmtFinType1 + BsmtFinSF1 + BsmtFinType2 + BsmtFinSF2 + BsmtUnfSF + TotalBsmtSF + Heating + HeatingQC + CentralAir + Electrical + xtrain$X1stFlrSF + xtrain$X2ndFlrSF + LowQualFinSF + GrLivArea + BsmtFullBath + BsmtHalfBath + FullBath + HalfBath + BedroomAbvGr + KitchenAbvGr + KitchenQual + TotRmsAbvGrd + Functional + Fireplaces + FireplaceQu + GarageType + GarageYrBlt + GarageFinish + GarageCars + GarageArea + GarageQual + GarageCond + PavedDrive + WoodDeckSF + OpenPorchSF + EnclosedPorch + xtrain$X3SsnPorch + ScreenPorch + PoolArea + Fence + MiscVal + MoSold + YrSold + SaleType + SaleCondition, data = xtrain[,-1])

#Adjusted R-squared: 0.9337166

print("modelfit\_without\_interactions: ")

summary(modelfit\_without\_interactions)$adj.r.squared

modelfit\_with\_interactions <- lm(ytrain$SalePrice ~ MSSubClass + MSZoning + LotFrontage + LotArea + Street + LotShape + LandContour + Utilities + LotConfig + LandSlope + Neighborhood + Condition1 + Condition2 + BldgType + HouseStyle + OverallQual + OverallCond + YearBuilt + YearRemodAdd + RoofStyle + RoofMatl + Exterior1st + Exterior2nd + MasVnrType + MasVnrArea + ExterQual + ExterCond + Foundation + BsmtQual + BsmtCond + BsmtExposure + BsmtFinType1 + BsmtFinSF1 + BsmtFinType2 + BsmtFinSF2 + BsmtUnfSF + TotalBsmtSF + Heating + HeatingQC + CentralAir + Electrical + xtrain$X1stFlrSF + xtrain$X2ndFlrSF + LowQualFinSF + GrLivArea + BsmtFullBath + BsmtHalfBath + FullBath + HalfBath + BedroomAbvGr + KitchenAbvGr + KitchenQual + TotRmsAbvGrd + Functional + Fireplaces + FireplaceQu + GarageType + GarageYrBlt + GarageFinish + GarageCars + GarageArea + GarageQual + GarageCond + PavedDrive + WoodDeckSF + OpenPorchSF + EnclosedPorch + xtrain$X3SsnPorch + ScreenPorch + PoolArea + Fence + MiscVal + MoSold + YrSold + SaleType + SaleCondition +

# Iteraction terms

KitchenQual \* Neighborhood +

SaleCondition \* Neighborhood +

OverallCond \* OverallQual +

GarageYrBlt \* GarageCars \* GarageArea,

data = xtrain[,-1])

# Adjusted R-squared: 0.9469788

print("modelfit\_with\_interactions: ")

summary(modelfit\_with\_interactions)$adj.r.squared

```

```{r, echo = FALSE, include = FALSE}

#Remove High Leverage Points and Cooks D and create new train set from this

cooksd <- cooks.distance(modelfit\_all\_fields)

sample\_size <- nrow(xtrain)

plot(cooksd, pch="\*", cex=2, main="Influential Obs by Cooks distance") # plot cook's distance

abline(h = 4/sample\_size, col="red") # add cutoff line

text(x=1:length(cooksd)+1, y=cooksd, labels=ifelse(cooksd>4/sample\_size, names(cooksd),""), col="red") # add labels

#Removing outliers with high cooks d

influential <- (as.numeric(names(cooksd)[(cooksd > (4/sample\_size))]))

influential <- influential[!is.na(influential)]

xtrain <- as.data.frame(xtrain[-influential,])

ytrain <- as.data.frame(ytrain[-influential,])

names(ytrain) <- c("SalePrice")

```

```{r Regression models2, echo=FALSE, include = FALSE}

# retraining the models after removing the influential data points (imporoves the adjusted R^2)

#Running an initial regression model with all the data

modelfit\_all\_fields\_no\_influential <- lm(ytrain$SalePrice ~ ., data = xtrain[,-1])

#Adjusted R-squared: increased from 0.9337166 to 0.968239

print("modelfit\_all\_fields\_no\_influential: ")

summary(modelfit\_all\_fields\_no\_influential)$adj.r.squared

# Testing model interactions from problem 2 analysis

modelfit\_without\_interactions\_no\_influential <- lm(ytrain$SalePrice ~ MSSubClass + MSZoning + LotFrontage + LotArea + Street + LotShape + LandContour + Utilities + LotConfig + LandSlope + Neighborhood + Condition1 + Condition2 + BldgType + HouseStyle + OverallQual + OverallCond + YearBuilt + YearRemodAdd + RoofStyle + RoofMatl + Exterior1st + Exterior2nd + MasVnrType + MasVnrArea + ExterQual + ExterCond + Foundation + BsmtQual + BsmtCond + BsmtExposure + BsmtFinType1 + BsmtFinSF1 + BsmtFinType2 + BsmtFinSF2 + BsmtUnfSF + TotalBsmtSF + Heating + HeatingQC + CentralAir + Electrical + xtrain$X1stFlrSF + xtrain$X2ndFlrSF + LowQualFinSF + GrLivArea + BsmtFullBath + BsmtHalfBath + FullBath + HalfBath + BedroomAbvGr + KitchenAbvGr + KitchenQual + TotRmsAbvGrd + Functional + Fireplaces + FireplaceQu + GarageType + GarageYrBlt + GarageFinish + GarageCars + GarageArea + GarageQual + GarageCond + PavedDrive + WoodDeckSF + OpenPorchSF + EnclosedPorch + xtrain$X3SsnPorch + ScreenPorch + PoolArea + Fence + MiscVal + MoSold + YrSold + SaleType + SaleCondition, data = xtrain[,-1])

#Adjusted R-squared: increased from 0.9337166 to 0.968239

print("modelfit\_without\_interactions: ")

summary(modelfit\_without\_interactions\_no\_influential)$adj.r.squared

modelfit\_with\_interactions\_no\_influential <- lm(ytrain$SalePrice ~ MSSubClass + MSZoning + LotFrontage + LotArea + Street + LotShape + LandContour + Utilities + LotConfig + LandSlope + Neighborhood + Condition1 + Condition2 + BldgType + HouseStyle + OverallQual + OverallCond + YearBuilt + YearRemodAdd + RoofStyle + RoofMatl + Exterior1st + Exterior2nd + MasVnrType + MasVnrArea + ExterQual + ExterCond + Foundation + BsmtQual + BsmtCond + BsmtExposure + BsmtFinType1 + BsmtFinSF1 + BsmtFinType2 + BsmtFinSF2 + BsmtUnfSF + TotalBsmtSF + Heating + HeatingQC + CentralAir + Electrical + xtrain$X1stFlrSF + xtrain$X2ndFlrSF + LowQualFinSF + GrLivArea + BsmtFullBath + BsmtHalfBath + FullBath + HalfBath + BedroomAbvGr + KitchenAbvGr + KitchenQual + TotRmsAbvGrd + Functional + Fireplaces + FireplaceQu + GarageType + GarageYrBlt + GarageFinish + GarageCars + GarageArea + GarageQual + GarageCond + PavedDrive + WoodDeckSF + OpenPorchSF + EnclosedPorch + xtrain$X3SsnPorch + ScreenPorch + PoolArea + Fence + MiscVal + MoSold + YrSold + SaleType + SaleCondition +

# Iteraction terms

KitchenQual \* Neighborhood +

SaleCondition \* Neighborhood +

OverallCond \* OverallQual +

GarageYrBlt \* GarageCars \* GarageArea,

data = xtrain[,-1])

# Adjusted R-squared: increased from 0.9469788 to 0.9706955

print("modelfit\_with\_interactions\_no\_influential: ")

summary(modelfit\_with\_interactions\_no\_influential)$adj.r.squared

```

```{r, echo = FALSE, include = FALSE}

library(caret)

library(leaps)

library(parallel)

library(doParallel)

cluster <- makeCluster(detectCores() - 1) # convention to leave 1 core for OS

registerDoParallel(cluster)

train <- cbind(xtrain[,-1], ytrain)

trainControl = trainControl(method = "cv", number = 10, verboseIter = FALSE, allowParallel = TRUE)

model\_all\_fields = train(SalePrice ~ .,

data = train,

method = "lm",

trControl = trainControl)

model\_without\_interactions = train(SalePrice ~ MSSubClass + MSZoning + LotFrontage + LotArea + Street + LotShape + LandContour + Utilities + LotConfig + LandSlope + Neighborhood + Condition1 + Condition2 + BldgType + HouseStyle + OverallQual + OverallCond + YearBuilt + YearRemodAdd + RoofStyle + RoofMatl + Exterior1st + Exterior2nd + MasVnrType + MasVnrArea + ExterQual + ExterCond + Foundation + BsmtQual + BsmtCond + BsmtExposure + BsmtFinType1 + BsmtFinSF1 + BsmtFinType2 + BsmtFinSF2 + BsmtUnfSF + TotalBsmtSF + Heating + HeatingQC + CentralAir + Electrical + X1stFlrSF + X2ndFlrSF + LowQualFinSF + GrLivArea + BsmtFullBath + BsmtHalfBath + FullBath + HalfBath + BedroomAbvGr + KitchenAbvGr + KitchenQual + TotRmsAbvGrd + Functional + Fireplaces + FireplaceQu + GarageType + GarageYrBlt + GarageFinish + GarageCars + GarageArea + GarageQual + GarageCond + PavedDrive + WoodDeckSF + OpenPorchSF + EnclosedPorch + X3SsnPorch + ScreenPorch + PoolArea + Fence + MiscVal + MoSold + YrSold + SaleType + SaleCondition,

data = train,

method = "lm",

trControl = trainControl)

model\_with\_interactions = train(SalePrice ~ MSSubClass + MSZoning + LotFrontage + LotArea + Street + LotShape + LandContour + Utilities + LotConfig + LandSlope + Neighborhood + Condition1 + Condition2 + BldgType + HouseStyle + OverallQual + OverallCond + YearBuilt + YearRemodAdd + RoofStyle + RoofMatl + Exterior1st + Exterior2nd + MasVnrType + MasVnrArea + ExterQual + ExterCond + Foundation + BsmtQual + BsmtCond + BsmtExposure + BsmtFinType1 + BsmtFinSF1 + BsmtFinType2 + BsmtFinSF2 + BsmtUnfSF + TotalBsmtSF + Heating + HeatingQC + CentralAir + Electrical + X1stFlrSF + X2ndFlrSF + LowQualFinSF + GrLivArea + BsmtFullBath + BsmtHalfBath + FullBath + HalfBath + BedroomAbvGr + KitchenAbvGr + KitchenQual + TotRmsAbvGrd + Functional + Fireplaces + FireplaceQu + GarageType + GarageYrBlt + GarageFinish + GarageCars + GarageArea + GarageQual + GarageCond + PavedDrive + WoodDeckSF + OpenPorchSF + EnclosedPorch + X3SsnPorch + ScreenPorch + PoolArea + Fence + MiscVal + MoSold + YrSold + SaleType + SaleCondition +

# Iteraction terms

KitchenQual \* Neighborhood +

SaleCondition \* Neighborhood,

data = train,

method = "lm",

trControl = trainControl)

# auto feature selection

model\_forward\_selection = train(SalePrice ~ .,

data = train,

method = "leapForward",

tuneGrid = data.frame(nvmax = seq(from = 200, to = 300, by = 10)),

trControl = trainControl)

model\_backward\_selection = train(SalePrice ~ .,

data = train,

method = "leapBackward",

tuneGrid = data.frame(nvmax = seq(from = 200, to = 300, by = 10)),

trControl = trainControl)

#model\_stepwise\_selection = train(SalePrice ~ .,

# data = train,

# method = "leapSeq",

# tuneGrid = data.frame(nvmax = seq(from = 200, to = 300, by = 10)),

# trControl = trainControl)

```

```{r}

model\_random\_forest = train(SalePrice ~ .,

data = train,

tuneLength = 1,

method = "ranger",

importance = 'impurity',

trControl = trainControl)

```

```{r}

options(scipen=999)

models\_list = list(model\_all\_fields = model\_all\_fields,

model\_without\_interactions = model\_without\_interactions,

model\_with\_interactions = model\_with\_interactions,

model\_forward\_selection = model\_forward\_selection,

model\_backward\_selection = model\_backward\_selection)

## model\_random\_forest = model\_random\_forest)

# model\_stepwise\_selection = model\_stepwise\_selection )

resamples = resamples(models\_list)

summary(resamples)

bwplot(resamples, metric = "RMSE")

bwplot(resamples, metric = "Rsquared")

bwplot(resamples, metric = "MAE")

```

```{r}

parallelplot(resamples, metric = "RMSE")

parallelplot(resamples, metric = "Rsquared")

parallelplot(resamples, metric = "MAE")

```

```{r, echo = FALSE, include = FALSE}

`%ni%` <- Negate(`%in%`)

x = model.matrix(ytrain$SalePrice~., data = xtrain[,-1])

y = (ytrain$SalePrice)

cv.out <- cv.glmnet(x,y,alpha = 1)

plot(cv.out)

bestlambda <- cv.out$lambda.1se

c<- coef(cv.out, s = bestlambda, extract = TRUE)

inds <- which(c!=0)

variables <- row.names(c)[inds]

variables <- variables[variables %ni% ('Intercept')]

variables

lasso.model <- lm(train$SalePrice ~ MSZoning + LotArea + OverallQual + YearBuilt + YearRemodAdd+

YearRemodAdd + BsmtFinType1 + BsmtFinSF1 + TotalBsmtSF + CentralAir +

X1stFlrSF + GrLivArea + BsmtFullBath + FireplaceQu + GarageType + GarageCars + GarageArea, data = xtrain[,-1])

summary(lasso.model)

library(MASS)

#Forward selection model #JM, below code is giving me errors: Could not find function ols\_step\_forward\_p

#model <- lm(ytrain$SalePrice ~ ., data = xtrain[,-1]) #added [,-1] part so that we remove the id from model

#modelforward <- tryCatch({

# ols\_step\_forward(model)

#}, warning = function(w) {

# ols\_step\_forward\_p(model)

#}, error = function(e) {

# ols\_step\_forward\_p(model)

#})

#modelforward

#summary(modelforward)

#steps <- modelforward$steps

#modelforward$adjr[steps]

#modelforward$rmse[steps]

#modelforward$predictors

#plot(modelforward)

forward.mass <- stepAIC(modelfit\_with\_interactions, direction = "forward", trace= FALSE, steps = 100)

summary(forward.mass)

#Backward Selection model #Taking 18 hours to run so far (JM) Corrected code, but taking to long.

#Code does not work as desired

#modelbackward <- ols\_step\_backward(model)

#modelbackward

#jm - I usually use the MASS package for feature selection and this worked.

backward.mass <- stepAIC(modelfit\_with\_interactions, direction = "backward", trace = FALSE, steps = 100)

summary(backward.mass)

stepwise.mass <- stepAIC(modelfit\_with\_interactions, direction = "both", trace = FALSE, steps = 500)

summary(stepwise.mass)

#summary(modelbackward)

#steps <- modelbackward$steps

#modelbackward$adjr[steps]

#modelbackward$rmse[steps]

#modelbackward$predictors

#plot(modelbackward)

#Stepwise Selection model

#modelstepwise <- ols\_step\_both(model)

#modelstepwise

#plot(modelstepwise)

models\_list = list(model\_all\_fields = model\_all\_fields,

model\_without\_interactions = model\_without\_interactions,

model\_with\_interactions = model\_with\_interactions,

model\_forward\_selection = model\_forward\_selection,

model\_backward\_selection = model\_backward\_selection,

model\_stepwise\_selection = model\_stepwise\_selection,

forward.mass = forward.mass,

backward.mass = backward.mass,

stepwise.mass = stepwise.mass)

resamples = resamples(models\_list)

summary(resamples)

```

#MSDS 6371 Project 1

## Setup and Loading packagges

## Data cleaning and preparation

## Some exploratory Analysis

##Regression Models

So far we've created four regression models

\* Initial model with all variables: modelfit1

\* LASSO model

\* Forward selection model using p-values as criteria: modelfoward

\* Backward selection model using p-values as criteria: modelbackward

\* Stepwise selection model using p-values as criteria: modelstepwise

##Outlier Analysis

#Lasso Model

```{r}

options(scipen=999)

bwplot(resamples, metric = "RMSE")

bwplot(resamples, metric = "Rsquared")

bwplot(resamples, metric = "MAE")

summary(lasso.model)

```

##Pending tasks

\* Run several models from all the selection processes. - Rene

\* Create table comparing comparing AIC or R^2 from all models, and make decision - Rene

\* Enter interaction terms (ex: GarageYrBuilt \* GarageCars \* GarageArea ) -- Samira (DONE!)

\*\* I added the following interaction terms, which improved adjusted $R^2$ from 0.9179 to 0.9313

\*\*\* KitchenQual \* Neighborhood

\*\*\* SaleCondition \* Neighborhood

\*\*\* OverallCond \* OverallQual

\*\*\* GarageYrBlt \* GarageCars \* GarageArea

\*\* (Samira) Re-trained the linear models after removing the influenctial datapoints and it increased the adjusted R-squared for the model with interactions form 0.9313 to 0.9611. This means adding the three interaction terms along with removing influential points improves the adjusted R-squared from 0.9179 to 0.9611.

\* (Samira): Added logic for model performance comparision between different feature selection strategies.

\* Clean up and prep final doc within the rmarkdown? - Samira?

\* Perform the 2-way ANOVA analysis for the second part of the project assignment --Rajat

SAS Code (Problem #2)

FILENAME REFFILE '/folders/myfolders/CleanedDataForANOVA.csv';

PROC IMPORT DATAFILE=REFFILE

DBMS=CSV

OUT=WORK.KaggData replace;

GETNAMES=YES;

RUN;

proc means data=KaggData n mean max min range std fw=8;

class KitchenQual Neighborhood;

var SalePrice;

output out=meansout mean=mean std=std;

title 'Summary of Sales Prices';

run;

/\* Remove the spurious obervations \*/

data summarystats;

set meansout;

if \_TYPE\_=0 then delete;

if \_TYPE\_=1 then delete;

if \_TYPE\_=2 then delete;

run;

/\* Prepare the data to be plotted, calc upper and lower ends of standard dev bars \*/

data plottingdata;

set summarystats;

lower=mean - std;

upper=mean + std;

run;

/\* Sort the data by Neighborhood \*/

proc sort data=plottingdata;

by Neighborhood;

run;

proc print data=plottingdata;

run;

/\* Plotting the data \*/

proc sgplot data=plottingdata;

scatter x=Neighborhood y=mean / group=KitchenQual yerrorlower=lower

yerrorupper=upper

markerattrs=(symbol=CircleFilled) ;

series x=Neighborhood y=mean / group=KitchenQual ;

title1 'Plot Means with Standard Deviations Bars from Calculated Data';

label mean='Average Sales Price';

run;

/\* Above Plot shows that spread is higher at some higher values of mean, hence

Non Constant variance could be present( points at site No Ridge). Residual plot would clarify further \*/

proc glm data=KaggData plots=(DIAGNOSTICS RESIDUALS);

class KitchenQual Neighborhood;

model SalePrice = KitchenQual Neighborhood KitchenQual\*Neighborhood;

run;

/\* Residual Plots show funnel like pattern and also skewed residuals.

Log Transformation could help in rectifying such situation \*/

/\* Performing Log Transformation and Model Assumption Validation \*/

data kaggdata;

set kaggdata;

LogSalePrice = Log(SalePrice); /\* In \*/

run;

/\* To check, for outliers based upon high GrLiving Area as they might not

be representative of all other houses in the area\*/

proc sgscatter data=Work.Kaggdata;

plot SalePrice\*GrLivArea;

run;

/\* Houses with Gr Living Area > 4000 are not representative of other houses in data set

hence removing there outlier values \*/

data kaggdata;

set kaggdata;

if GrLivArea < 4000;

run;

proc means data=KaggData n mean max min range std fw=8;

class KitchenQual Neighborhood;

var LogSalePrice;

output out=meansoutlog mean=mean std=std;

title 'Summary of Log Sales Prices';

run;

/\* Remove the spurious obervations \*/

data summarystatsafterlogT;

set meansoutlog;

if \_TYPE\_=0 then delete;

if \_TYPE\_=1 then delete;

if \_TYPE\_=2 then delete;

run;

/\* Prepare the data to be plotted, calc upper and lower ends of standard dev bars \*/

data plottingdataafterlogT;

set summarystatsafterlogT;

lower=mean - std;

upper=mean + std;

run;

/\* Sort the data by Neighborhood \*/

proc sort data=plottingdataafterlogT;

by Neighborhood;

run;

proc sgplot data=plottingdataafterlogT;

scatter x=Neighborhood y=mean / group=KitchenQual yerrorlower=lower

yerrorupper=upper

markerattrs=(symbol=CircleFilled) ;

series x=Neighborhood y=mean / group=KitchenQual ;

title1 'Plot Means with Standard Deviations Bars from Calculated Data';

label mean='Average Log Sales Price';

run;

/\* Running the Model, Using proc GLM for obtaining Type 3 SS table and R sq(effect size)

and then using Proc Mixed to obtain formatted Multiple Comparisons table, if necessary\*/

proc glm data=KaggData;

class KitchenQual Neighborhood;

model LogSalePrice = KitchenQual Neighborhood KitchenQual\*Neighborhood;

run;

/\* Storing Comparison Table to a dataset so that only statistically

significant differences can be extracted \*/

ods output diffs=ComparisonData;

proc mixed data=KaggData plots=RESIDUALPANEL;

class KitchenQual Neighborhood;

model LogSalePrice = KitchenQual Neighborhood KitchenQual\*Neighborhood;

lsmeans KitchenQual\*Neighborhood/pdiff diff cl adjust=tukey;

run;

ods output on;

ods exclude none;

/\* Find Same site Diff in Kitchen Qual on Sale Price \*/

/\* This further bolsters that interaction is important as we see for some sites that

differences in Kitchen Qual on Price are not significant but for few sites, diff are highly significant \*/

data CompareSameSite\_KitchenQual;

set ComparisonData;

if Neighborhood = \_Neighborhood;

run;

/\* To convert log back to normal scale. Interpretation would be Media Y/Median X = e ^ estimate\*/

data CompareSameSite\_KitchenQual;

set CompareSameSite\_KitchenQual;

Estimate\_NormalScale = exp(Estimate);

UpperCI\_NormalScale = exp(AdjUpper);

LowerCI\_NormalScale = exp(AdjLower);

run;

proc sort data=CompareSameSite\_KitchenQual;

by AdjP;

run;

proc print data=CompareSameSite\_KitchenQual;

var KitchenQual Neighborhood \_KitchenQual \_Neighborhood Adjp Estimate\_NormalScale UpperCI\_NormalScale LowerCI\_NormalScale;

run;

/\* To find all ab kind off means that are statistically significant \*/

/\* 863 off 2485 pairs are statistically significant, interaction is important one and complex in nature as

nearly 25% of pairs differ \*/

data StatisticallySigDiffs;

set ComparisonData;

Estimate\_NormalScale = exp(Estimate);

UpperCI\_NormalScale = exp(AdjUpper);

LowerCI\_NormalScale = exp(AdjLower);

if AdjP <= 0.05;

run;

proc sort data=StatisticallySigDiffs;

by AdjP descending Estimate\_NormalScale;

run;

proc print data=StatisticallySigDiffs;

var KitchenQual Neighborhood \_KitchenQual \_Neighborhood Adjp Estimate\_NormalScale UpperCI\_NormalScale LowerCI\_NormalScale;

run;