**Predicting the Price of Homes in Ames, Iowa**

**14th April 2025**

**Chloe Barker & Tracy Dower**

**INTRODUCTION**

For this study we examine what features drive home sale prices in the residential market of Ames, Iowa. Using the Ames Housing dataset (1,460 homes; 79 explanatory variables), we apply linear‑regression techniques, variable transformations, and feature‑selection methods to identify the factors that best predict a home’s selling price. The first analysis for real estate company Century 21 Ames, focuses on the NAmes, Edwards, and BrkSide neighborhoods, estimating how SalePrice changes for every additional 100 square feet of living area (c\_GrLivArea) and testing whether this slope differs by neighborhood. The second analysis will include a city‑wide prediction across all neighborhoods, building and comparing several linear regression models, to determine which subset of the 79 variables most accurately forecasts future sale prices. The findings provide actionable insights for realtors and guide the design of market‑preferred homes in Ames.

**DATA DESCRIPTION**

**Source:** Ames Housing dataset (Dean De Cock, via Kaggle)

**Observations:** 1,460 residential properties sold between 2006 and 2010

* **Variables:** 79 explanatory variables describing many aspects of residential homes

**Variables of Interest:**

* **SalePrice:** Final selling price of the house
* **GrLivArea:** Total above-ground living area in
* **c\_GrLivArea:** Total above-ground living area in 100 sq. ft.
* **Neighborhood:** Physical location in Ames
  + Analysis 1: Neighborhood classification (Edwards, NAmes, BrkSide)
* **OverallQual:** overall material and finish quality
* **FullBath:** Number of full bathrooms above ground
* **TotalQualityInt:** ExterQualInt + BsmtQualInt + ExterCondInt + PoolQCInt + GarageQualInt + GarageCondInt + HeatingQCInt + FireplaceQuInt + KitchenQualInt + BsmtCondInt
* **c\_GarageArea:** Size of garage in 100 sq. ft.
* **AgeofHouse**: YrSold (Year house was sold) - YearBuilt (Year house was built)
* **MSSubClass:** Class of building (1945 & older, 1946 & newer, etc.)
* **FullBath:** Number of full bathrooms above ground
* **BldgType:** Type of dwelling
* **YrSold:** Year sold (between 2006 and 2010)
* **YearBuilt:** Year sold (between 1872 and 2010)
* **KitchenQualityInt:** Kitchen quality - Excellent (5), Good(4), Typical/Average(3), Fair(2), Poor(1)

**Reference:**

Kaggle dataset: <https://www.kaggle.com/c/house-prices-advanced-regression-techniques>

**ANALYSIS QUESTION 1**

**RESTATEMENT OF THE PROBLEM**

What is the relationship between the square footage of the living area of the house and SalePrice? Do the different neighborhoods (Edwards, NAmes, BrkSide) affect this relationship?

The soundness of conclusions we reach from linear regression depends on the assumptions that the predictors have a linear relationship with the metric we wish to predict, that each observation is independent of each other observation, that errors are normally distributed, and that there is constant variance of errors across all predictors.

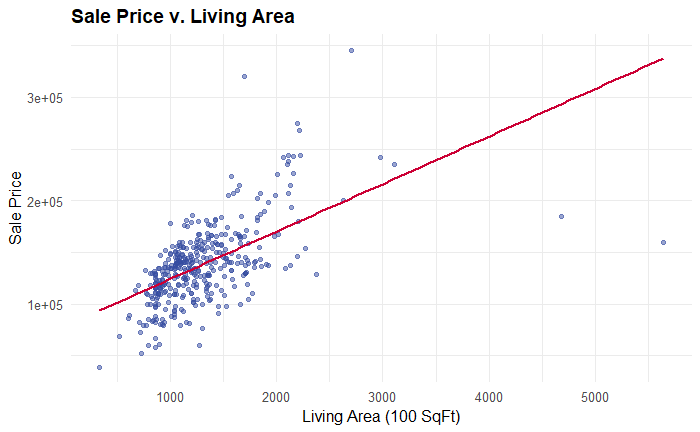
**BUILD AND FIT THE MODEL**

Predicted log(SalePrice) = β0 + β1(log(c\_GrLivArea))+ β2(Edwards)+ β3(BrkSide)

+ β4(Edwards \*log(c\_GrLivArea)) + β5(BrkSide\*log(c\_GrLivArea))

**ADDRESSING ASSUMPTIONS of LINEAR REGRESSION**

**Untransformed Data**



From our scatterplot with a regression line, we see considerable deviation between the observations and the regression. For the untransformed data, we do not see a clear linear relationship. There is a fanning-out of the residuals at the higher range. Thus we pursued a log-transformation of both SalePrice and GRLivingArea.

**Log-transformed SalePrice and GrLivingArea**

A graph and diagram of a graph

AI-generated content may be incorrect.

We gained some improvement in fit from the log-transformations. From our residuals plot, we see homoscedasticity (constant variance). On the histogram, we see that our residuals are approximately normally distributed. On the Q-Q plot we see that, except for the very high and low ends of SalePrice, the residuals fall very close to the reference line which suggests that they are normally distributed. From our scatterplot with a regression line, we see evidence of a linear relationship between the predictors (GrLivingArea) and the outcome (SalePrice). We sought to improve on this model by controlling for Neighborhood.

**Log-transformed SalePrice and GrLivingArea, Controlling for Neighborhood**

**A graph of a graph and a graph of a graph

AI-generated content may be incorrect.**

Once we control for Neighborhood and the interaction between neighborhood and additional square feet of GrfLivingArea, on the Q-Q plot the observations fall much closer to the diagonal reference line. Leverage plot, only a handful of the residuals have an absolute value >=2. There are a few outliers, however, on the Cook’s D plot of potentially influential observations, we see that only one observation has a high Cook's D relative to others, but it is still considerably less than 1. As for Independence of Errors, the Durbin-Watson test statistic D was 1.873, close to the ideal value of 2.0, indicating no significant autocorrelation.

* Is the relationship significantly different for different neighborhoods?

**A screenshot of a computer

AI-generated content may be incorrect.**The impact of each additional 100 ft2 of living area on home sale price is significantly different per Neighborhood (F-statistic 8.649, p-value = 0.0002).

**COMPARING COMPETING MODELS**

|  |  |  |  |
| --- | --- | --- | --- |
| Models | Adj R^2 | Internal CV press | AIC |
| {SalePrice|GrLivArea} | 0.3406 | 3.93E+11 | 8308.39646 |
| {log(SalePrice) | log(GrLivArea)} | 0.4188 | 16.8364 | -814.0689 |
| {log(SalePrice) | log(GrLivArea), Neighborhood} | 0.4857 | 15.00131 | -858.86316 |
| {log(SalePrice) | log(GrLivArea), Neighborhood} - flexible slopes | 0.5056 | 14.60908 | -872.04521 |

**PARAMETERS**

**Estimates**

Predicted log(SalePrice) =10.671 +0.473(log(c\_GrLivArea))-0.271(Edwards)-0.984(BrkSide)

+0.347(BrkSide\*log(c\_GrLivArea))

**BrkSide:**

Predicted log(SalePrice) =9.687 +0.820(log(c\_GrLivArea))

**Edwards:**

Predicted log(SalePrice) =10.400 +0.473(log(c\_GrLivArea))

**NAmes:**

Predicted log(SalePrice) =10.671 +0.473(log(c\_GrLivArea))

A screenshot of a data table

AI-generated content may be incorrect.

**Interpretation and Confidence Intervals**

* A doubling of 100 square feet in living area is associated with a 20.473 = 1.38 multiplicative increase in the median SalePrice, or a 38% increase, holding neighborhood constant. This effect corresponds to the slope for NAmes, our reference group.
* A 95% confidence interval for the multiplicative increase in median SalePrice after a doubling of 100 sq. ft of living area is in the interval (20.384, 20.562) = (1.58, 1.74). This corresponds to a 58% to 74% increase in the median SalePrice.
* The interaction term between Edwards and log living area is not statistically significant (p = 0.52), meaning the effect of living area on SalePrice in Edwards does not differ meaningfully from NAmes. Therefore, a separate slope for Edwards is not included in the final model.
* The interaction term for BrkSide is highly significant, supporting the inclusion of a different slope for this neighborhood (p=0.001). This provides strong evidence to retain the interaction terms between neighborhood and log(living area).
* For Brkside, doubling of 100 square feet in living area is associated with a 20.820 = 1.75 multiplicative increase in median Sale Price or a 75% increase or a 20.347 = 1.27 multiplicative increase in median Sale Price from the NAmes and Edwards group. A 95% confidence interval for the multiplicative increase in median of Sale Price after a doubling of 100 sq. ft of living area for the BrkSide neighborhood is a (20.180, 20.513) = (1.13, 1.43) = (13%, 43%) increase. This corresponds to a 13% to 43% increase in median SalePrice from the Edwards and NAmes groups, on average.
* The difference in intercepts between BrkSide and NAmes is statistically significant (p = 0.001), suggesting that BrkSide homes start from a significantly lower price baseline.
* In contrast, the intercept difference for Edwards vs. NAmes is not statistically significant (p = 0.14). While this term is retained in the model, it only suggests a slight downward shift in Edwards pricing.
* Our best estimate of this difference is that homes in Edwards sell for approximately e-0.271= 24% less than NAmes homes but follow the same rate of price increase with additional living area.
* Overall, there is significant evidence that the relationship between 100 square feet of living area and SalePrice is stronger in BrkSide than in NAmes or Edwards.

**CONCLUSION**

Overall, we found sufficient evidence to suggest that SalePrice (and its relationship to square footage) is influenced by which neighborhood the house is located in. A doubling of 100 sq. ft. of living area for residential homes in the NAmes and Edwards neighborhoods equates to a multiplicative change of 20.437 = 1.36 in the median distribution of sale prices for the given Ames dataset. In other words, a doubling of 100 sq. ft. of living area increases the estimated median sale price by 36%. A 95% confidence interval for the multiplicative increase in median sale price after a doubling of 100 sq. ft. of living area in the Edwards and NAmes neighborhoods is within the interval (20.384, 20.562) = (1.58, 1.74). This corresponds to a 58% to 74% increase in the median sale price. This is an observational study thus only associations can be made. It is estimated that 50.56% of the variance in Sales Price can be explained by its relationship with general living area (per 100 sq. ft.) and neighborhood designation (NAmes, Edwards, and BrkSide).

[**R Shiny: Price v. Living Area Chart**](https://tracydower.shinyapps.io/StatsFinal_HomeSalesAmes/)

[**https://tracydower.shinyapps.io/StatsFinal\_HomeSalesAmes/**](https://tracydower.shinyapps.io/StatsFinal_HomeSalesAmes/)

**ANALYSIS QUESTION 2**

**RESTATEMENT OF THE PROBLEM**

Identify the subset of predictors that most accurately explains and forecasts the SalePrice for residential homes, including all neighborhoods in our Ames dataset. To do so, we will fit four candidate linear‑regression models - one SLR and two MLRs - and compare their adjusted R², CV‑PRESS, and Kaggle scores to determine which variables and model offer the strongest, most reliable prediction of future sale prices.

**CANDIDATE MODELS**

**Simple Linear Regression**

Predicted log(SalePrice) = β0+ β1(OverallQual)

**Multiple Linear Regression 1**

Predicted log(SalePrice) = β0+ β1(GrLivArea)+ β2(FullBath2) + β3(FullBath3)

**Multiple Linear Regression 2**

Predicted log(SalePrice) = β0+ β1(log(c\_GrLivArea)) + β2(OverallQual) + β3(TotalQualityInt)

+ β4(AgeofHouse) + β5(MSSubClass) + ∑ βi(Neighborhood*i*)

**Multiple Linear Regression 3**

Predicted log(SalePrice) = β0 + β1(log(c\_GrLivArea)) + β2(OverallQual) + β3(TotalQualityInt)

+ β4(YrSold) + β5(KitchenQualityInt) + β6(MSSubClass) + β7(YearBuilt) +∑ βi(BldgTypei) +∑ βi(Neighborhoodi)

**COMPARING COMPETING MODELS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Predictive Models | Adjusted R2 | CV PRESS | AIC | Kaggle Score |
| Simple Linear Regression | 0.6676 | 7.75E+01 | -2823.47692 | 0.22896 |
| Multiple Linear Regression 1 | 0.5601 | 102.84847 | -2412.5151 | 0.28306 |
| Multiple Linear Regression 2 | 0.8609 | 33.9667 | -4067.80922 | 0.15243 |
| Multiple Linear Regression 3 | 0.8665 | 32.68418 | -4121.93136 | 0.14999 |

A graph with a line drawn on it

AI-generated content may be incorrect.A graph of residuals with red line

AI-generated content may be incorrect.

A graph with blue dots and red lines

AI-generated content may be incorrect.A graph with numbers and lines

AI-generated content may be incorrect.

**Addressing the Assumptions of Linear Regression: Model 4**

Our final model has a much better fit as we see on the Q-Q plot. Only the residuals at the high and low ends of Sale Price vary significantly from the regression line. The histogram confirms that the errors are normally distributed for the most part, except for a longer left tail. As our data sets contain homes built between 1872 and 2010, having as few as zero bedrooms and as many as 8, it is reasonable to consider that even homes in the same town may not be able to fit one model tightly, without overfitting. On left side of the Residuals vs. Leverage plot we see that most observations have low leverage and reasonably small, standardized residuals - most falling within 4 standard deviations. Note the two homes with relatively high leverage, however they do not have extremely large residuals. On the Cook's D plot we see that a handful of homes stand head-and-shoulders above the rest, but even these have Cook's D well below 1 which is reassuring that no single outlier is having undo impact on the model. As for Independence of Errors, the Durbin-Watson test statistic D was nearly the ideal value of 2.0 (2.0034, p-value 0.94), indicating no significant autocorrelation in the residuals. This supports the assumption of independent residuals.

**CONCLUSION**

Based on the table, the third MLR model performs best in terms of being able to predict future sale prices of homes in Ames, Iowa. The adjusted R2 value is the highest for this model in that it explains approximately 86.54% of the variance in log sale prices. The CV press is also the lowest for this model, indicating it had less error under our cross-validation process than the other two models. The high CV press for the second model makes this model a worse choice as multicollinearity could play a part in the higher statistic. Furthermore, our lowest AIC was found in the third multiple linear regression model, reaffirming our confidence in the predictive power of our last model. Lastly, looking at the Kaggle Score value, we were able to achieve by far the best score with the last model, making it our best estimate for predicting sale price in Ames, Iowa for this data.

**GitHub Pages Websites:**

**Chloe Barker:**[**https://github.com/chloedbarker/chloedbarker.github.io**](https://github.com/chloedbarker/chloedbarker.github.io)

**Tracy Dower:** [**https://github.com/tracydower/HomeSales\_AmesIowa\_Stats**](https://github.com/tracydower/HomeSales_AmesIowa_Stats)

APPENDIX

R and Graphics

# Install and Load Required Packages

packages <- c("olsrr","ggplot2","boot","flextable","gtsummary","labelled","overviewR","stringr","tidyverse","dplyr","kableExtra","ggfortify","car")  
to\_install <- packages[!packages %in% installed.packages()[,"Package"]]  
if (length(to\_install)) install.packages(to\_install)  
library(tidyverse) # includes dplyr and %>%

library(ggplot2) # pretty plots  
library(gtsummary) # create publication-ready summary tables with minimal code

library(flextable) # pretty tables

library(labelled) # for set\_variable\_labels

library(overviewR)

library(dplyr) # dplyr is included in the tidyverse but it's acting funny, so load it explicitly  
library(readr)  
library(stringr)  
hexSmuBlue <- "#354CA1"  
hexSmuRed <- "#CC0035"

## Load Train and Test Data

Select useful columns We want SalePrice plus only variables that are AVAILABLE in both datasets, otherwise they are useless as predictors.

train <- read\_csv("train.csv", show\_col\_types = FALSE)  
test <- read\_csv("test.csv", show\_col\_types = FALSE)  
train <- train %>% mutate(DataSet = "train")  
test <- test %>% mutate(DataSet = "test")  
combinedData <- bind\_rows(train, test)  
# names(combinedData)

# Exploratory Data Analysis

## Data Preparation

### Correct missing or nonsense values

Dr. Sadler reports that all missing are missing completely at random, MCAR. #### Garages

library(dplyr)  
# If GarageCars==0 then set all Garage Categorical Variables = "No Garage"  
combinedData <- combinedData %>% mutate(across(c(GarageType, GarageYrBlt, GarageFinish, GarageQual, GarageCond), ~ifelse(GarageCars == 0, NA, .)))  
combinedData <- combinedData %>% mutate(across(c(GarageType, GarageFinish, GarageQual, GarageCond), ~ifelse(GarageCars == 0, NA, .))) %>%  
 mutate(GarageYrBlt = ifelse(GarageCars == 0, NA, GarageYrBlt))  
# Where ID = 2760, GarageYrBlt = 2207 which is nonsense, and YearBuilt (the year the home was built) is 2007, so we will assume that the garage was built the same year.  
combinedData$GarageYrBlt <- as.numeric(combinedData$GarageYrBlt) # NA will still be NA  
combinedData[combinedData$Id == 2760, "GarageYrBlt"] <- 2007  
combinedData$GarageArea[is.na(combinedData$GarageArea)] <- 0  
combinedData$GarageArea <- ifelse(is.na(combinedData$GarageArea),0,combinedData$GarageArea)  
combinedData$GarageExists <- ifelse(combinedData$GarageArea > 0, 1, 0)  
combinedData <- transform(combinedData, c\_GarageArea = GarageArea / 100)  
combinedData$lc\_GarageArea <- log(ifelse(combinedData$GarageArea == 0, 1, combinedData$GarageArea))  
# cat(sort(names(combinedData)), sep = "\n")

#### Transformations

combinedData <- combinedData %>% mutate(lSalePrice = log(SalePrice))  
combinedData <- transform(combinedData, c\_GrLivArea = GrLivArea / 100)  
combinedData <- combinedData %>% mutate(lc\_GrLivArea = log(c\_GrLivArea))

# Datedness = years since most recent of (YearBuilt, Year Remodeled)

How long before the sale of the home was it last built or remodeled?

combinedData$FreshDate = (combinedData$YrSold - pmax(combinedData$YrSold, combinedData$YearRemodAdd))  
# Not significant. Do not use.  
combinedData$YearFromRemodelToSale = (combinedData$YrSold - combinedData$FreshDate)  
combinedData$AgeofHouse = (combinedData$YrSold - combinedData$YearBuilt)

#### Lot Frontage

Out of 2920 homes in our data (1460 in each set), 487 were missing data for LotFrontage: train: 259/1460 = 18% test: 228/1460 = 16% We explored various methods of predictive imputation of missing values for LotFrontage. We expected a strong relationship between LotFrontage and LotArea, but the relationship was weak ( R-squared: 0.1816). Accounting for LotArea/LotFrontage, GrLivArea, and Neighborhood raised our R-squared to 0.5744. We decided to impute the missing values based on the median by neighborhood.

lot\_data <- combinedData %>%  
 filter(LotArea <= quantile(LotArea, 1, na.rm = TRUE)) %>%  
 mutate(LotLength = LotArea / LotFrontage)  
library(ggplot2)  
ggplot(lot\_data, aes(x = LotArea, y = LotFrontage)) +  
 geom\_point(alpha = 0.5) +  
 geom\_smooth(method = "lm", color = hexSmuBlue) +  
 labs(title = "Lot Area vs. Lot Frontage", x = "Lot Area ", y = "Lot Frontage (feet)") +  
 theme\_minimal()

A graph with a line and a line

AI-generated content may be incorrect.

model <- lm(LotFrontage ~ LotArea + GrLivArea + LotShape + Neighborhood, data = combinedData[!is.na(combinedData$LotFrontage), ])  
 # summary(model)  
model <- lm(LotFrontage ~ (LotArea/LotFrontage) + GrLivArea + Neighborhood, data = lot\_data)  
 # summary(model)   
 combinedData <- combinedData %>% group\_by(Neighborhood) %>%  
 mutate(LotFrontage = ifelse(is.na(LotFrontage), median(LotFrontage, na.rm = TRUE), LotFrontage)) %>% ungroup()

#### Miscellaneous Missing Values

Pools, Alleys, Basements, Masonry, Fireplaces, Electrical, Fences, MiscFeature

# If PoolArea == 0 then PoolQC = "No Pool"  
combinedData <- combinedData %>% mutate(PoolQC = ifelse(PoolArea == 0, "No Pool", PoolQC))  
# If Alley is NA set Alley = "None"  
combinedData <- combinedData %>% mutate(Alley = ifelse(is.na(Alley), "None", Alley))  
# Handle basement NA values and consistency  
combinedData <- combinedData %>% mutate(BsmtTotalSF = BsmtFinSF1 + BsmtFinSF2 + BsmtUnfSF)  
# Where BsmtTotalSF == 0, set each categorical basement field = NA  
 rows <- which(combinedData$BsmtTotalSF==0)  
 cols <- c("BsmtCond", "BsmtExposure", "BsmtFinType1", "BsmtFinType2", "BsmtQual", "BsmtFullBath", "BsmtHalfBath")  
 combinedData[rows,cols] <-NA  
# Where BsmtTotalSF > 0, if BsmtExposure is NA, that's a data entry error, and it should be "No"  
 combinedData <- combinedData %>% mutate(BsmtExposure = ifelse(is.na(BsmtExposure) & BsmtTotalSF > 0, "No", BsmtExposure))  
 combinedData <- combinedData %>% mutate(BsmtFinType1 = ifelse(is.na(BsmtFinSF1) | BsmtFinSF1 == 0, "Unf", BsmtFinType1))  
# Where BsmtFinSF2 >0 and BsmtFinType2 NA, set BsmtFinType2="Unf"  
 combinedData <- combinedData %>% mutate(BsmtFinType2 = ifelse(is.na(BsmtFinType2) & BsmtFinSF2 > 0, "Unf", BsmtFinType2))  
# count rows where at least one of these fields is NA but not ALL of these fields are NA  
 sum(rowSums(is.na(combinedData[, c("BsmtCond", "BsmtExposure", "BsmtFinType1", "BsmtFinType2", "BsmtQual")])) > 0 &  
 rowSums(is.na(combinedData[, c("BsmtCond", "BsmtExposure", "BsmtFinType1", "BsmtFinType2", "BsmtQual")])) < 7)

## [1] 84

# Fix MasVnrType  
# Only 2 homes in our combined dataset had a value for MasVnrType other than none/NA when MasVnrArea when 0 or NA. So we set MasVnrArea = 0 when MasVnrArea = NA and MasVnrType = "None" to MasVnrArea = 0.  
combinedData <- combinedData %>% mutate(MasVnrArea = ifelse(is.na(MasVnrArea), 0, MasVnrArea))  
combinedData <- combinedData %>% mutate(MasVnrType = ifelse(MasVnrArea==0, "None" , MasVnrType))  
# If no Fireplaces, Fireplace quality = "No Fireplace"  
combinedData <- combinedData %>% mutate(FireplaceQu = ifelse(Fireplaces==0, "No Fireplace" , FireplaceQu))  
# 92% of homes in our dataset (2671 or 2919) had Sbrker for electrical type. 1 home had NA. We assumed Sbrker for that home.  
combinedData <- combinedData %>% mutate(Electrical = ifelse(is.na(Electrical), "SBrkr", Electrical))  
# Fence = Fence Quality. Many homes have no fence.  
combinedData <- combinedData %>% mutate(Fence = ifelse(is.na(Fence), "None" , Fence))  
# Most homes have no MiscFeature such as an elevator, 2nd Garage, large Shed, or Tennis Court.  
combinedData <- combinedData %>% mutate(MiscFeature = ifelse(is.na(MiscFeature ), "None" , MiscFeature ))

### Final check for missing values in combined data. Exclude from this check columns where NA is reasonable

# Remove specified columns  
combinedData <- combinedData[, !(colnames(combinedData) %in% c("BsmtCond", "BsmtExposure", "BsmtFinSF1", "BsmtFinSF2",  
 "BsmtFinType1", "BsmtFinType2", "BsmtFullBath",  
 "BsmtHalfBath", "BsmtQual", "BsmtTotalSF", "BsmtUnfSF"))]  
FinalVariables <- c("DataSet","Id","SalePrice","lSalePrice", "AgeofHouse", "BldgType", "FullBath", "KitchenQualInt",  
 "lc\_GrLivArea", "MSSubClass", "Neighborhood", "OverallQual",  
 "TotalQualityInt", "YearBuilt", "YrSold")  
final\_cols <- intersect(FinalVariables, colnames(combinedData))  
na\_counts <- colSums(is.na(combinedData[final\_cols]))  
na\_counts <- colSums(is.na(combinedData))  
na\_counts <- na\_counts[na\_counts > 0]  
cat(paste(names(na\_counts), na\_counts, sep = "\t"), sep = "\n")

## MSZoning 4  
## Utilities 2  
## Exterior1st 1  
## Exterior2nd 1  
## MasVnrType 1  
## TotalBsmtSF 1  
## KitchenQual 1  
## Functional 2

### Transform and derive variables

### Add levels to categorical variables

# names(combinedData)  
# nrow(combinedData)  
encode\_levels <- function(myLevels, myVariables, myData) {  
 for (var in myVariables) {  
 new\_var <- paste0(var, "Int")  
 mapped <- myLevels[as.character(myData[[var]])]  
 stopifnot(length(mapped) == nrow(combinedData))  
 myData[[new\_var]] <- ifelse(is.na(mapped), 0, mapped)  
 }  
 return(myData)  
}

## Create Integers for Measures of Quality

# names(combinedData)  
myVariables <- c("ExterQual", "ExterCond", "HeatingQC","KitchenQual", "FireplaceQu", "GarageQual", "GarageCond", "PoolQC")  
# missing\_vars <- setdiff(myVariables, names(combinedData))  
# print(missing\_vars)  
myLevels <- c("Po" = 2, "Fa" = 2, "TA" = 3, "Gd" = 4, "Ex" = 5)  
 combinedData <- encode\_levels(myLevels, myVariables, combinedData)  
# table(combinedData$KitchenQualInt, useNA = "ifany")  
 combinedData$KitchenQualInt <- ifelse(combinedData$KitchenQualInt > 1, combinedData$KitchenQualInt, 2)  
 combinedData <- combinedData %>% mutate(TotalQualityInt = ExterQualInt + ExterCondInt + PoolQCInt + GarageQualInt + GarageCondInt + HeatingQCInt + FireplaceQuInt + KitchenQualInt )  
# table(combinedData$KitchenQualInt, useNA = "ifany")

### Utility Access

myLevels <- c("AllPub" = 4, "NoSewr" = 3, "NoSeWa" = 2, "ELO" = 1)  
 myVariables <- c("Utilities")  
 combinedData <- encode\_levels(myLevels, myVariables, combinedData)

#### Recode N Y P as integers

YesNo\_map <- c("N" = 0, "P" = 0.5, "Y" = 1)  
combinedData$PavedDriveInt <- YesNo\_map[as.character(combinedData$PavedDrive)]  
combinedData$CentralAirInt <- YesNo\_map[as.character(combinedData$CentralAir)]

## Bedrooms and Bathrooms

There were 12 homes with 0 values for FullBath and/or BedroomAbvGr. As these where all single-family homes, we assumed these were data entry errors and imputed the missing values by using linear regression for HouseStyle and GrLivArea. Additionally, the test dataset has more levels than the train data set so when FullBath = 4 set FullBath = 3.

bed\_model <- lm(BedroomAbvGr ~ GrLivArea + HouseStyle, data = combinedData[combinedData$BedroomAbvGr > 0, ])  
bed\_rows <- which(combinedData$BedroomAbvGr == 0)  
combinedData$BedroomAbvGr[bed\_rows] <- round(predict(bed\_model, newdata = combinedData[bed\_rows, ]))  
bath\_model <- lm(FullBath ~ GrLivArea + HouseStyle + BedroomAbvGr, data = combinedData[combinedData$FullBath > 0, ])  
bath\_rows <- which(combinedData$FullBath == 0)  
combinedData$FullBath[bath\_rows] <- round(predict(bath\_model, newdata = combinedData[bath\_rows, ]))  
# str(combinedData$FullBath)  
combinedData$FullBath[combinedData$FullBath > 3] <- 3

#### Simplify Home Exterior

Simplify Home Exterior into fewer categories. | Original Exterior Values | Simplified Category | |————————————–|———————-| | Stone, Stucco | StoneOrStucco | | Wd Sdng, Wd Shng, WdShing | WoodShingle | | Brick, Brk Cmn, BrkComm, BrkFace | Brick | | All other values | Other |

exterior\_map <- function(x) {  
 case\_when(  
 x %in% c("Stone", "Stucco") ~ "StoneOrStucco",  
 x %in% c("Wd Sdng", "Wd Shng", "WdShing") ~ "WoodShingle",  
 x %in% c("Brick", "Brk Cmn", "BrkComm", "BrkFace") ~ "Brick",  
 TRUE ~ "Other" # all other cases will be classified as "other"  
 )  
}  
combinedData$ExteriorSimplified <- exterior\_map(combinedData$Exterior1st)

# Analysis 1: Century21Ames

Linear Regression of Sale Price and GrLIvArea for Neighborhoods NAmes, Edwards and BrkSide. SalePrice by SquareFoot DEFINITELY varies by Neighborhood, so do the interactions.

# Filter combined data to only the neighborhoods of interest  
dataCentury21Ames <- subset(combinedData, Neighborhood %in% c("NAmes", "Edwards", "BrkSide"))  
ggplot(dataCentury21Ames, aes(x = GrLivArea, y = SalePrice)) +  
geom\_point(color = hexSmuBlue, alpha = 0.5) +  
 geom\_smooth(method = "lm", se = FALSE, color = hexSmuRed) +  
 labs( title = "Sale Price v. Living Area", x = "Living Area (100 SqFt)", y = "Sale Price" )+  
 theme\_minimal(base\_size = 12) +  
 theme(plot.title = element\_text(face = "bold"))

A graph with a red line and blue dots

AI-generated content may be incorrect.

modelCentury21Ames\_SLR <- lm(lSalePrice ~ lc\_GrLivArea, data = dataCentury21Ames)  
# summary(modelCentury21AmesSLR)  
# Regression with interaction:  
dataCentury21Ames$Neighborhood <- factor(dataCentury21Ames$Neighborhood)  
dataCentury21Ames$Neighborhood <- relevel(dataCentury21Ames$Neighborhood, ref = "NAmes") # Set NAmes as the reference Neighborhood  
modelCentury21Ames\_MLR <- lm(lSalePrice ~ lc\_GrLivArea \* Neighborhood, data = dataCentury21Ames)  
# summary(modelCentury21Ames\_MLR)  
ggplot(dataCentury21Ames, aes(x = lc\_GrLivArea, y = lSalePrice, color = Neighborhood)) +  
 geom\_point(alpha = 0.5) +  
 geom\_smooth(method = "lm", se = FALSE) +  
 labs( title = "Sale Price v. Living Area by Neighborhood",  
 subtitle="With Interactions between Living Area and Neighborhood", x = "Living Area log(100 SqFt)", y = "Sale Price (log-transformed)" )

A graph of a living area

AI-generated content may be incorrect.

theme\_minimal()

# ANALYSIS 1 QUESTION 2

## Build your own ANOVA to answer

Is the relationship significantly different for different neighborhoods? The impact of each additional 100 ft^2 of living area on home sale price is significantly different per Neighborhood (F-statistic 8.649, p-value = 0.0002).

# Simple Linear Regression  
tbl\_regression(modelCentury21Ames\_SLR, exponentiate = FALSE)

| **Characteristic** | **Beta** | **95% CI** | **p-value** |
| --- | --- | --- | --- |
| lc\_GrLivArea | 0.57 | 0.50, 0.64 | <0.001 |
| Abbreviation: CI = Confidence Interval | | | |

# Multiple Linear Regression with Interaction  
tbl\_regression(modelCentury21Ames\_MLR, exponentiate = FALSE)

| **Characteristic** | **Beta** | **95% CI** | **p-value** |
| --- | --- | --- | --- |
| lc\_GrLivArea | 0.47 | 0.38, 0.56 | <0.001 |
| Neighborhood |  |  |  |
| NAmes | — | — |  |
| BrkSide | -0.98 | -1.4, -0.57 | <0.001 |
| Edwards | -0.27 | -0.63, 0.09 | 0.14 |
| lc\_GrLivArea \* Neighborhood |  |  |  |
| lc\_GrLivArea \* BrkSide | 0.35 | 0.18, 0.51 | <0.001 |
| lc\_GrLivArea \* Edwards | 0.05 | -0.10, 0.19 | 0.5 |
| Abbreviation: CI = Confidence Interval | | | |

# Test whether the full model is significantly better  
# anova(modelCentury21Ames\_SLR, modelCentury21Ames\_MLR)   
modelCentury21Ames\_SLR <- lm(log(SalePrice) ~ log(GrLivArea) + Neighborhood, data = dataCentury21Ames)  
modelCentury21Ames\_MLR <- lm(log(SalePrice) ~ log(GrLivArea) \* Neighborhood, data = dataCentury21Ames)  
# Compare models using ANOVA  
anova\_results <- anova(modelCentury21Ames\_SLR, modelCentury21Ames\_MLR)  
# Display ANOVA comparison table  
library(kableExtra)

anova\_results %>%  
 kable(caption = "ANOVA: Does the Relationship Differ by Neighborhood?") %>%  
 kable\_styling(full\_width = FALSE, position = "left")

# A screenshot of a computer AI-generated content may be incorrect.

# ANALYSIS 2

## Explore other vairables

library(patchwork)  
train\_clean <- combinedData %>% filter(DataSet == "train")  
cat\_vars <- c("Alley", "BedroomAbvGr", "BldgType", "CentralAir", "Electrical", "ExterCond",  
 "Fireplaces", "Foundation", "FullBath", "GarageCars", "GarageQual", "HalfBath",  
 "HouseStyle", "KitchenQual", "LandContour", "LandSlope", "MSZoning", "Neighborhood",  
 "OverallCond", "OverallQual", "PoolQC", "RoofStyle", "SaleCondition", "SaleType",  
 "Utilities")  
num\_vars <- c("MiscVal", "WoodDeckSF", "OpenPorchSF", "EnclosedPorch","ScreenPorch","PoolArea", "GarageArea", "LotFrontage",  
 "LotArea", "MasVnrArea", "TotalBsmtSF", "YearBuilt", "YearRemodAdd", "YrSold", "c\_GrLivArea", "lc\_GrLivArea", "AgeofHouse",  
 "TotalQualityInt", "OverallQual", "MSSubClass")  
final\_cols <- intersect( colnames(combinedData), cat\_vars)  
final\_cols <- intersect( colnames(combinedData), num\_vars)  
# cat\_vars <- names(combinedData)[ sapply(combinedData, function(x) { is.character(x) || is.factor(x) || (is.numeric(x) && length(unique(x)) < 5) })]  
# num\_vars <- names(combinedData)[sapply(combinedData, function(x) is.numeric(x) && length(unique(x)) > 5)]  
# num\_vars <- num\_vars[!grepl("Int$", num\_vars)]  
# num\_vars <- num\_vars[!grepl("\_", num\_vars)]  
# num\_vars <- setdiff(num\_vars, "Id")  
## Categorical Graph  
train\_clean <- train\_clean %>% mutate(across(all\_of(cat\_vars), as.factor))  
cat\_var\_stats <- map\_df(cat\_vars, function(v) {  
 f <- reformulate(v, response = "lSalePrice")  
 mod <- lm(f, data = train\_clean)  
 tibble(var = v, adj\_r2 = summary(mod)$adj.r.squared)  
})  
cat\_var\_stats %>%  
 arrange(desc(adj\_r2)) %>%  
 slice\_max(adj\_r2, n = 10) %>%  
 mutate(var = fct\_reorder(var, adj\_r2)) %>%  
 ggplot(aes(adj\_r2, var)) +  
 geom\_col(fill = hexSmuRed) +  
 labs(x = "Adjusted R\u00B2",y = "Explanatory Variables",  
 title = "Adjusted R\u00B2 of Categoricals") +  
 theme\_minimal(base\_size = 12)

A graph of a bar graph

AI-generated content may be incorrect.

## Specific Categorical Variable Graphs  
plot\_box = function(df, var) {  
 medians = df %>% group\_by(.data[[var]]) %>% summarize(median\_price = median(lSalePrice, na.rm = TRUE)) %>%  
 arrange(median\_price)  
 df[[var]] = factor(df[[var]], levels = medians[[var]])  
 ggplot(df, aes(x = .data[[var]], y = lSalePrice)) +  
 geom\_boxplot(outlier.alpha = 0.2, width = 0.6, fill = hexSmuBlue) +  
 stat\_summary(fun = median, geom = "point", shape = 21, size = 2, fill = hexSmuRed, color = hexSmuBlue) +  
 labs(title = paste("Log Sale Price vs", var), y = "log(Sale Price) ($)") +  
 theme\_minimal(base\_size = 12) +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1),legend.position = "none")  
}  
# Insignificant variation  
 plot\_box(train\_clean, "Neighborhood")

A graph of a price

AI-generated content may be incorrect.

plot\_box(train\_clean, "MSSubClass")

A graph of blue and red lines

AI-generated content may be incorrect.

plot\_box(train\_clean, "FullBath")

A graph of blue and red lines

AI-generated content may be incorrect.

plot\_box(train\_clean, "OverallQual")

A graph with blue squares and red dots

AI-generated content may be incorrect.

# significant variation between levels  
 plot\_box(train\_clean, "MasVnrType")

A graph of blue and red squares

AI-generated content may be incorrect.

plot\_box(train\_clean, "ExterCond")

A graph with blue squares and red dots

AI-generated content may be incorrect.

plot\_box(train\_clean, "HouseStyle")

A graph of a graph with blue and red squares

AI-generated content may be incorrect.

plot\_box(train\_clean, "BedroomAbvGr")

A graph of a price

AI-generated content may be incorrect.

## Numeric Variables Graph  
train\_clean <- train\_clean %>% mutate(across(all\_of(num\_vars), as.numeric))  
num\_var\_stats <- map\_df(num\_vars, function(v) {  
 f = as.formula(paste("lSalePrice ~", v))  
 mod = lm(f, data = train\_clean)  
 tibble(var = v,adj\_r2 = summary(mod)$adj.r.squared)  
})  
num\_var\_stats %>%  
 arrange(desc(adj\_r2)) %>%  
 mutate(var = fct\_reorder(var, adj\_r2)) %>%  
 ggplot(aes(adj\_r2, var)) +  
 geom\_col(fill = hexSmuBlue) +  
 labs(x = "Adjusted R\u00B2", y = "Explanatory Variables", title = "Adjusted R\u00B2 of Categorical Predictors for Sale Prices")

A graph of a number of individuals

AI-generated content may be incorrect.

theme\_minimal(base\_size = 12)

## Specific Numeric Variable Graphs  
plot\_scatter <- function(df, var) {  
 ggplot(df, aes(x = .data[[var]], y = lSalePrice)) +  
 geom\_point(alpha = 0.5, color = hexSmuBlue) +  
 geom\_smooth(method = "lm", se = FALSE, color = hexSmuRed, linewidth = 1) +  
 labs(title = paste("Log Sale Price vs", var),x = var,y = "log(SalePrice)") +  
 theme\_minimal(base\_size = 12)  
}  
# Significant Variation Visually  
plot\_scatter(train\_clean, "lc\_GrLivArea")

A graph with blue dots and red line

AI-generated content may be incorrect.

plot\_scatter(train\_clean, "AgeofHouse")

A graph of blue dots

AI-generated content may be incorrect.

plot\_scatter(train\_clean, "TotalQualityInt")

A graph with blue and red lines

AI-generated content may be incorrect.

plot\_scatter(train\_clean, "lc\_GarageArea")

A graph showing a line of blue dots

AI-generated content may be incorrect.

# Insignificant Variation Visually  
plot\_scatter(train\_clean, "MasVnrArea")

A graph with blue dots and a red line

AI-generated content may be incorrect.

plot\_scatter(train\_clean, "YearRemodAdd")

A graph showing the difference between a line and a line

AI-generated content may be incorrect.

plot\_scatter(train\_clean, "MSSubClass")

A graph of a graph

AI-generated content may be incorrect. ## Try various models

trainModel <- combinedData %>% filter(DataSet == "train")  
testModel <- combinedData %>% filter(DataSet == "test")  
# Limit Combine to the columns we found useful and then create CSVs for R.  
subsetData <- combinedData[,FinalVariables]  
trainFinal <- subsetData %>% filter(DataSet == "train")  
testFinal <- subsetData %>% filter(DataSet == "test")  
write.csv(trainFinal, "C:/HousesCleanTrain8.csv", row.names = FALSE)  
write.csv(testFinal, "C:/HousesCleanTest8.csv", row.names = FALSE)

# Report

## Variables Used

library(olsrr)

# Attach nice labels to the data for pretty reporting  
myData <- subsetData %>%  
 set\_variable\_labels(  
 "SalePrice" = "Property's Sale Price in US Dollars",  
 "lc\_GrLivArea" = "Ground Floor Living Area (100 Sq Ft, log transformed)",  
 "YearBuilt" = "Original Construction Date",  
 "YrSold" = "Year Sold",  
 "Neighborhood" = "Physical Locations Within Ames City Limits",  
 "AgeofHouse" = "Year Sold less Year Built",  
 "OverallQual" = "Overall Material And Finish Quality",  
 "BldgType" = "Type of Dwelling",  
 "TotalQualityInt" = "Exterior + Basement + Pool + Garage + Heating + Fireplace + Kitchen",  
 "FullBath" = "Full Bathrooms Above Grade",  
 "KitchenQualInt" = "Kitchen Quality",  
 "MSSubClass" = "Building Class"  
 )

## Final Model

library(gtsummary)  
library(gt)

trainFinal <- read\_csv("C:/HousesCleanTrain8.csv", show\_col\_types = FALSE)  
testFinal <- read\_csv("C:/HousesCleanTest8.csv", show\_col\_types = FALSE)  
trainFinal <- read\_csv("C:/HousesCleanTrain8.csv", show\_col\_types = FALSE)  
testFinal <- read\_csv("C:/HousesCleanTest8.csv", show\_col\_types = FALSE)  
# table(trainFinal$KitchenQualInt, useNA = "ifany")  
# table(testFinal$KitchenQualInt, useNA = "ifany")  
trainFinal$FullBath = as.factor(trainFinal$FullBath)  
 testFinal$FullBath = as.factor(testFinal$FullBath)  
trainFinal$MSSubClass = as.numeric(trainFinal$MSSubClass)  
 testFinal$MSSubClass = as.numeric(testFinal$MSSubClass)  
trainFinal$KitchenQualInt = as.factor(trainFinal$KitchenQualInt)  
 testFinal$KitchenQualInt = as.factor(testFinal$KitchenQualInt)  
run\_model <- function(myModel, myModelName) {  
 tableCoefficients <- tbl\_regression(myModel, exponentiate = FALSE)  
 RegressionTable<- as\_gt(tableCoefficients) %>% gt::tab\_header(title = myModelName)  
 return(RegressionTable)  
}  
# Simple Linear Regression  
model1 = lm(lSalePrice~OverallQual, data = trainFinal)  
# summary(model1) # Multiple R-squared: 0.6678, Adjusted R-squared: 0.6676  
myModel1 <- run\_model(model1, "Simple Linear Regression")  
myModel1

Table : Simple Linear Regression

| **Characteristic** | **Beta** | **95% CI** | **p-value** |
| --- | --- | --- | --- |
| OverallQual | 0.24 | 0.23, 0.24 | <0.001 |
| Abbreviation: CI = Confidence Interval | | | |

# Multiple Linear Regression 1  
model2 = lm(lSalePrice~lc\_GrLivArea + FullBath, data = trainFinal)  
# summary(model2) # Multiple R-squared: 0.561, Adjusted R-squared: 0.5601  
myModel2 <- run\_model(model2, "Multiple Linear Regression 1")  
myModel2

Table : Multiple Linear Regression 1

| **Characteristic** | **Beta** | **95% CI** | **p-value** |
| --- | --- | --- | --- |
| lc\_GrLivArea | 0.70 | 0.65, 0.75 | <0.001 |
| FullBath |  |  |  |
| 1 | — | — |  |
| 2 | 0.17 | 0.13, 0.20 | <0.001 |
| 3 | 0.29 | 0.19, 0.40 | <0.001 |
| Abbreviation: CI = Confidence Interval | | | |

# Multiple Linear Regression 2 -- # Multiple R-squared: 0.8637, Adjusted R-squared: 0.8609  
model3 = lm(lSalePrice ~ OverallQual + Neighborhood + AgeofHouse + lc\_GrLivArea + TotalQualityInt + MSSubClass, data = trainFinal)  
# summary(model3)  
myModel3 <- run\_model(model2, "Multiple Linear Regression 2")  
myModel3

Table : Multiple Linear Regression 2

| **Characteristic** | **Beta** | **95% CI** | **p-value** |
| --- | --- | --- | --- |
| lc\_GrLivArea | 0.70 | 0.65, 0.75 | <0.001 |
| FullBath |  |  |  |
| 1 | — | — |  |
| 2 | 0.17 | 0.13, 0.20 | <0.001 |
| 3 | 0.29 | 0.19, 0.40 | <0.001 |
| Abbreviation: CI = Confidence Interval | | | |

# Multiple Linear Regression 3  
model4 = lm(lSalePrice ~ YrSold + OverallQual + MSSubClass + YearBuilt + Neighborhood + BldgType + lc\_GrLivArea + TotalQualityInt + KitchenQualInt, data = trainFinal)  
# summary(model4) # Multiple R-squared: 0.8699, Adjusted R-squared: 0.8665  
myModel4 <- run\_model(model4, "Multiple Linear Regression 3")  
myModel4

Table : Multiple Linear Regression 3

| **Characteristic** | **Beta** | **95% CI** | **p-value** |
| --- | --- | --- | --- |
| YrSold | 0.00 | -0.01, 0.00 | 0.4 |
| OverallQual | 0.07 | 0.06, 0.08 | <0.001 |
| MSSubClass | 0.00 | 0.00, 0.00 | <0.001 |
| YearBuilt | 0.00 | 0.00, 0.00 | <0.001 |
| Neighborhood |  |  |  |
| Blmngtn | — | — |  |
| Blueste | -0.01 | -0.23, 0.21 | >0.9 |
| BrDale | -0.09 | -0.21, 0.02 | 0.11 |
| BrkSide | -0.02 | -0.12, 0.08 | 0.7 |
| ClearCr | 0.15 | 0.06, 0.25 | 0.002 |
| CollgCr | 0.02 | -0.07, 0.10 | 0.7 |
| Crawfor | 0.15 | 0.06, 0.24 | 0.002 |
| Edwards | -0.08 | -0.17, 0.01 | 0.070 |
| Gilbert | -0.04 | -0.13, 0.05 | 0.4 |
| IDOTRR | -0.19 | -0.29, -0.08 | <0.001 |
| MeadowV | -0.05 | -0.15, 0.06 | 0.4 |
| Mitchel | 0.04 | -0.06, 0.13 | 0.4 |
| NAmes | 0.00 | -0.08, 0.09 | >0.9 |
| NoRidge | 0.16 | 0.07, 0.25 | <0.001 |
| NPkVill | 0.08 | -0.05, 0.20 | 0.2 |
| NridgHt | 0.14 | 0.05, 0.22 | 0.001 |
| NWAmes | 0.01 | -0.08, 0.10 | 0.8 |
| OldTown | -0.10 | -0.19, -0.01 | 0.036 |
| Sawyer | 0.00 | -0.09, 0.09 | >0.9 |
| SawyerW | -0.01 | -0.09, 0.08 | >0.9 |
| Somerst | 0.05 | -0.03, 0.13 | 0.2 |
| StoneBr | 0.19 | 0.10, 0.29 | <0.001 |
| SWISU | -0.04 | -0.15, 0.07 | 0.4 |
| Timber | 0.07 | -0.02, 0.16 | 0.13 |
| Veenker | 0.19 | 0.07, 0.30 | 0.002 |
| BldgType |  |  |  |
| 1Fam | — | — |  |
| 2fmCon | 0.15 | 0.07, 0.22 | <0.001 |
| Duplex | -0.03 | -0.08, 0.02 | 0.3 |
| Twnhs | -0.10 | -0.17, -0.02 | 0.011 |
| TwnhsE | -0.03 | -0.09, 0.02 | 0.2 |
| lc\_GrLivArea | 0.42 | 0.38, 0.45 | <0.001 |
| TotalQualityInt | 0.02 | 0.01, 0.02 | <0.001 |
| KitchenQualInt |  |  |  |
| 2 | — | — |  |
| 3 | 0.04 | -0.01, 0.10 | 0.094 |
| 4 | 0.07 | 0.01, 0.13 | 0.015 |
| 5 | 0.15 | 0.09, 0.22 | <0.001 |
| Abbreviation: CI = Confidence Interval | | | |

### Address the Assumptions

library(ggfortify)

library(broom)

# Augment model for residual diagnostics  
aug <- augment(model4)  
resid\_vals <- residuals(model4)  
# 2. Normal Q-Q Plot  
p\_qqplot <- ggplot(aug, aes(sample = .std.resid)) +  
 stat\_qq(color = hexSmuBlue) +  
 stat\_qq\_line(color = hexSmuRed, size = 1) +  
 ggtitle("Normal Q-Q Plot") +  
 xlab("Theoretical Quantiles") +  
 ylab("Standardized Residuals") +  
 theme\_minimal(base\_size = 12) +  
 theme(plot.title = element\_text(face = "bold"))

p\_histogramResiduals <- ggplot(aug, aes(x = .resid)) +  
 geom\_histogram(aes(y = ..density..), bins = 30, fill = hexSmuBlue, color = "white", alpha = 0.8) +  
 stat\_function(fun = dnorm, args = list(mean = mean(resid\_vals), sd = sd(resid\_vals)),  
 color = hexSmuRed, size = 1) +   
 ggtitle("Histogram of Residuals with Normal Curve") +  
 xlab("Residuals") +  
 ylab("Density") +  
 theme\_minimal(base\_size = 12) +  
 theme(plot.title = element\_text(face = "bold"))  
p\_cooks <- ggplot(aug, aes(seq\_along(.cooksd), .cooksd)) +  
 geom\_bar(stat = "identity", fill = hexSmuBlue) +  
 ggtitle("Cook's Distance") +  
 xlab("Observation") +  
 ylab("Cook's distance") +  
 theme\_minimal(base\_size = 12) +  
 theme(plot.title = element\_text(face = "bold"))  
p\_leverage <- ggplot(aug, aes(x = .hat, y = .std.resid)) +  
 geom\_point(color = hexSmuBlue, alpha = 0.6) +  
 geom\_hline(yintercept = 0, color = hexSmuRed) +   
 geom\_smooth(method = "loess", se = FALSE, color = hexSmuRed) +  
 xlab("Leverage") +  
 ylab("Standardized Residuals") +  
 ggtitle("Residuals vs Leverage") +  
 theme\_minimal(base\_size = 12) +  
 theme(plot.title = element\_text(face = "bold"))

# Final Model Assumptions

p\_cooks

A graph with numbers and lines

AI-generated content may be incorrect.

p\_qqplot

A graph with a red line

AI-generated content may be incorrect.

p\_histogramResiduals

A graph of a normal curve

AI-generated content may be incorrect.

p\_leverage

## `geom\_smooth()` using formula = 'y ~ x'

A graph with blue dots and red lines

AI-generated content may be incorrect. # Durbin Watson

library(car)

# Calculate Durbin-Watson statistic  
durbinWatsonTest(model4)

## lag Autocorrelation D-W Statistic p-value  
## 1 -0.002011493 2.003485 0.926  
## Alternative hypothesis: rho != 0

# Create CSVs for Kaggle

# For submission to Kaggle for Kaggle Score  
pred = predict(model1, newdata = testFinal)  
 submission = data.frame(Id = testFinal$Id, SalePrice = exp(pred))  
 write.csv(submission, "C:/submission\_model1\_8.csv", row.names = FALSE)  
pred = predict(model2, newdata = testFinal)  
 submission = data.frame(Id = testFinal$Id, SalePrice = exp(pred))  
 write.csv(submission, "C:/submission\_model2\_8.csv", row.names = FALSE)  
pred = predict(model3, newdata = testFinal)  
 submission = data.frame(Id = testFinal$Id, SalePrice = exp(pred))  
 write.csv(submission, "C:/submission\_model3\_8.csv", row.names = FALSE)  
pred = predict(model4, newdata = testFinal)  
 submission = data.frame(Id = testFinal$Id, SalePrice = exp(pred))  
 write.csv(submission, "C:/submission\_model4\_8.csv", row.names = FALSE)

SAS Feature Selection

/\* SLR \*/

**proc** glmselect data=train plots=all;

**model** lSalePrice = OverallQual/ selection=stepwise(stop=CV) cvmethod=random(5) stats=all cvdetails;

**run;**

**proc** glm data=train plots=all;

**model** lSalePrice = OverallQual;

run;quit;

/\* MLR 1\*/

**proc** glmselect data=train plots=all;

**class** FullBath (ref="1");

**model** lSalePrice = FullBath lc\_GrLivArea

/selection=stepwise(stop=CV) cvmethod=random(5) stats=all cvdetails;

**run;**

**proc** glm data=train plots=all;

**class** FullBath (ref='1');

**model** lSalePrice = FullBath lc\_GrLivArea / solution;

**run;**

**quit;**

/\* MLR 2\*/

**proc** glmselect data=train plots=all;

**class** Neighborhood (ref="NAmes");

**model** lSalePrice = Neighborhood lc\_GrLivArea OverallQual TotalQualityInt AgeofHouse MSSubClass

/ selection=stepwise(stop=CV) cvmethod=random(5) stats=all **cvdetails** showpvalues;

**run;**

**proc** glm data=train plots=all;

**class** Neighborhood (ref="NAmes");

**model** lSalePrice = Neighborhood lc\_GrLivArea OverallQual TotalQualityInt AgeofHouse MSSubClass;

**run;**

**quit;**

/\* MLR 3\*/

**proc** glmselect data=train plots=all;

**class** Neighborhood BldgType KitchenQualInt;

**model** lSalePrice = YrSold OverallQual MSSubClass YearBuilt Neighborhood BldgType Neighborhood lc\_GrLivArea TotalQualityInt KitchenQualInt

/ selection=stepwise(stop=CV) cvmethod=random(5) stats=all **cvdetails** showpvalues;

**run;**

**proc** glmselect data = train plots = all;

**class** Neighborhood (ref="NAmes");

**model** lSalePrice = Neighborhood lc\_GrLivArea OverallQual TotalQualityInt AgeofHouse KitchenQualInt GarageArea MSSubClass / **showpvalues** selection = Stepwise(stop = adjrsq SLE = .2 SLS = .2) stats = adjrsq;

**run;**

/\* SLR: Stepwise External Cross Validation \*/

**proc** glmselect

data=trainClean

testdata=testClean plots(stepaxis=number)=(criterionpanel ASEPlot);

**model** lSalePrice = OverallQual

/ selection=stepwise(choose=rsquare stop=rsquare) stats=all;

**run;**

/\* MLR 1: Stepwise External Cross Validation \*/

**proc** glmselect

data=trainClean

testdata=testClean

plots(stepaxis=number)=(criterionpanel ASEPlot);

**class** FullBath (ref="1");

**model** lSalePrice = FullBath lc\_GrLivArea

/ selection=stepwise(choose=rsquare stop=rsquare) stats=all;

**run;**

/\* MLR 2: Stepwise External Cross Validation \*/

**proc** glmselect

data=trainClean

testdata=testClean

plots(stepaxis=number)=(criterionpanel ASEPlot);

**class** Neighborhood (ref="NAmes");

**model** lSalePrice = Neighborhood lc\_GrLivArea OverallQual TotalQualityInt AgeofHouse MSSubClass

/ selection=stepwise(choose=rsquare stop=rsquare) stats=all;

**run;**

/\* MLR 3: Stepwise External Cross Validation \*/

**proc** glmselect

data=trainClean

testdata=testClean

plots(stepaxis=number)=(criterionpanel ASEPlot);

**class** Neighborhood BldgType KitchenQualInt;

**model** lSalePrice = YrSold OverallQual MSSubClass YearBuilt Neighborhood BldgType lc\_GrLivArea TotalQualityInt KitchenQualInt/ selection=stepwise(choose=rsquare stop=rsquare) stats=all;

**run;**