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 1900 and 2009

- **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.)

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?

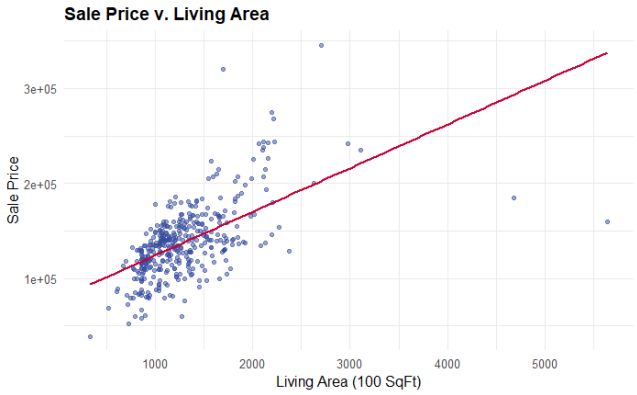
The soundness of conclusions we reach from linear regression depends on the assumptions that the predictors have a linear relationship with the metric we wish to predict, that each observation is independent of each other observation, that errors are normally distributed, and that there is constant variance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

BUILD AND FIT THE MODEL

Predicted log(SalePrice) = $\beta_0 + \beta_1$(log(c_GrLivArea))+ β_2(Edwards)+ β_3(BrkSide)

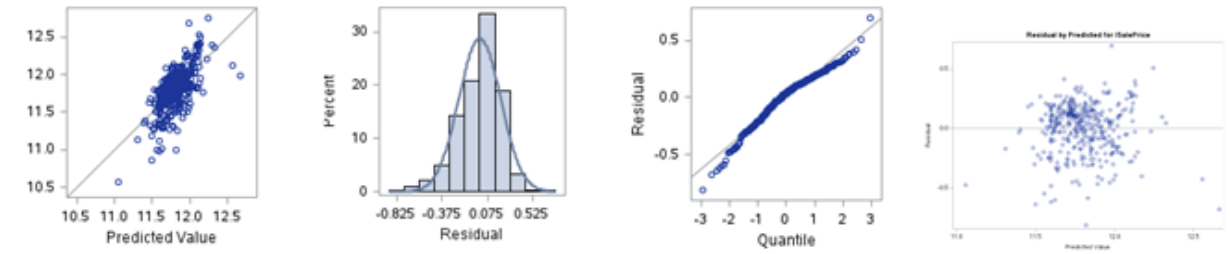
+ β_4(Edwards *log(c_GrLivArea)) + β_5(BrkSide*log(c_GrLivArea))

Untransformed Data



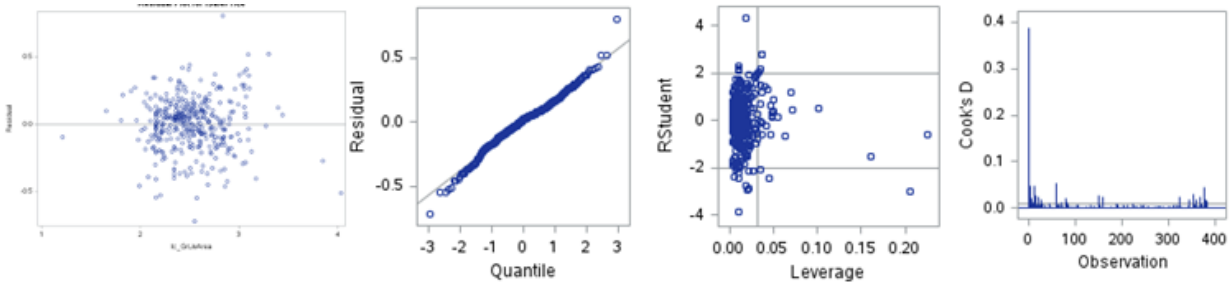
From our scatterplot with a regression line, we see considerable deviation between the observations and the regression. For the untransformed data, we do not see a clear linear relationship. There is a fanning-out of the residuals at the higher range. Thus we pursued a log-transformation of both SalesPrice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

Log-transformed SalePrice and GrLivingArea



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



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.

1. 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).

ANOVA: Does the Relationship Differ by Neighborhood?

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
379	14.57726	NA	NA	NA	NA
377	13.93775	2	0.6395096	8.648997	0.0002125

COMPARING COMPETING MODELS

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

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|	Variance Inflation	95% Confidence Limits	
Intercept	1	10.67108	0.11567	92.26	<.0001	0	10.44365	10.89852
lc_GrLivArea	1	0.47302	0.04543	10.41	<.0001	2.07604	0.38370	0.56235
edw	1	-0.27142	0.18466	-1.47	0.1424	68.15296	-0.63451	0.09168
brk	1	-0.98354	0.21054	-4.67	<.0001	59.00924	-1.39752	-0.56957
edw_sqft	1	0.04664	0.07248	0.64	0.5203	68.63864	-0.09587	0.18916
brk_sqft	1	0.34662	0.08482	4.09	<.0001	57.98240	0.17984	0.51340

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 $2^{0.820} = 1.75$ multiplicative increase in median Sale Price or a 75% increase or a $2^{0.347} = 1.27$ multiplicative increase in median Sale Price from the NAmes and Edwards group. A 95% confidence interval for the multiplicative increase in median of Sales Price after a doubling of 100 sq. ft of living area for the BrkSide neighborhood is a $(2^{0.180}, 2^{0.513}) = (1.13, 1.43) = (13\%, 43\%)$ 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 SalesPrice (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 $2^{0.437} = 1.36$ in the median distribution of Sales Price for the given Ames dataset. In other words, a doubling of 100 sq. ft. of living area increases the estimated median Sales Price by 36%. A 95% confidence interval for the multiplicative increase in median Sales 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 SalePrice. 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/

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 R^2, 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) = $\beta_0 + \beta_1$(OverallQual)

Multiple Linear Regression 1

Predicted log(SalePrice) = $\beta_0 + \beta_1$(GrLivArea)+ β_2(FullBath1)+ β_3(FullBath2) + β_4(FullBath3)

Multiple Linear Regression 2

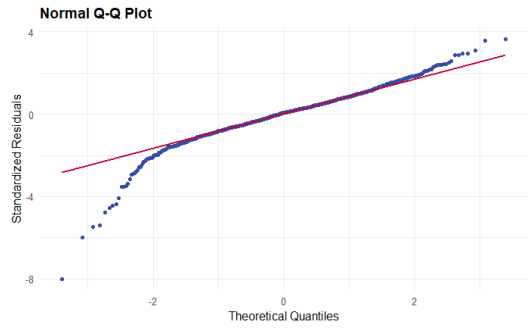
Predicted log(SalePrice) = $\beta_0 + \beta_1$(log(c_GrLivArea)) + β_2(OverallQual) + β_3(TotalQualityInt) + β_4(GarageArea) + β_5(AgeofHouse) + β_6(MSSubClass) + β_7(AgeofHouse) +$\sum \beta_i$(Neighborhood$_i$)

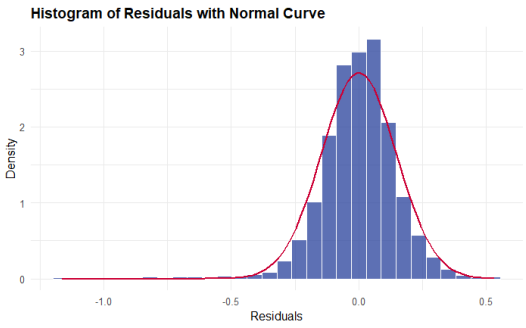
Multiple Linear Regression 3

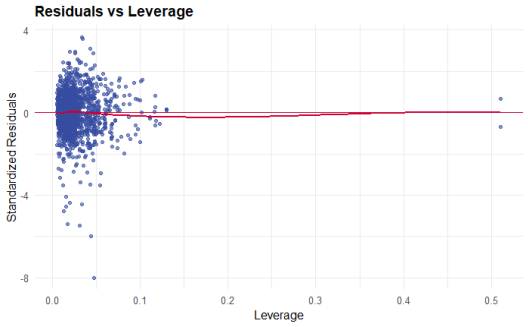
Predicted log(SalePrice) = β_0 + β_1(log(c_GrLivArea)) + β_2(OverallQual) + β_3(TotalQualityInt) + β_4(GarageArea) + β_5(YrSold) + β_6(MSSubClass) + β_7(YearBuilt) + $\sum \beta_i$ (KitchenQualityInt$_i$) +$\sum$ β_i(BldgType$_i$) +$\sum \beta_i$(Neighborhood$_i$)

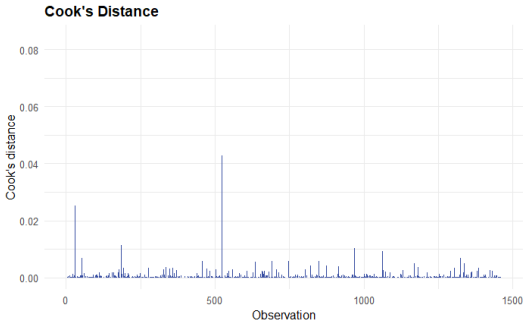
COMPARING COMPETING MODELS

Predictive Models	Adjusted R^2	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









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 R^2 value is the highest for this model in that it explains approximately 86.54% of the variance in log sales 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 is alarming and should be further evaluated as multicollinearity could play a part in this statistic. Furthermore, our lowest AIC was found in the third 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 sales prices in Ames, Iowa for this data.

GitHub Pages Websites:
Chloe Barker: **https://github.com/chloedbarker/chloedbarker.github.io**
Tracy Dower: **https://github.com/tracydower/HomeSales_AmesIowa_Stats**

APPENDIX

R and Graphics

Install and Load Required Packages

```
packages <- c("olsrr","ggplot2","boot","flextable","gtsummary","labelled","ov
erviewR","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 cod
e

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, Gar
ageFinish, GarageQual, GarageCond), ~ifelse(GarageCars == 0, NA, .)))
combinedData <- combinedData %>% mutate(across(c(GarageType, GarageFinish, Ga
rageQual, GarageCond), ~ifelse(GarageCars == 0, NA, .))) %>%
 mutate(GarageYrBlt = ifelse(GarageCars == 0, NA, GarageYrBlt))
# Where ID = 2760, GarageYrBlt = 2207 which is nonsense, and YearBuilt (the y
ear 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 st
ill be NA
combinedData[combinedData$Id == 2760, "GarageYrBlt"] <- 2007
combinedData$GarageArea[is.na(combinedData$GarageArea)] <- 0
combinedData$GarageArea <- ifelse(is.na(combinedData$GarageArea),0,combinedDa
ta$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, com
binedData$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)

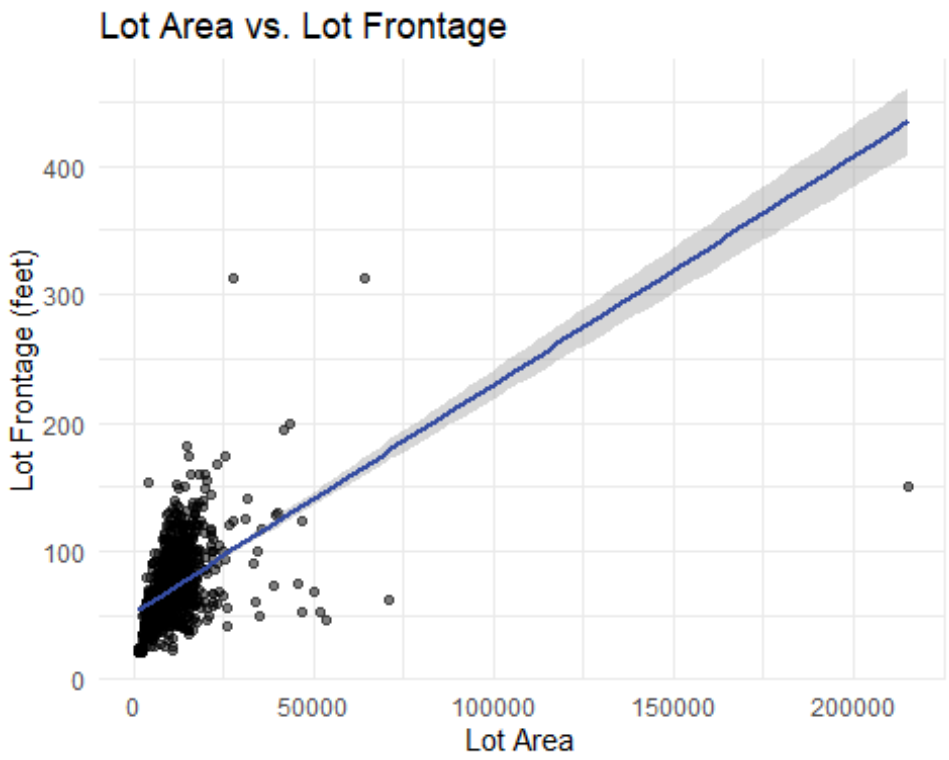
How long before the sale of the home was it last built or remodeled?

```
combinedData$FreshDate = (combinedData$YrSold - pmax(combinedData$YrSold, com
binedData$YearRemodAdd))
# Not significant. Do not use.
combinedData$YearFromRemodelToSale = (combinedData$YrSold - combinedData$Fres
hDate)
combinedData$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()
```




```
model <- lm(LotFrontage ~ LotArea + GrLivArea + LotShape + Neighborhood, data
= combinedData[!is.na(combinedData$LotFrontage), ])
 # summary(model)
model <- lm(LotFrontage ~ (LotArea/LotFrontage) + GrLivArea + Neighborhood, d
ata = 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 Po
ol", 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", "BsmtQ
ual", "BsmtFullBath", "BsmtHalfBath")
 combinedData[rows,cols] <-NA
# Where BsmtTotalSF > 0, if BsmtExposure is NA, that's a data entry error, an
d it should be "No"
 combinedData <- combinedData %>% mutate(BsmtExposure = ifelse(is.na(BsmtExpo
sure) & BsmtTotalSF > 0, "No", BsmtExposure))
 combinedData <- combinedData %>% mutate(BsmtFinType1 = ifelse(is.na(BsmtFinS
F1) | BsmtFinSF1 == 0, "Unf", BsmtFinType1))
# Where BsmtFinSF2 >0 and BsmtFinType2  NA, set BsmtFinType2="Unf"
 combinedData <- combinedData %>% mutate(BsmtFinType2 = ifelse(is.na(BsmtFinT
ype2) & BsmtFinSF2 > 0, "Unf", BsmtFinType2))
# count rows where at least one of these fields is NA but not ALL of these fi
elds 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 MasVnrAre
a = NA and MasVnrType = "None" to MasVnrArea = 0.
combinedData <- combinedData %>% mutate(MasVnrArea = ifelse(is.na(MasVnrArea)
, 0, MasVnrArea))
combinedData <- combinedData %>% mutate(MasVnrType = ifelse(MasVnrArea==0, "N
one" , 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(MiscFeatur
e ), "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", "B
smtExposure", "BsmtFinSF1", "BsmtFinSF2",
 "BsmtFinType1", "BsmtFinType2", "BsmtFullBath",
 "BsmtHalfBath", "BsmtQual", "BsmtTotalSF", "BsmtUnfSF"))]
FinalVariables <- c("DataSet","Id","SalePrice","lSalePrice", "AgeofHouse", "B
ldgType", "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", "Firepl
aceQu", "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$KitchenQualInt > 1, combi
nedData$KitchenQualInt, 2)
 combinedData <- combinedData %>% mutate(TotalQualityInt = ExterQualInt + Ext
erCondInt + PoolQCInt + GarageQualInt + GarageCondInt + HeatingQCInt + Firepl
aceQuInt + 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[co
mbinedData$BedroomAbvGr > 0, ])
bed_rows <- which(combinedData$BedroomAbvGr == 0)
combinedData$BedroomAbvGr[bed_rows] <- round(predict(bed_model, newdata = com
binedData[bed_rows, ]))
bath_model <- lm(FullBath ~ GrLivArea + HouseStyle + BedroomAbvGr, data = com
binedData[combinedData$FullBath > 0, ])
bath_rows <- which(combinedData$FullBath == 0)
```

```
combinedData$FullBath[bath_rows] <- round(predict(bath_model, newdata = combi
nedData[bath_rows, ]))
# str(combinedData$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 |

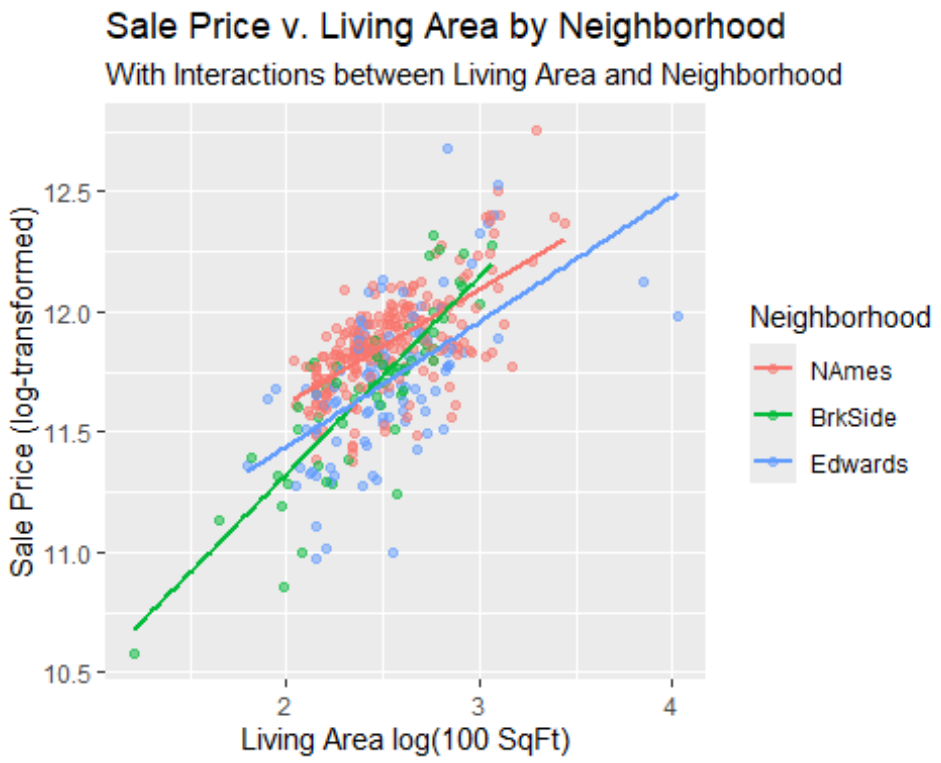
```
exterior_map <- function(x) {
 case_when(
 x %in% c("Stone", "Stucco") ~ "StoneOrStucco",
 x %in% c("Wd Sdng", "Wd Shng", "WdShing") ~ "WoodShingle",
 x %in% c("Brick", "Brk Cmn", "BrkComm", "BrkFace") ~ "Brick",
 TRUE ~ "Other" # all other cases will be classified as "other"
 )
}
combinedData$ExteriorSimplified <- exterior_map(combinedData$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", "Edwar
ds", "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"))
```


```
modelCentury21Ames_SLR <- lm(lSalePrice ~ lc_GrLivArea, data = dataCentury21A
mes)
# 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 = Neigh
borhood)) +
 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 = "Livi
ng Area log(100 SqFt)", y = "Sale Price (log-transformed)" )
```




```
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")
```

ANOVA: Does the Relationship Differ by Neighborhood?

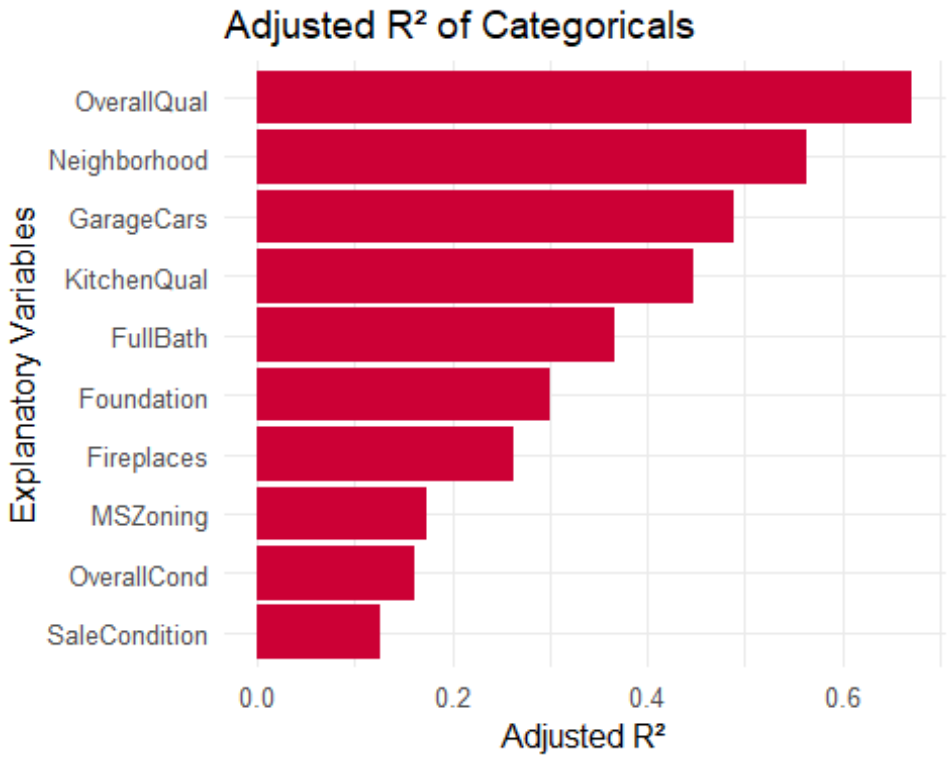
Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
379	14.57726	NA	NA	NA	NA
377	13.93775	2	0.6395096	8.648997	0.0002125

ANALYSIS 2

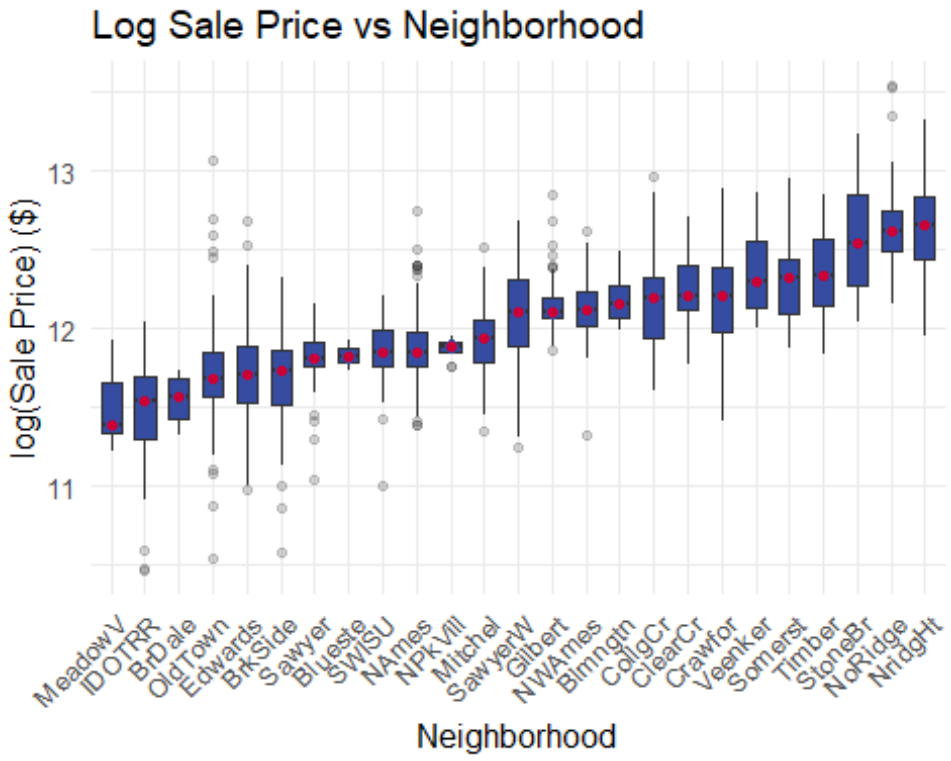
Explore other vairables

```
library(patchwork)
train_clean <- combinedData %>% filter(DataSet == "train")
cat_vars <- c("Alley", "BedroomAbvGr", "BldgType", "CentralAir", "Electrical"
, "ExterCond",
 "Fireplaces", "Foundation", "FullBath", "GarageCars", "GarageQual", "HalfBat
h",
 "HouseStyle", "KitchenQual", "LandContour", "LandSlope", "MSZoning", "Neighb
```

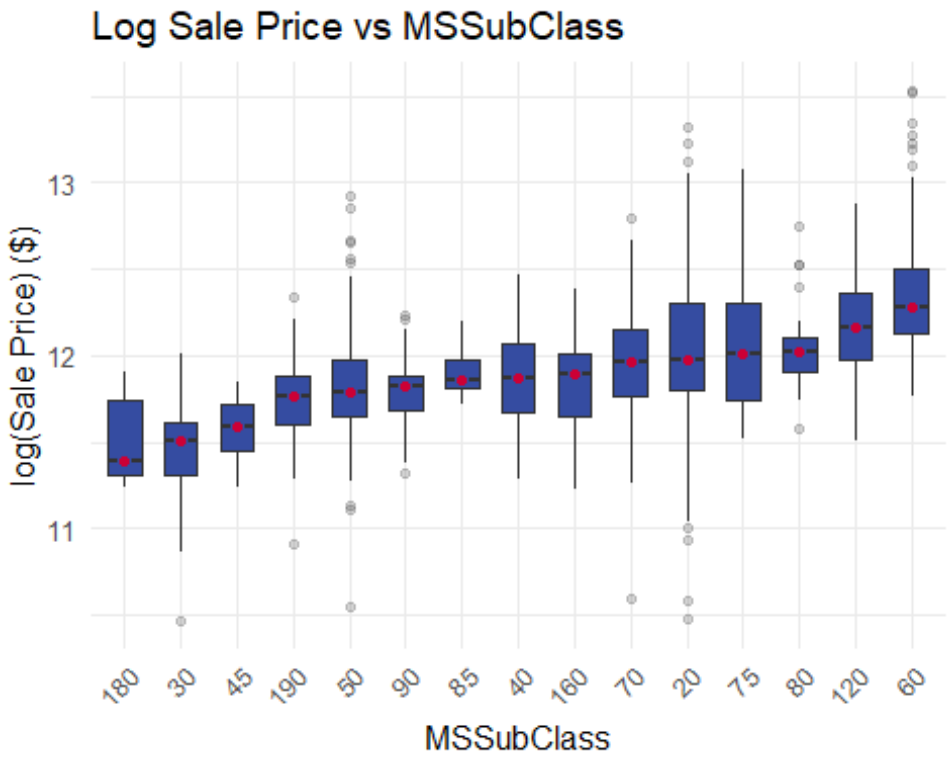
```
orhood",
 "OverallCond", "OverallQual", "PoolQC", "RoofStyle", "SaleCondition", "SaleT
ype",
 "Utilities")
num_vars <- c("MiscVal", "WoodDeckSF", "OpenPorchSF", "EnclosedPorch","Screen
Porch","PoolArea", "GarageArea", "LotFrontage",
 "LotArea", "MasVnrArea", "TotalBsmtSF", "YearBuilt", "YearRemodAdd", "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.char
acter(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)
```



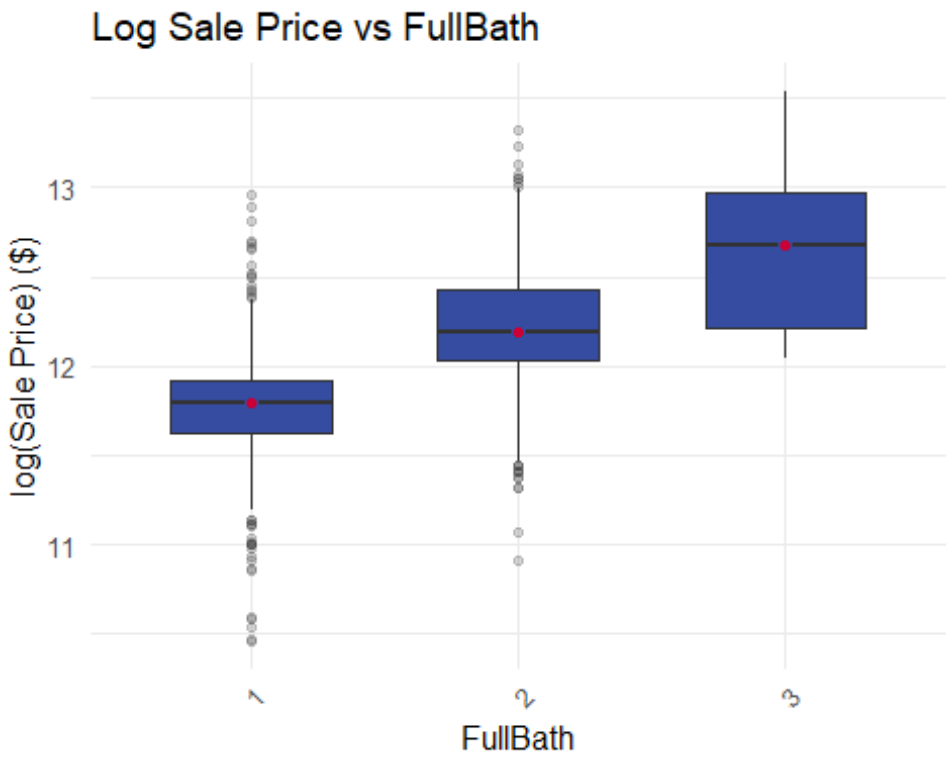

```
## Specific Categorical Variable Graphs
plot_box = function(df, var) {
 medians = df %>% group_by(.data[[var]]) %>% summarize(median_price = median(
lSalePrice, na.rm = TRUE)) %>%
 arrange(median_price)
 df[[var]] = factor(df[[var]], levels = medians[[var]])
 ggplot(df, aes(x = .data[[var]], y = lSalePrice)) +
 geom_boxplot(outlier.alpha = 0.2, width = 0.6, fill = hexSmuBlue) +
 stat_summary(fun = median, geom = "point", shape = 21, size = 2, fill = hexS
muRed, 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 = "n
one")
}
# Insignificant variation
 plot_box(train_clean, "Neighborhood")
```



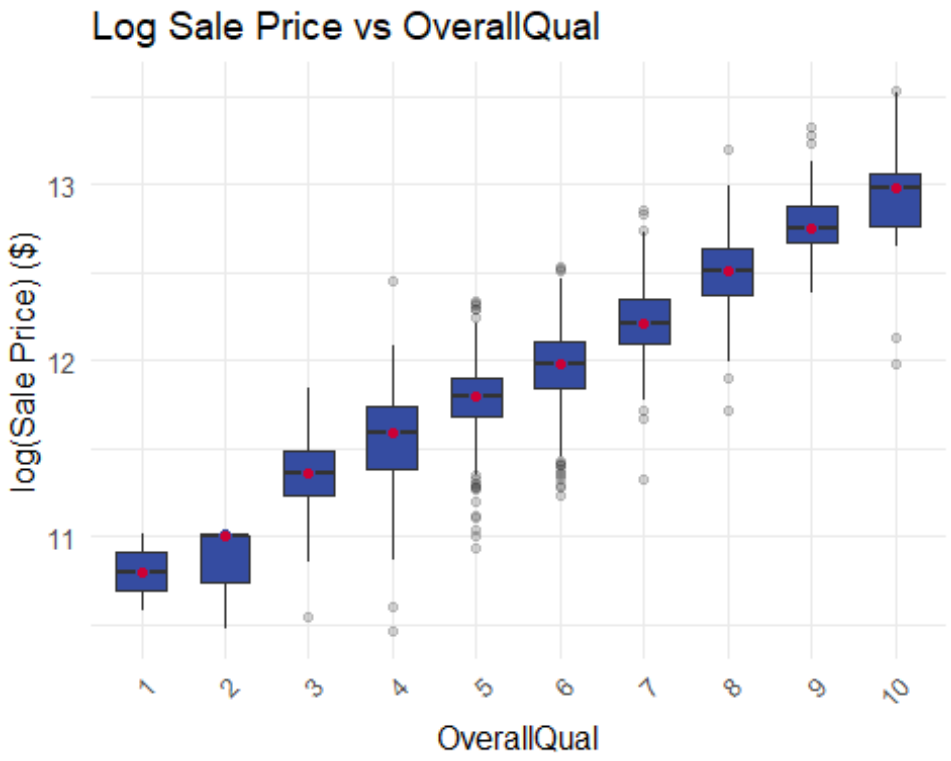

```
plot_box(train_clean, "MSSubClass")
```



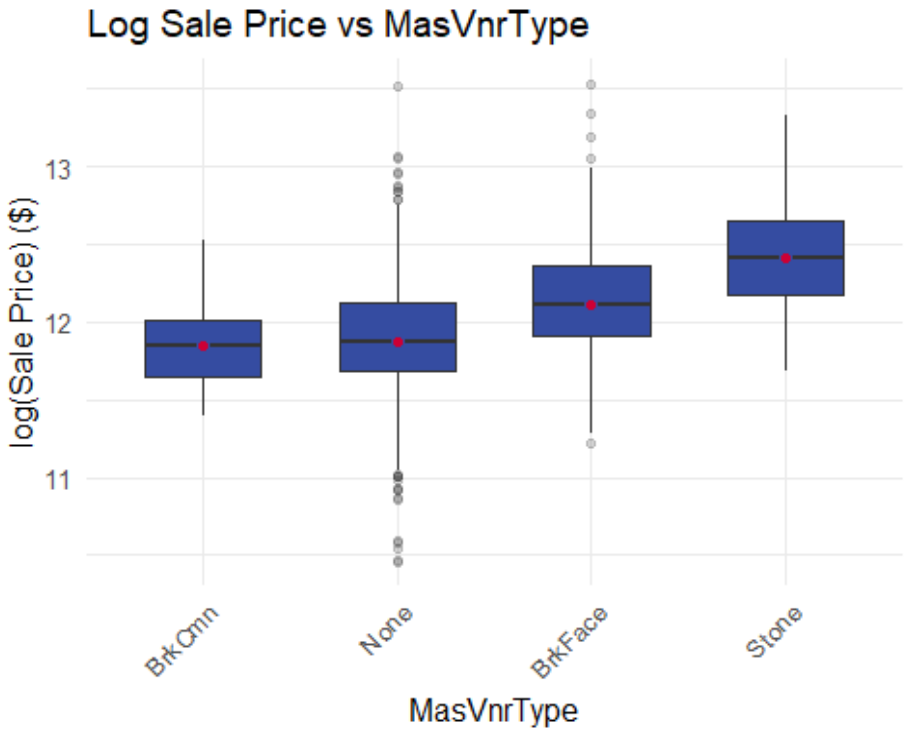

```
plot_box(train_clean, "FullBath")
```



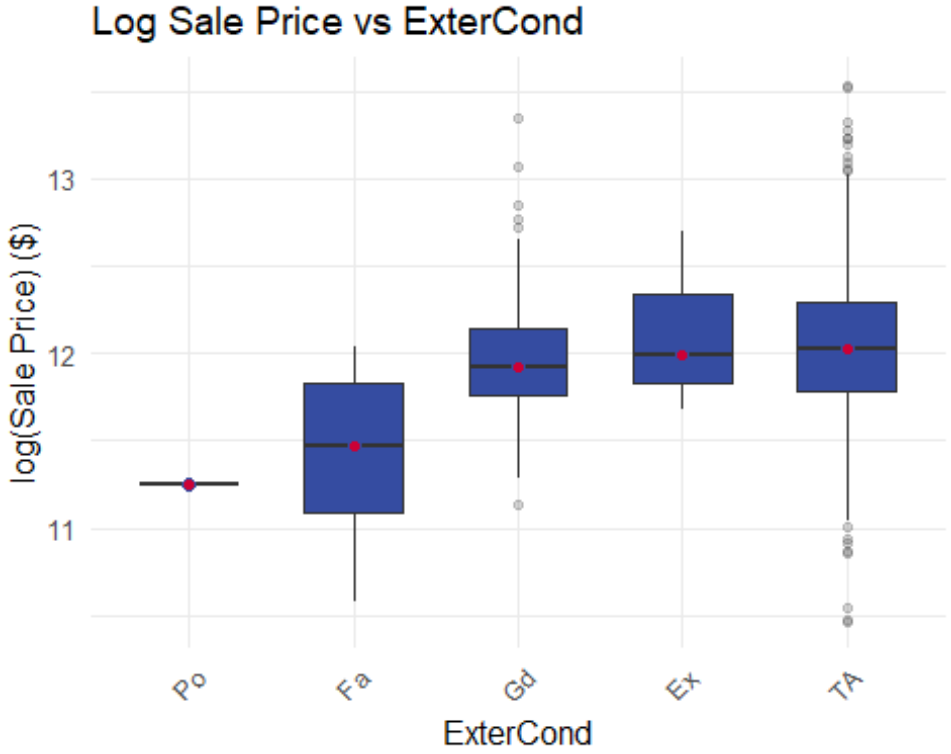

```
plot_box(train_clean, "OverallQual")
```



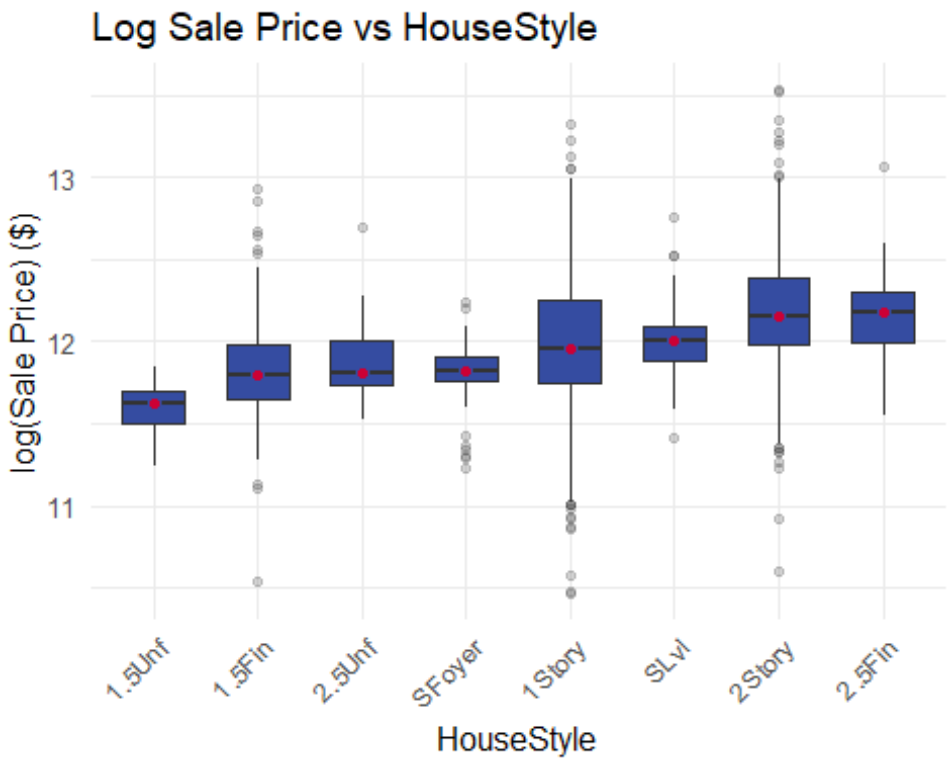
```r
# significant variation between levels
 plot_box(train_clean, "MasVnrType")
```



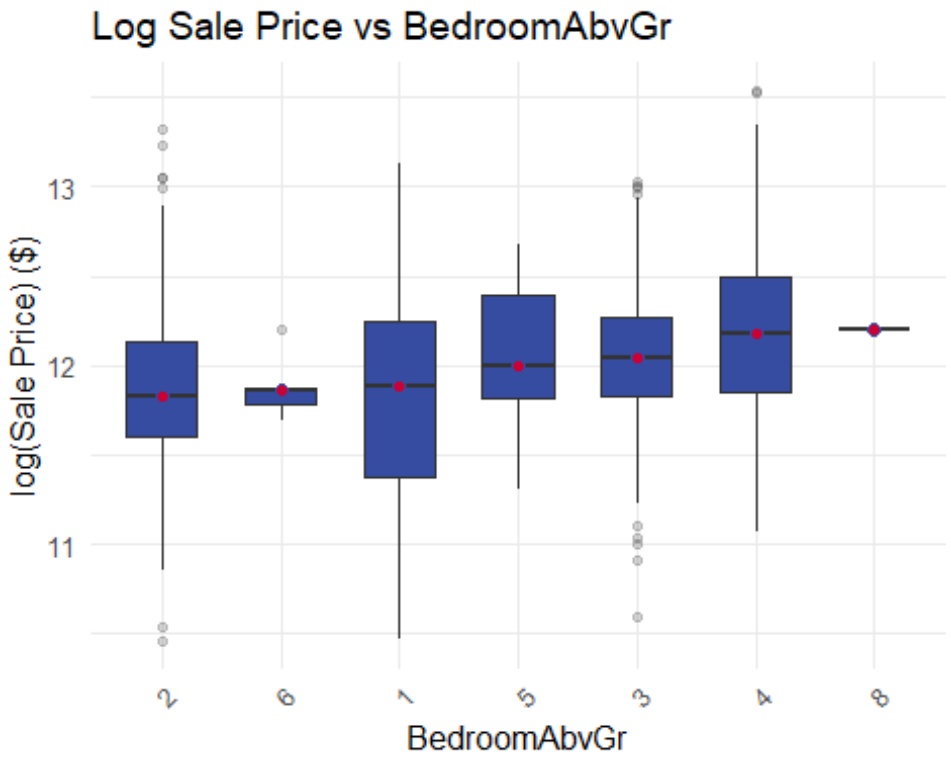

```r
 plot_box(train_clean, "ExterCond")
```



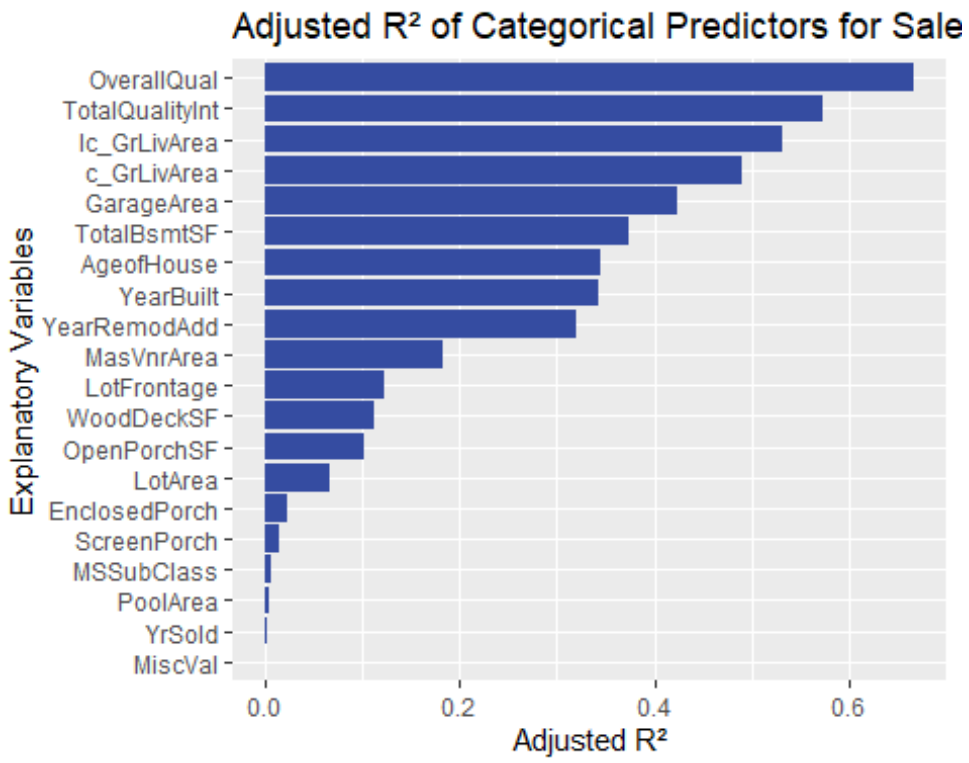
```
plot_box(train_clean, "HouseStyle")
```



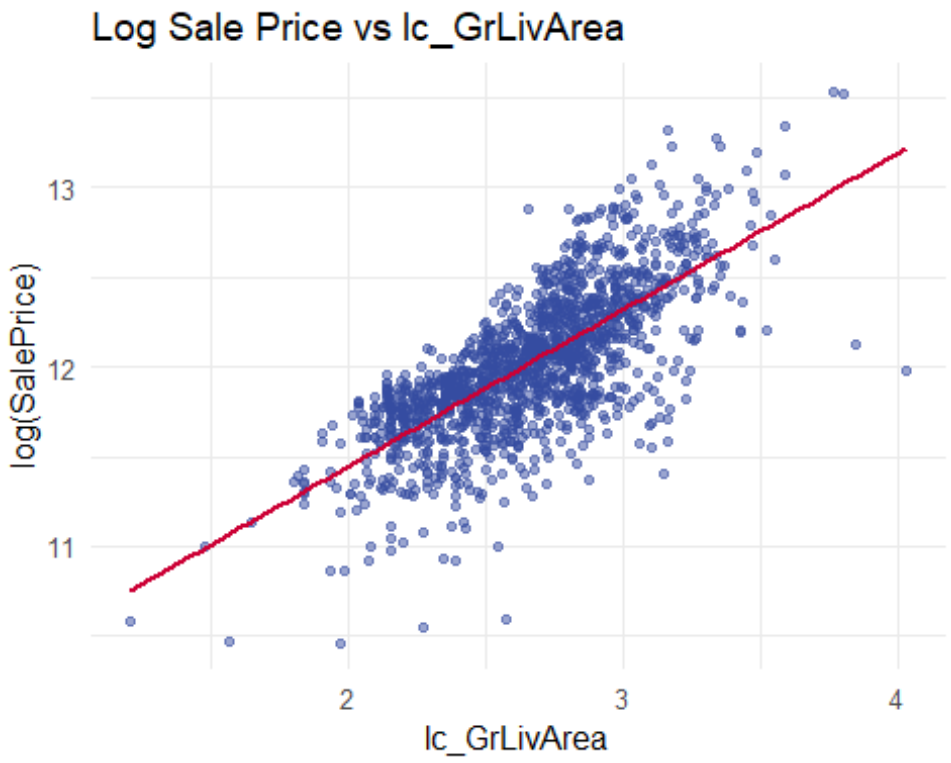

```
plot_box(train_clean, "BedroomAbvGr")
```



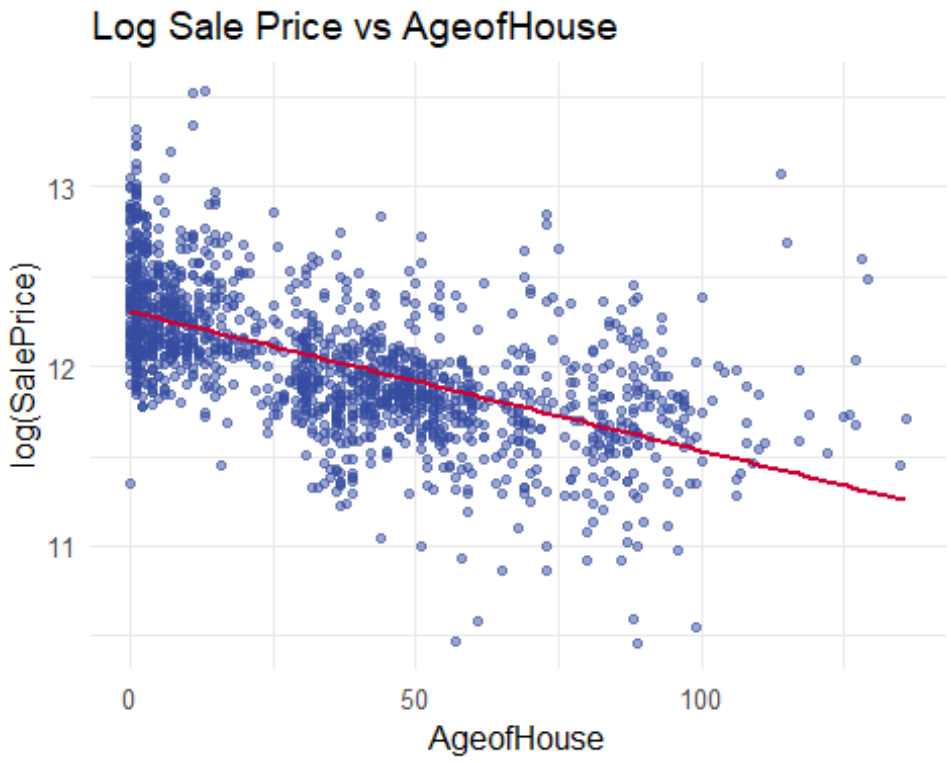
```r
 ## Numeric Variables Graph
train_clean <- train_clean %>% mutate(across(all_of(num_vars), as.numeric))
num_var_stats <- map_df(num_vars, function(v) {
 f = as.formula(paste("lSalePrice ~", v))
 mod = lm(f, data = train_clean)
 tibble(var = v,adj_r2 = summary(mod)$adj.r.squared)
})
num_var_stats %>%
 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 Sales Price")
```



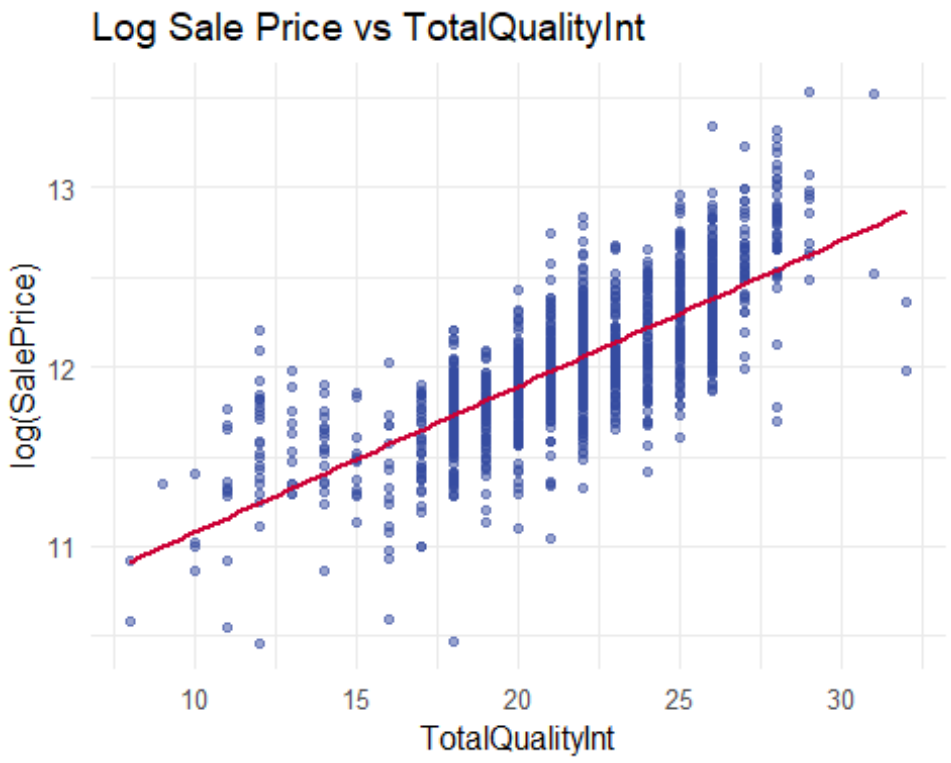

```r
 theme_minimal(base_size = 12)

## 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")
```



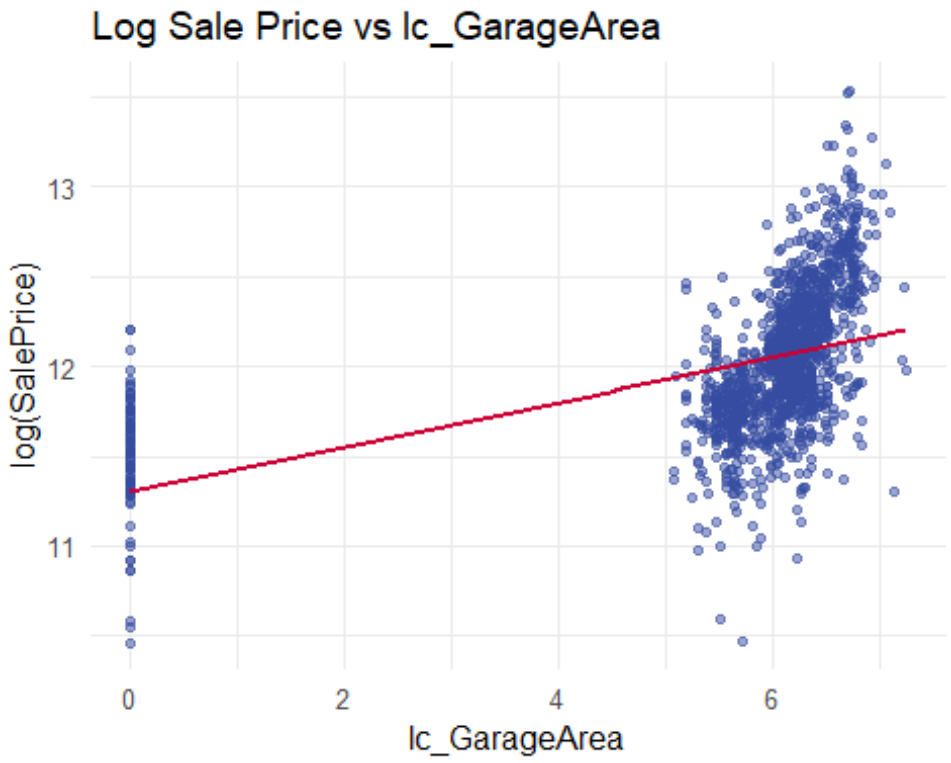

```
plot_scatter(train_clean, "AgeofHouse")
```



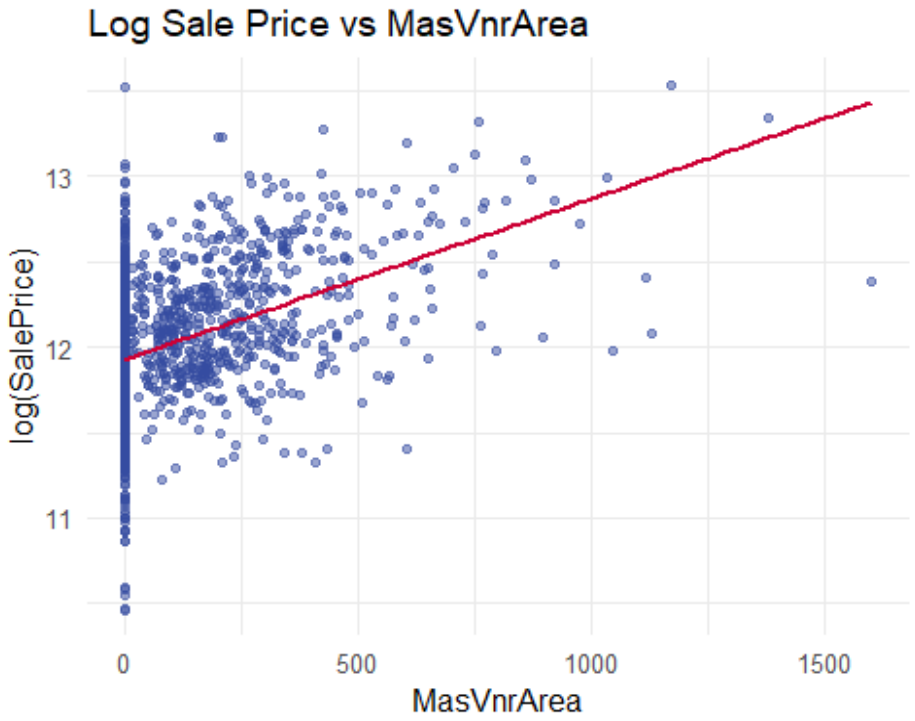

```
plot_scatter(train_clean, "TotalQualityInt")
```



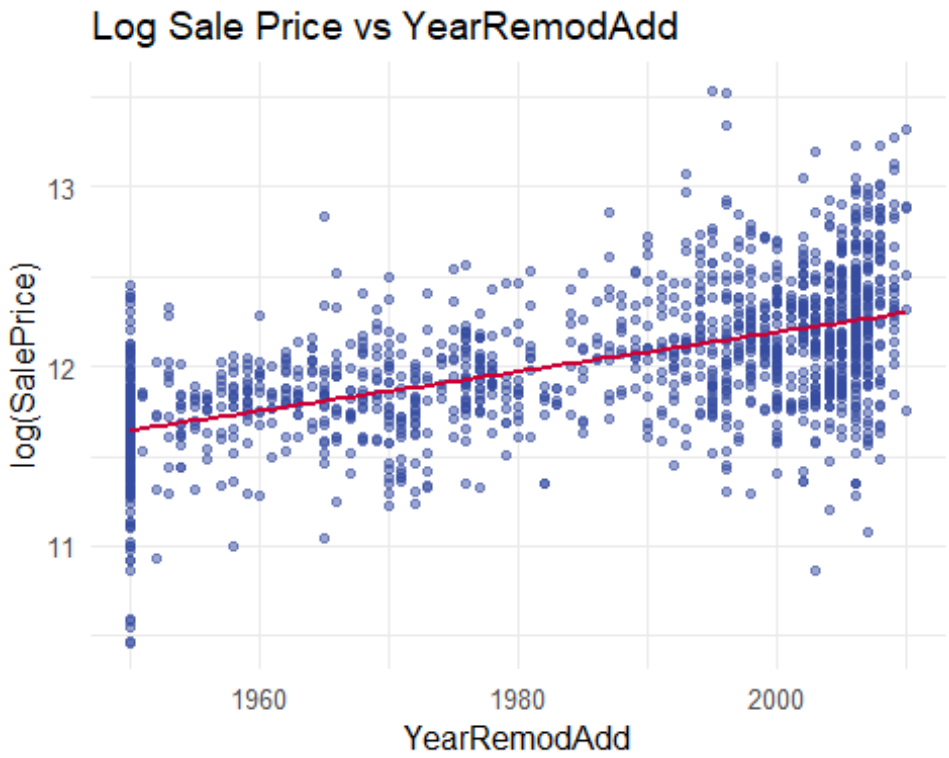

```r
plot_scatter(train_clean, "lc_GarageArea")
```



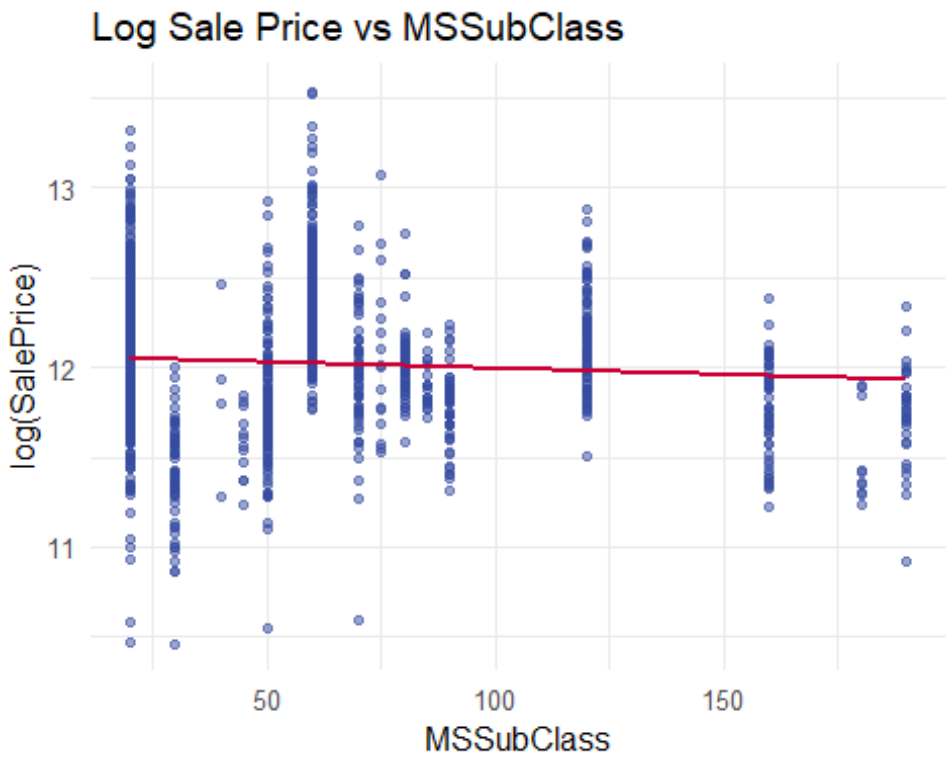
```r
# Insignificant Variation Visually
plot_scatter(train_clean, "MasVnrArea")
```




```r
plot_scatter(train_clean, "YearRemodAdd")
```



```
plot_scatter(train_clean, "MSSubClass")
```



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 + Firepla
ce + 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 = myMode
lName)
 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 1: 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 2: 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 3: 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 + Neighborhood + 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 4: 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	—	—	

Characteristic	Beta	95% CI	p-value
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

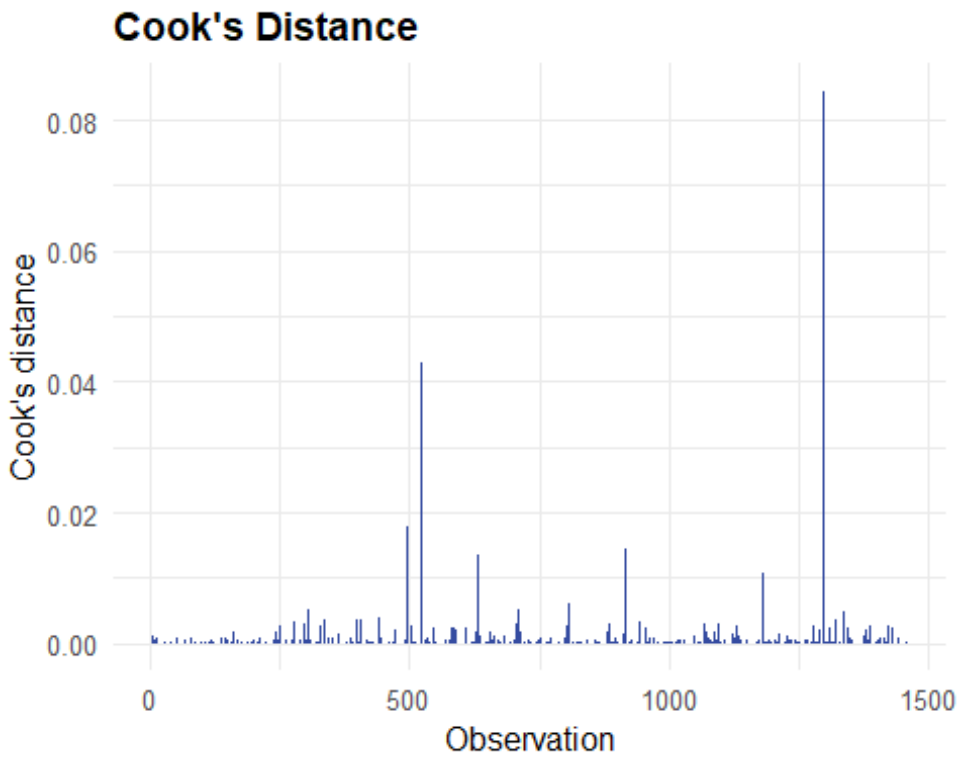
Address the Assumptions

```
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(resi
d_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") +
```

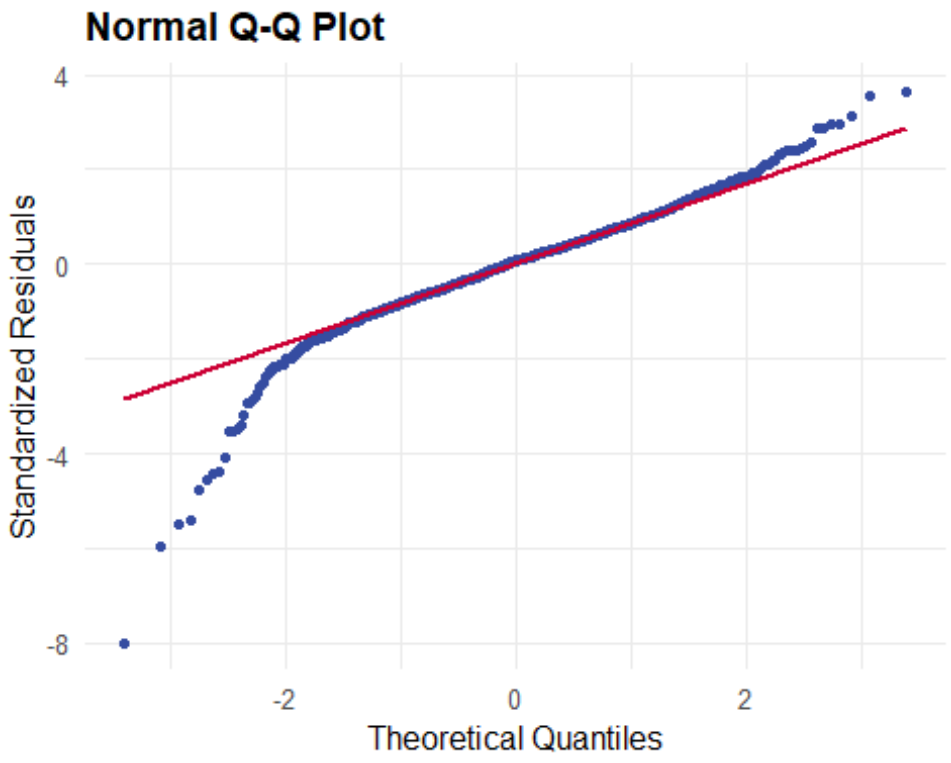
```
  theme_minimal(base_size = 12) +
  theme(plot.title = element_text(face = "bold"))
```

Final Model Assumptions

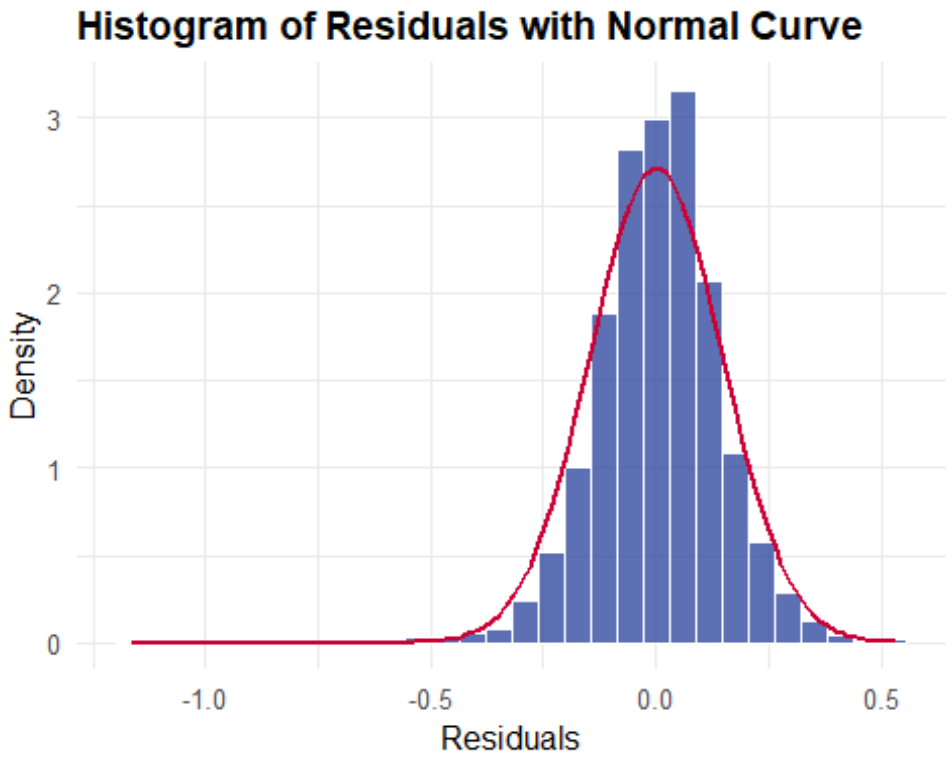
```
p_cooks
```




```
p_qqplot
```

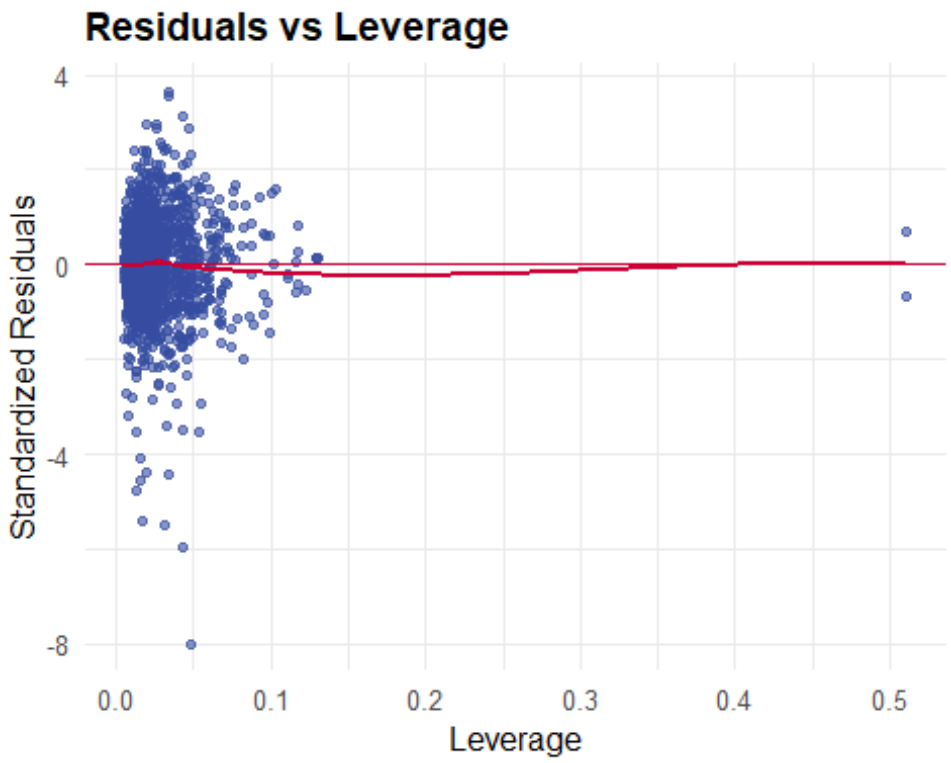



```
p_histogramResiduals
```




```
p_leverage
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



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;
```