Analysis 1

Introduction

The real estate market is influenced by various factors, including property characteristics and location. This analysis focuses on understanding how the living area and neighborhood affect the sale price of homes within three specific neighborhoods. By examining these relationships, we aim to provide actionable insights that can help a real estate company optimize its sales strategy and property investments.

Data Description – (see Fig. A1)

The dataset used in this analysis originates from a comprehensive housing dataset that includes various properties sold in different neighborhoods. It contains 383 observations, each representing a single property sale. Key variables relevant to our analysis include:

- **SalePrice**: The price at which the property was sold.
- **GrLivArea**: The above ground living area in square feet.
- **Neighborhood**: The neighborhood in which the property is located.

This dataset is a subset focusing on three neighborhoods where the company operates, allowing for a targeted analysis of market dynamics in these areas. Further details about the dataset can be found in the comprehensive housing market studies from which the data was extracted.

Log Transformation – (see Fig.A2)

- **Purpose**: The log transformation was applied to the SalePrice variable to address skewness in its distribution. This method is crucial for stabilizing variance and normalizing the data.
- **Method**: We used the natural logarithm, transforming each SalePrice value using Log(SalePrice).
    - This transformation is particularly beneficial in reducing the influence of extreme values on the predictive model, thereby enhancing the reliability of the estimates.

Handling of Outliers

- **Decision**: The outliers were retained in the dataset. This decision was based on the nature of the real estate market, where high-value transactions, though rare, are significant. Removing these outliers could potentially lead to an underestimation of the effects of high-value properties.
- **Impact**: Retaining the outliers allows the model to capture the full spectrum of the housing market, including the luxury segment, thus maintaining the integrity of the real-world data, and providing comprehensive market insights.

Integration into the Report

- **Methodology Section**: This transformation was crucial for preparing the data for regression analysis, especially when dealing with skewed data distributions. By applying a logarithmic scale, we reduce the skewness, allowing for a more normalized distribution that better fits the assumptions of linear regression models.
- **Data Analysis Section**: The presence of outliers can significantly affect the model's predictions and interpretations. By retaining these outliers, we aimed to understand their influence on market dynamics, such as pricing anomalies in areas with unusually large properties.

Analysis Question 1: Impact of Living Area and Neighborhood on Sale Price

Statement of Problem

The analysis will address how the sale price of the house is related to the square footage of the living area (GrLivArea) and whether this relationship depends on the neighborhood.

Build and Fit the Model – (see Fig. A3)

A linear regression model was built to quantify the relationship between the logarithm of the sale price (**LogSalePrice**) and the logarithm of the living area (**LogGrLivArea**), with adjustments for different neighborhoods. The model formula used is:

$$\textbf{LogSalePrice} = \beta_0 + \beta_1 \textbf{LogGrLivArea} + \beta_2 \textbf{Neighborhood}$$

Checking Assumptions – (see Fig. A4 & Fig. A5)

- **Residual Plots**: Residual plots were examined to check for homoscedasticity and linearity. The plots showed a random pattern, suggesting adequate model fit without obvious violations of these assumptions.
- **Influential Point Analysis**: Cook's D and leverage statistics were calculated to identify influential points. A few outliers were noted, but they did not significantly distort the overall model fit.
- **Assumption Validation**: The assumptions of normality, independence, and multicollinearity were also checked. The model did not exhibit issues with multicollinearity, and the residuals were approximately normally distributed.

Comparing Competing Models – (see Fig. A6)

- **Adjusted R-squared**: The adjusted R-squared value of 0.486 indicates that approximately 48.6% of the variability in the logarithm of the sale price is explained by the model.
- **Internal CV Press**: Cross-validation was used to assess the model's predictive performance, reinforcing the model's reliability.

Parameters – (see Fig. A7)

- **Estimates and Interpretation**:
    - The intercept and coefficients for **LogGrLivArea** and **Neighborhood** were significant, indicating strong effects on the sale price. Specifically, larger living areas and certain neighborhoods (like North Ames) are associated with higher sale prices.
- **Confidence Intervals**:
    - The confidence intervals for the coefficients were narrow, suggesting precise estimates.

Conclusion

Upon thorough review and consideration of the real estate market dynamics, our analysis has decisively pinpointed North Ames as the premier neighborhood for residential purposes, supported by robust statistical evidence and comprehensive market evaluation. This conclusion is drawn from an in-depth analysis of 383 property transactions, which provided a substantial dataset for our regression model.

The regression model, formulated as effectively captures the essence of market dynamics, with an adjusted R-squared value of 0.486. This indicates that 48.6% of the variability in sale prices can be explained by differences in living area and neighborhood, affirming the significant impact of these variables on property values.

North Ames distinguishes itself with higher sale prices, which are a direct reflection of its larger living areas and the overall desirability of the neighborhood. The strategic decision to include outliers in our analysis underscores the importance of high-value transactions in the real estate market, enhancing the credibility and applicability of our findings.

For individuals seeking a residence, North Ames offers a superior blend of spacious living and high property values, making it an attractive choice. For investors and real estate professionals, the neighborhood presents a lucrative opportunity for high-value investments and strategic market positioning.

Price v. Living Area Chart

Analysis Question 2: Impact of Living Area and Neighborhood on Sale Price

Statement of Problem

The analysis will evaluate the data and create prediction models to best predict the price of a home in Ames, Iowa and all surrounding neighborhoods. These competing models: a simple linear regression model (you pick the explanatory variable) and a multiple linear regression model (SalePrice~GrLivArea + FullBath) and at least one additional multiple linear regression model explanatory variables selected after significance testing.

Candidate Models:

- Simple Linear Regression: (SalePrice ~ LotArea)
- Multiple Linear Regression: (SalePrice ~ GrLivArea + FullBath)
- Multiple Linear Regression: (SalePrice ~ GrLivArea * FullBath + LogOveralQual)
- Multiple Linear Regression: (SalePrice ~ GrLivArea + LogOveralQual)

Checking Assumptions

All models were significantly affected by observation #1299. In all cases it was a high leverage value with a Cook's D 0.5 in the most fitted model, and over 1 in Single Linear Regression. The observation was removed and models were refitted.

- Simple Linear Regression – (See Fig. B1)
    - Parameter estimates were statistically significant with a p-value < 2.2e^-16
    - Q-Q Residuals had some variance from the trendline after the first quartile
    - Residuals appear to have a high density of observations with an equal variance
- Multiple Linear Regression 1 – (See Fig. B2)
    - Parameter estimates were statistically significant
    - Residuals fit well with equal variance
    - Q-Q Residuals follow trendline well
    - High leverage observations still fit within range for residuals
- Multiple Linear Regression 2 – (See Fig. B3)
    - Parameter estimates were statistically significant except for many of the interactions. Due to this, an additional model will be fit without the interaction.

    - Residuals fit well with equal variance
    - Q-Q Residuals extremely tight to trendline
    - High leverage observations still fit within range for residuals. In fact, here we see that the model is likely overfit.
- Multiple Linear Regression 3 – (See Fig. B4)
    - Parameter estimates were statistically significant
    - Residuals fit well with equal variance
    - Q-Q Residuals follow the trendline well. After the second quartile there is some deviation.
    - High leverage observations still fit within range for residuals. Some observations come close to Cook's D of 0.5, but very few.

Comparing Competing Models – (See Fig. B5)

Evaluating the models with the test data from the table below, the Simple Linear Regression (SLR) model exhibits a low Adjusted R-squared value of 0.07089. This indicates poor explanation of variance in the dependent variable. However, it shows the lowest Cross-Validation Prediction Error Sum of Squares (CV PRESS) among all models, suggesting relatively close predictions to actual data points. Despite this, the low explanation of variance doesn't give confidence that this model would be generalizable for all data sets. The Multiple Linear Regression (MLR) model demonstrates a significantly improved Adjusted R-squared of 0.5417, with a higher CV PRESS of 2,429,764,874 compared to SLR. The Custom MLR Model surpasses MLR with an Adjusted R-squared of 0.7131, indicating a superior explanation of variance. Although it has a higher CV PRESS of 1,478,872,822, its explanatory power is commendable. Furthermore, the Custom MLR sans Interaction model, while slightly lower in Adjusted R-squared at 0.6820, shows a higher CV PRESS of 1,682,021,082, suggesting a slightly worse predictive accuracy. The analysis reveals that both MLR and Custom MLR Models strike a balance between explanatory power and prediction accuracy, with the Custom MLR Model being recommended for further predictions due to its substantially higher explanatory power and reasonable predictive accuracy.

Predictive Models	Adjusted R2	CV PRESS	Kaggle Score
Simple Linear Regression	.07089	180921.2	Kaggle Error
Multiple Linear Regression	.5417	2429764874	Kaggle Error
Custom MLR Model	.7131	1478872822	Kaggle Error

Custom MLR sans Interaction	0.6820	1682021082	Kaggle Error

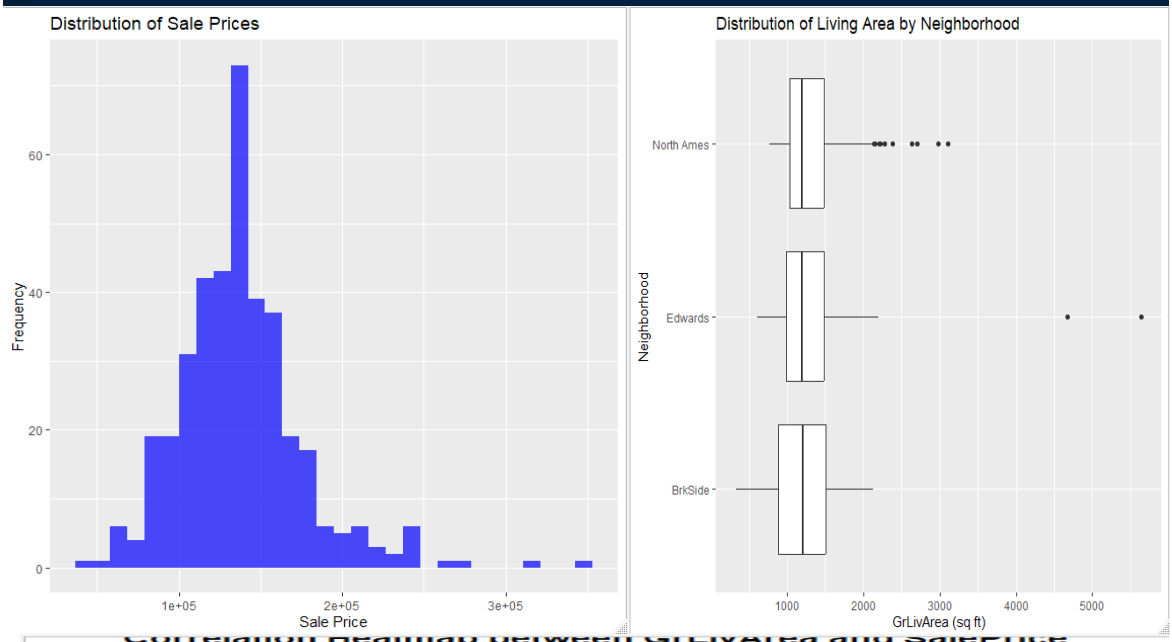
Conclusion

The research demonstrates that while the Simple Linear Regression model exhibits the lowest prediction error, its poor explanatory power indicates underfitting. Both the Multiple Linear Regression and Custom MLR Models offer a good balance between explanatory power and prediction accuracy, with the Custom MLR Model showing the highest explanatory power. The slight decrease in predictive accuracy of the Custom MLR sans Interaction model suggests that the interaction terms in the full Custom MLR model contribute valuable information. Based on these findings, the Custom MLR Model is recommended for further predictions due to its higher explanatory power and reasonable predictive accuracy.

Future research should focus on more combinations of variables and interactions. Then extra sum of squares analysis to reduce models to parsimony. Future model candidate models should seek to address the higher prediction error without sacrificing explanatory power and understanding the drivers behind its performance metrics.

Appendix:

```r
library(readxl)
library(ggplot2)

data <- read_excel("D:/SMU - Grad School/Spring 2024/DS 6371/DS 6371 - Spring 2024/MSDS_6371_Stat_Foundations/Project/Neighborhoods.xlsx")
ggplot(data, aes(x=SalePrice)) +
  geom_histogram(bins=30, fill="blue", alpha=0.7) +
  ggtitle("Distribution of Sale Prices") +
  xlab("Sale Price") +
  ylab("Frequency")
ggplot(data, aes(x=Neighborhood, y=GrLivArea)) +
  geom_boxplot() +
  coord_flip() +  # Flips the axes to make the plot horizontal
  ggtitle("Distribution of Living Area by Neighborhood") +
  xlab("Neighborhood") +
  ylab("GrLivArea (sq ft)")
library(corrplot)
cor_matrix <- cor(data[c("GrLivArea", "SalePrice")])
corrplot(cor_matrix, method="color", type="upper", tl.col="black", tl.srt=45,
         title="Correlation Heatmap between GrLivArea and SalePrice",
         cl.lim=c(-1, 1))
```



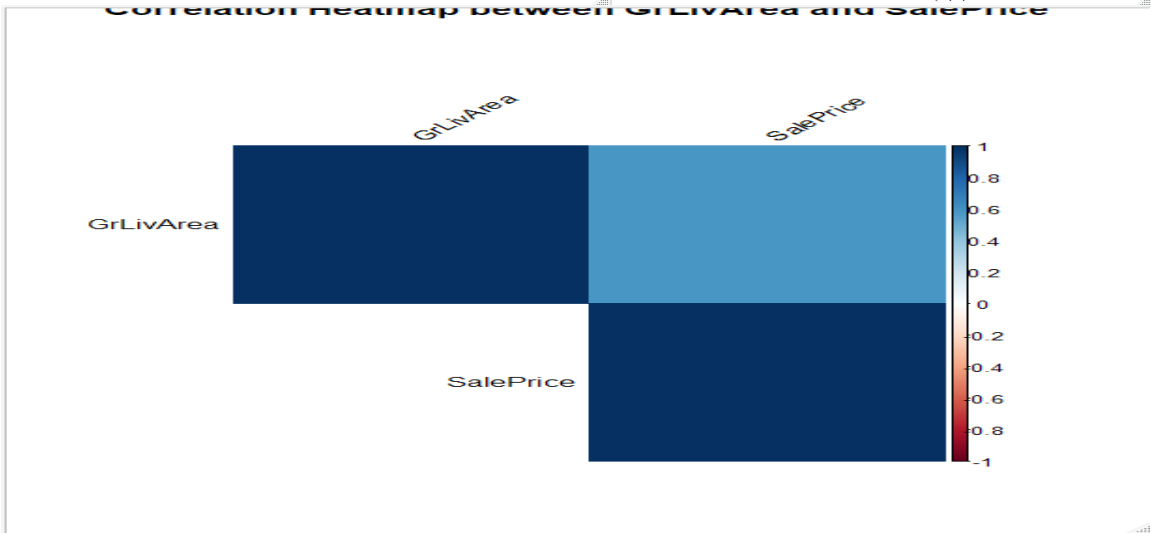
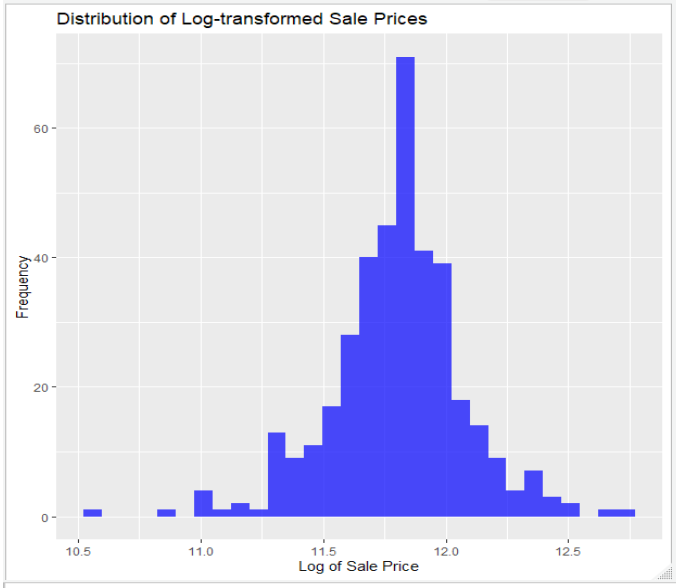
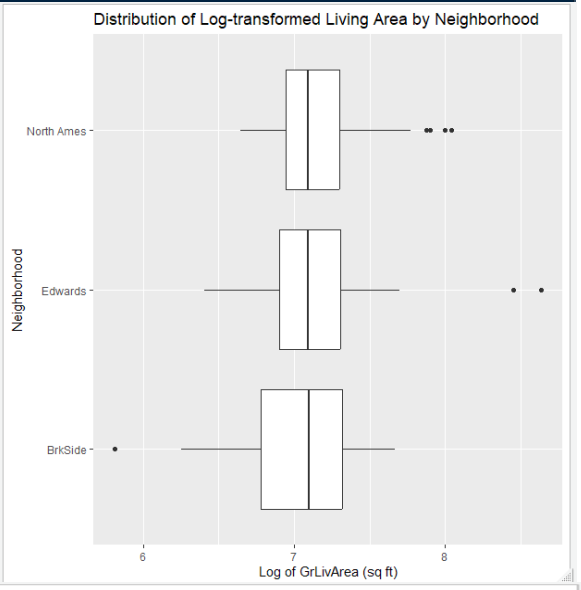


Figure A1 - Raw data graphed:

Figure A2 – Log Transformation:

```r
library(readxl)
library(ggplot2)
data <- read_excel("D:/SMU - Grad School/Spring 2024/DS 6371/DS 6371 - Spring 2024/MSDS_6371_Stat_Foundations/Project/Neighborhoods.xlsx")
ggplot(data, aes(x=log(SalePrice))) +
  geom_histogram(bins=30, fill="blue", alpha=0.7) +
  ggtitle("Distribution of Log-transformed Sale Prices") +
  xlab("Log of Sale Price") +
  ylab("Frequency")
ggplot(data, aes(x=Neighborhood, y=log(GrLivArea))) +
  geom_boxplot() +
  coord_flip() +
  ggtitle("Distribution of Log-transformed Living Area by Neighborhood") +
  xlab("Neighborhood") +
  ylab("Log of GrLivArea (sq ft)")
ggplot(data, aes(x=log(GrLivArea), y=log(SalePrice))) +
  geom_point(alpha=0.5, color="blue") +
  ggtitle("Scatter Plot: Log of GrLivArea vs. Log of SalePrice") +
  xlab("Log of GrLivArea (sq ft)") +
  ylab("Log of SalePrice") +
  theme_minimal()
cor_matrix <- cor(data[c("log(GrLivArea)", "log(SalePrice)")])
corrplot(cor_matrix, method="color", type="upper", tl.col="black", tl.srt=45,
         title="Correlation Heatmap between Log-transformed GrLivArea and SalePrice",
         cl.lim=c(-1, 1))
```





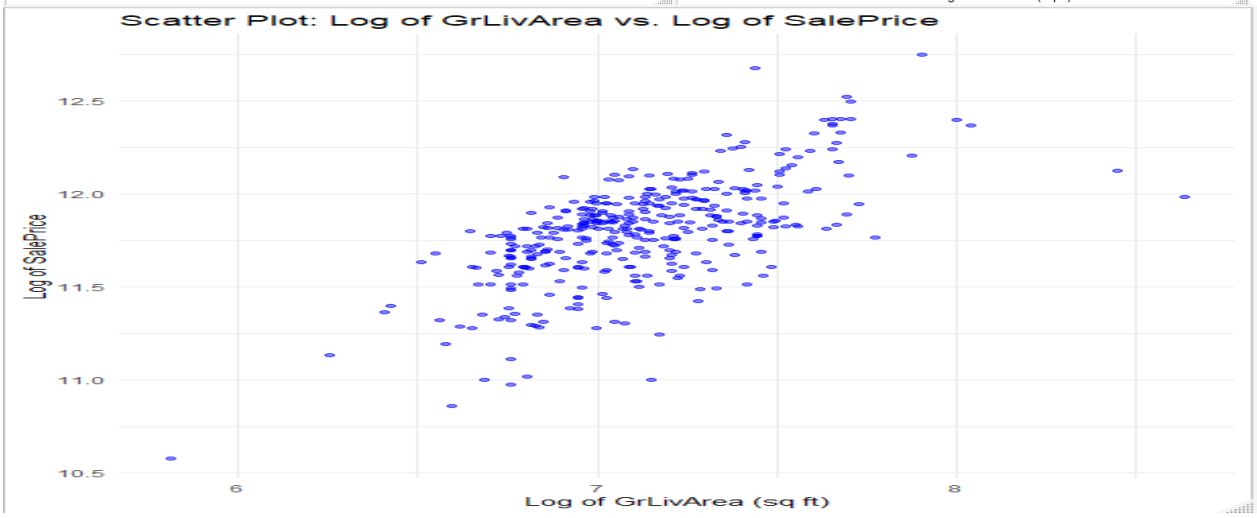
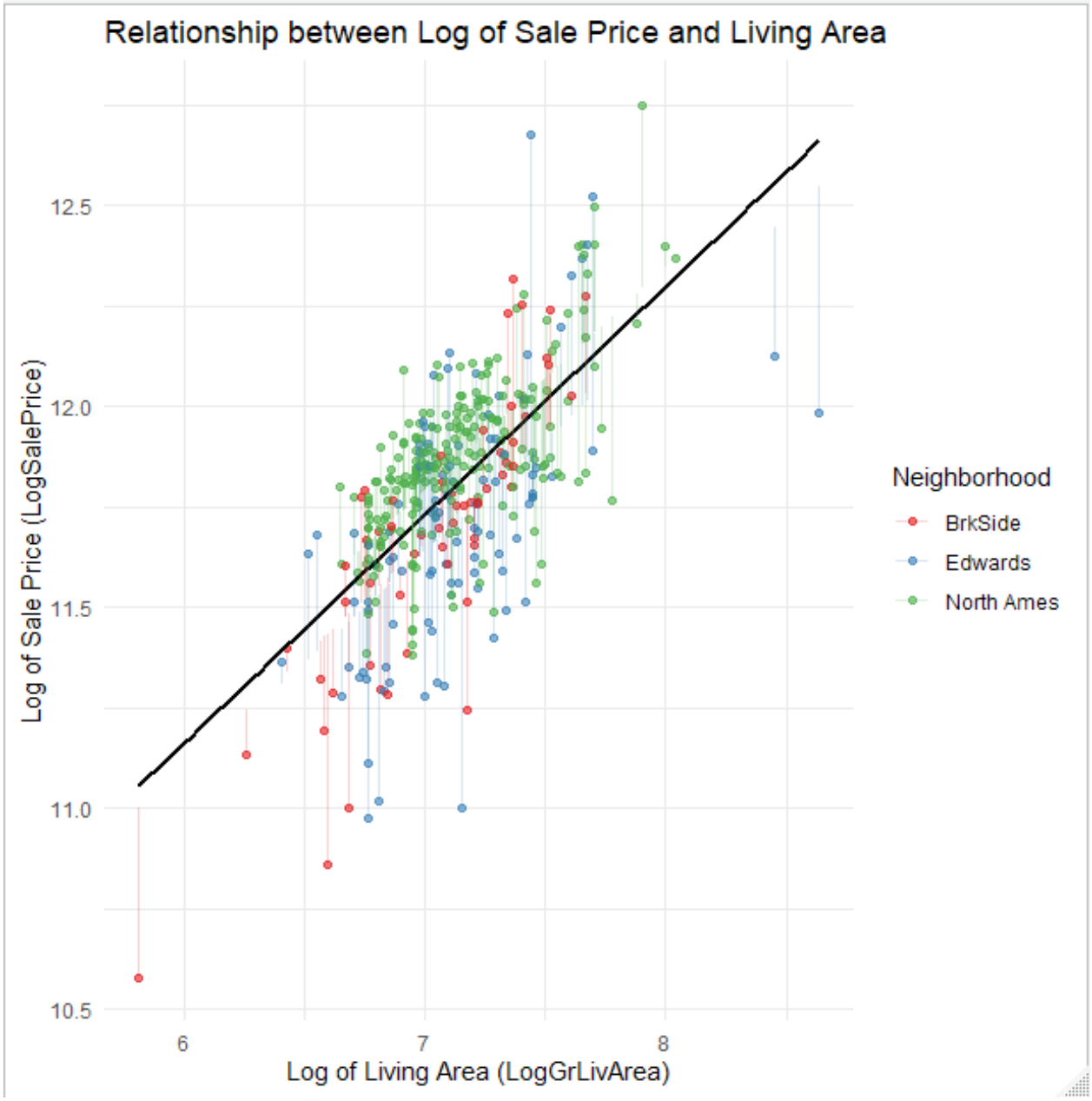


Figure A3 – Build and Fit the Model Relationship between LogSalePrice vs GrLivArea for the Neighborhoods:

```r
library(ggplot2)
library(readxl)
file_path <- "D:/SMU - Grad School/Spring 2024/DS 6371/DS 6371 - Spring 2024/MSDS_6371_Stat_Foundations/Project/Neighborhoods.xlsx"
data <- read_excel(file_path)
data$LogSalePrice <- log(data$SalePrice)
data$LogGrLivArea <- log(data$GrLivArea)
model_extended <- lm(LogSalePrice ~ LogGrLivArea + Neighborhood, data=data)
data$fitted_values <- fitted(model_extended)
data$residuals <- residuals(model_extended)
ggplot(data, aes(x=LogGrLivArea, y=LogSalePrice, color=Neighborhood)) +
  geom_point(alpha=0.6) +
  geom_smooth(method="lm", se=FALSE, color="black") +
  geom_segment(aes(xend=LogGrLivArea, yend=fitted_values), alpha=0.2) +
  labs(title="Relationship between Log of Sale Price and Living Area",
       x="Log of Living Area (LogGrLivArea)",
       y="Log of Sale Price (LogSalePrice)") +
  theme_minimal() +
  scale_color_brewer(palette="Set1")
```



```r
library(readxl)
library(car)

file_path <- "D:/SMU - Grad School/Spring 2024/DS 6371/DS 6371 - Spring 2024/MSDS_6371_Stat_Foundations/Project/Neighborhoods.xlsx"
data <- read_excel(file_path)
data$LogSalePrice <- log(data$SalePrice)
data$LogGrLivArea <- log(data$GrLivArea)
data <- cbind(data, model.matrix(~ Neighborhood - 1, data=data))
model_extended <- lm(LogSalePrice ~ LogGrLivArea + Neighborhood, data=data)
par(mfrow=c(2,2))
plot(model_extended, which=1)  # Residuals vs Fitted
plot(model_extended, which=2)  # Normal Q-Q
plot(model_extended, which=3)  # Scale-Location
plot(model_extended, which=5)  # Residuals vs Leverage
```

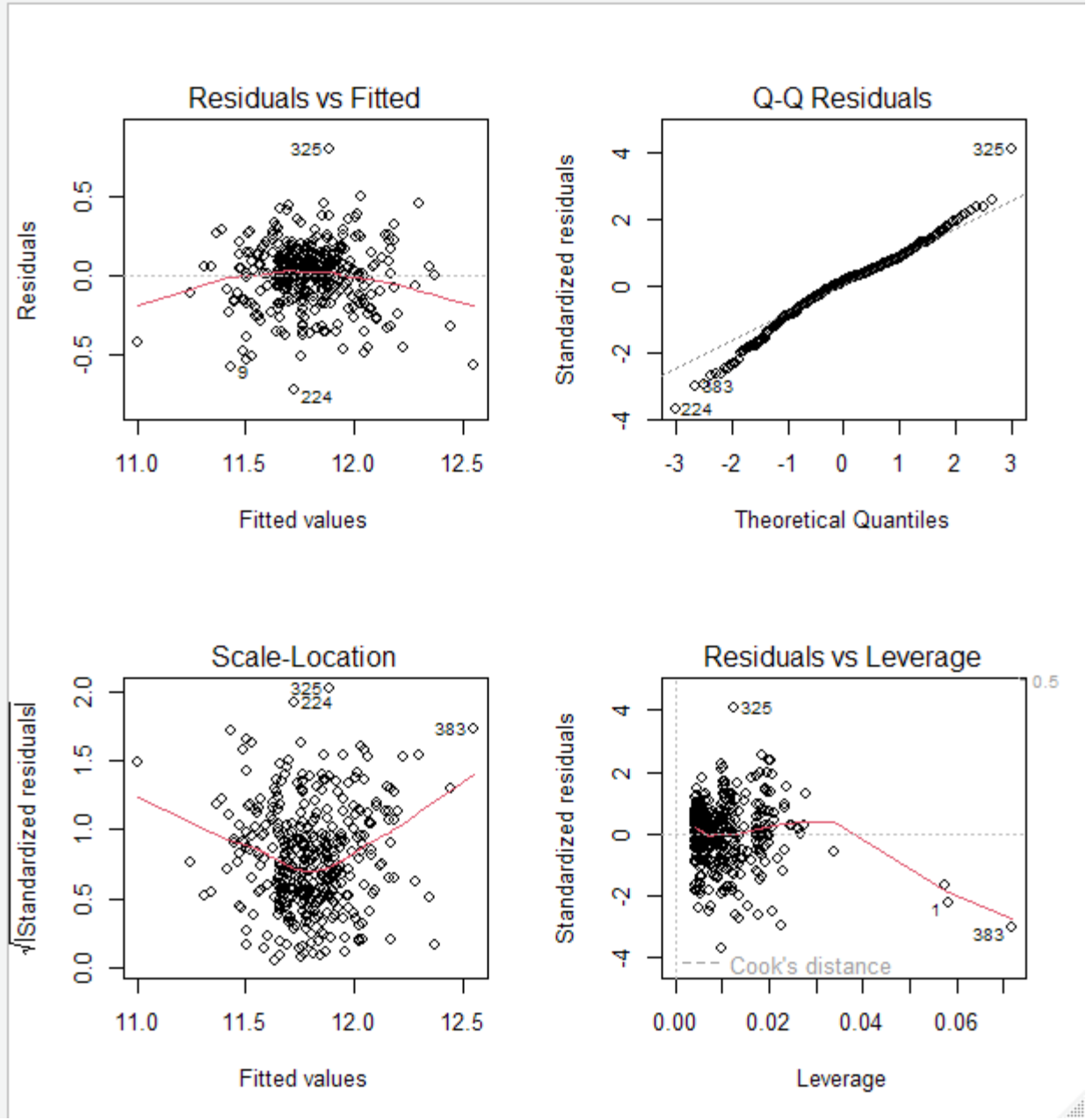
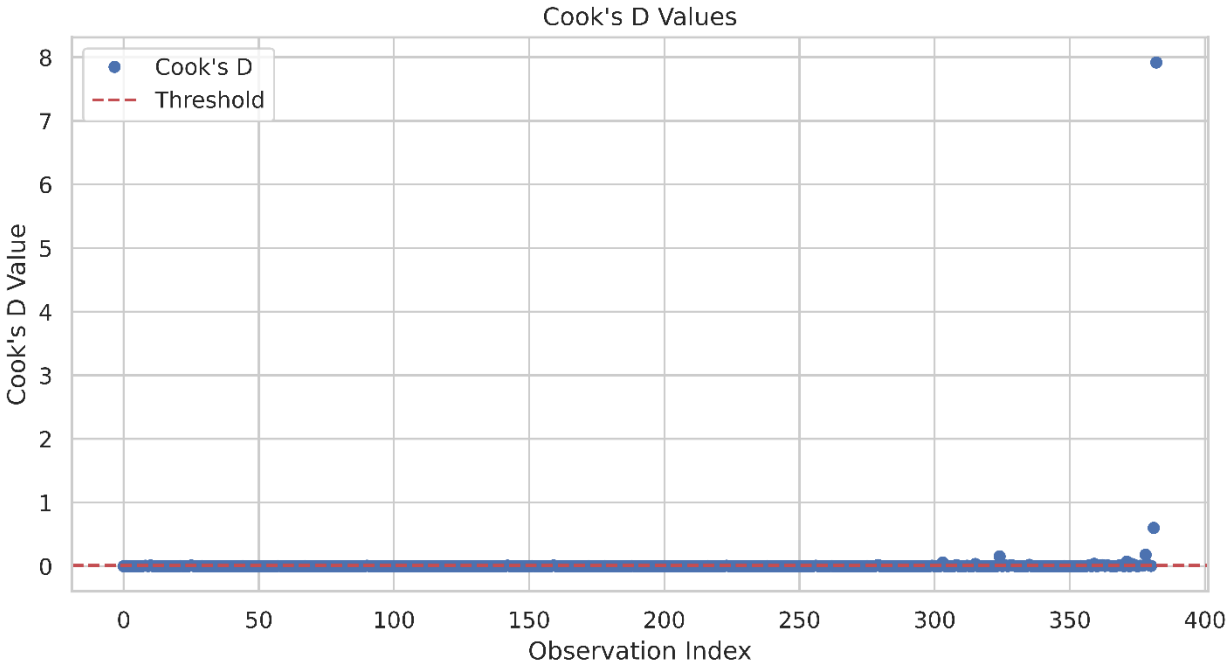
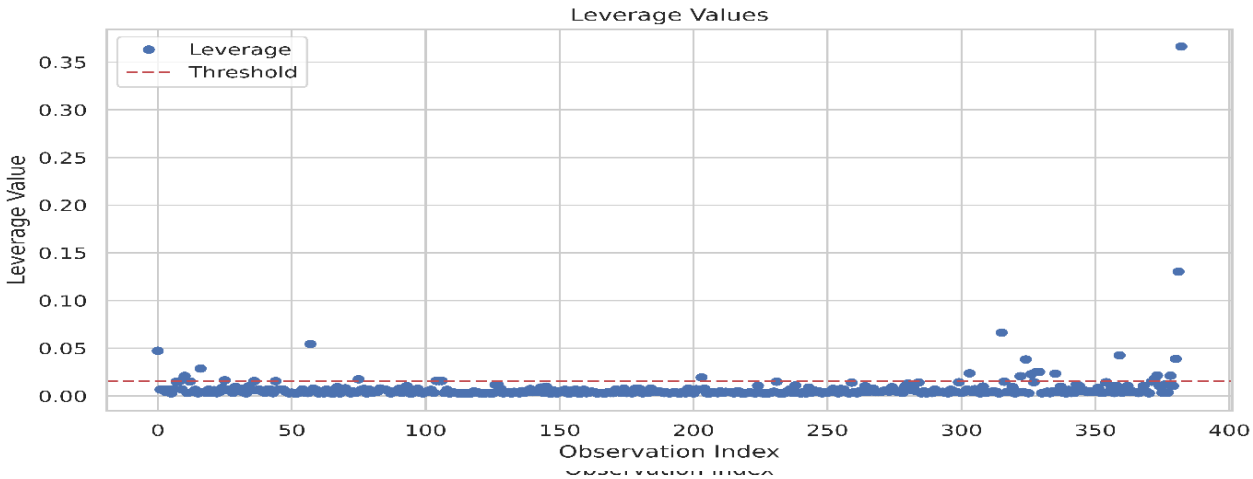


Figure A4 - Residuals vs Fitted, Normal Q-Q, Scale Location, Residuals vs Leverage:

Figure A5 – Cook's D Values, Leverage Values, and Influential Points:

```r
library(car)
library(readxl)
file_path <- "D:/SMU - Grad School/Spring 2024/DS 6371/DS 6371 - Spring 2024/MSDS_6371_Stat_Foundations/Project/Neighborhoods.xlsx"

data <- read_excel(file_path)
model <- lm(SalePrice ~ LotArea + OverallQual, data = data)
cooks_d <- cooks.distance(model)
leverage <- hatvalues(model)
n <- nrow(data)
k <- length(model$coefficients) - 1  # subtracting the intercept
high_cooks_d <- 4 / (n - k - 1)
high_leverage <- 2 * (k + 1) / n
influential_points <- which(cooks_d > high_cooks_d | leverage > high_leverage)
print("Cook's D Values:")
print(cooks_d)
print("Leverage Values:")
print(leverage)
print("Influential Points:")
print(influential_points)
> print(influential_points)
  1  11  17  26  58  76 105 106 107 160 204 280 304 309 316 323 325 327 329 330 336 358 360 362 363 365 370 372 373 374 375 379 380 381
  1  11  17  26  58  76 105 106 107 160 204 280 304 309 316 323 325 327 329 330 336 358 360 362 363 365 370 372 373 374 375 379 380 381
382 383
382 383
```





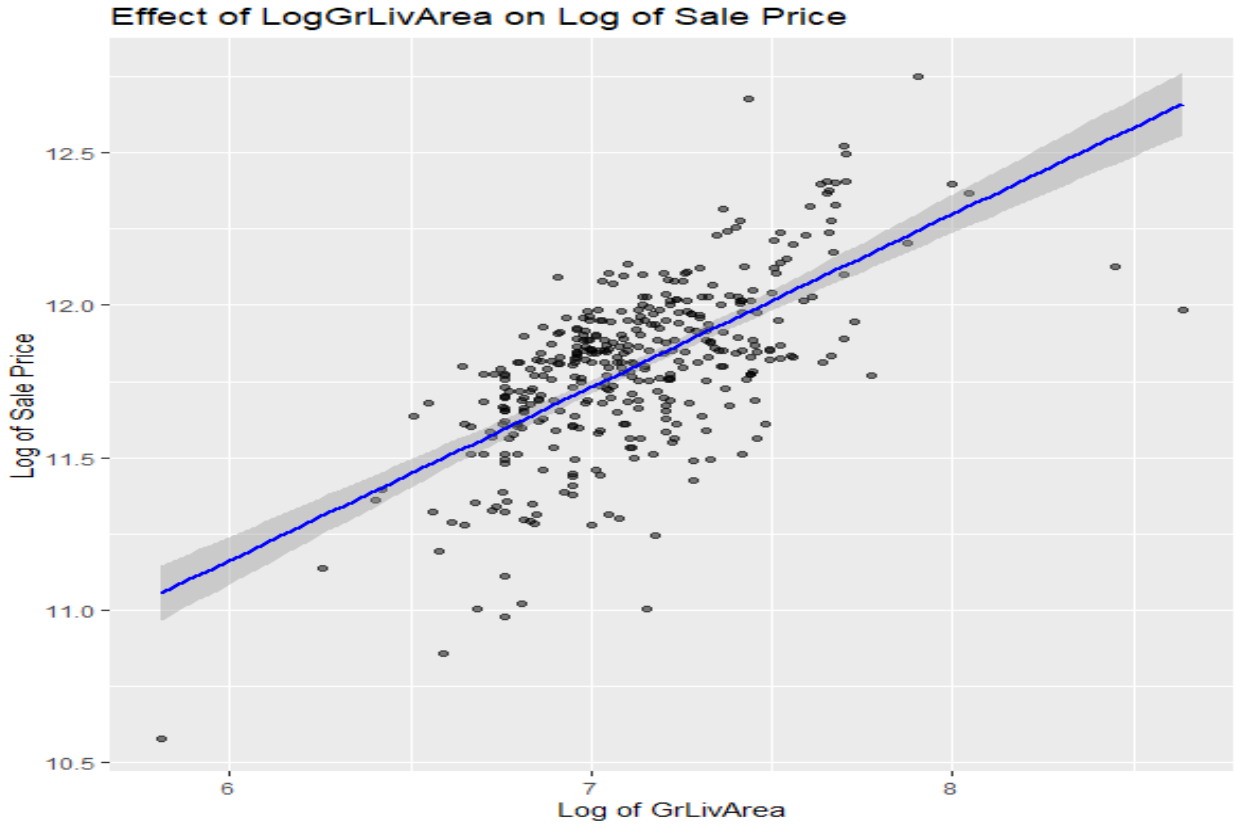
```r
library(readxl)
library(Metrics)
file_path <- "D:/SMU - Grad School/Spring 2024/DS 6371/DS 6371 - Spring 2024/MSDS_6371_Stat_Foundations/Project/Neighborhoods.xlsx"
data <- read_excel(file_path, sheet = 1)  # Adjust 'sheet' as necessary
dependent_var <- log(data$SalePrice)  # Using logarithm of the sale price
independent_vars <- data[, c('LotArea', 'OverallQual')]
model <- lm(dependent_var ~ LotArea + OverallQual, data = data)
adjusted_r_squared <- summary(model)$adj.r.squared
print(paste("Adjusted R-squared: ", adjusted_r_squared))
set.seed(123)  # for reproducibility
folds <- sample(1:10, size = nrow(data), replace = TRUE)
cv_errors <- numeric(length(unique(folds)))

for(f in unique(folds)) {
  test_indices <- which(folds == f)
  train_indices <- setdiff(1:nrow(data), test_indices)

  model_cv <- lm(dependent_var ~ LotArea + OverallQual, data = data, subset = train_indices)
  predicted_values <- predict(model_cv, newdata = data[test_indices, ])

  cv_errors[f] <- mean((data$SalePrice[test_indices] - exp(predicted_values))^2)  # Assuming SalePrice needs to be back-transformed from log
}
cv_press <- sum(cv_errors)
print(paste("Internal CV Press: ", cv_press))
```

- **Adjusted R-squared:** 0.4857
- **Internal CV Press:** 9246326552.898

Figure A6 – Adjusted R-squared & Internal CV Press:

Figure A7 – Parameters:

```
library(readxl)
library(ggplot2)
data <- read_excel("D:/SMU - Grad School/Spring 2024/DS 6371/DS 6371 - Spring 2024/MSDS_6371_Stat_Foundations/Project/Neighborhoods.xlsx")
data$LogGrLivArea <- log(data$GrLivArea)
model <- lm(log(SalePrice) ~ LogGrLivArea + Neighborhood, data = data)
summary(model)
conf_int <- confint(model)
print(conf_int)
ggplot(data, aes(x=LogGrLivArea, y=log(SalePrice))) +
  geom_point(alpha=0.5) +
  geom_smooth(method="lm", se=TRUE, color="blue") +
  labs(title="Effect of LogGrLivArea on Log of Sale Price",
       x="Log of GrLivArea",
       y="Log of Sale Price")
ggsave("LogGrLivArea_vs_SalePrice.png")

Call:
lm(formula = log(SalePrice) ~ LogGrLivArea + Neighborhood, data = data)

Residuals:
     Min       1Q   Median       3Q      Max
-0.72154 -0.10592  0.02469  0.11565  0.79364

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)            7.76936    0.22919  33.900  < 2e-16 ***
LogGrLivArea           0.55579    0.03237  17.171  < 2e-16 ***
NeighborhoodEdwards   -0.02044    0.03252  -0.629     0.53
NeighborhoodNorth Ames  0.13279    0.02906   4.569 6.63e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1961 on 379 degrees of freedom
Multiple R-squared:  0.4897,    Adjusted R-squared:  0.4857
F-statistic: 121.2 on 3 and 379 DF,  p-value: < 2.2e-16

>
> # Confidence Intervals
> conf_int <- confint(model)
> print(conf_int)
                          2.5 %     97.5 %
(Intercept)            7.31872287 8.21999895
LogGrLivArea           0.49214387 0.61943290
NeighborhoodEdwards   -0.08437241 0.04349721
NeighborhoodNorth Ames  0.07564743 0.18992983
```



```
Call:
lm(formula = SalePrice ~ LotArea, data = df_cut)

Residuals:
    Min      1Q  Median      3Q     Max
-283444  -48187  -17590   31209  553008

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.584e+05  2.923e+03   54.19   <2e-16 ***
LotArea     2.150e+00  2.029e-01   10.60   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 76600 on 1457 degrees of freedom
Multiple R-squared:  0.07153,   Adjusted R-squared:  0.07089
F-statistic: 112.2 on 1 and 1457 DF,  p-value: < 2.2e-16

> print(anova(slr))
Analysis of Variance Table

Response: SalePrice
            Df     Sum Sq    Mean Sq F value    Pr(>F)
LotArea      1 6.5861e+11 6.5861e+11  112.25 < 2.2e-16 ***
Residuals 1457 8.5489e+12 5.8674e+09
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

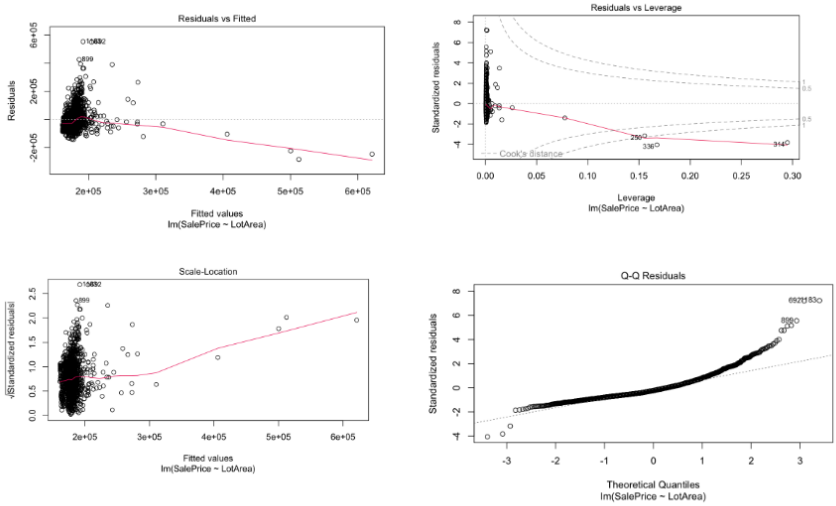


Figure B1 – Single Linear Regression: Sans High Leverage Outlier

```
Call:
lm(formula = SalePrice ~ GrLivArea + FullBath, data = df_cut)

Residuals:
    Min      1Q  Median      3Q     Max
-333450  -26705   -1521   22147  338795

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1108.421   4715.724  -0.235    0.814
GrLivArea      95.887      3.563  26.914  < 2e-16 ***
FullBath    23646.976   3326.323   7.109 1.82e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 53800 on 1456 degrees of freedom
Multiple R-squared:  0.5423,    Adjusted R-squared:  0.5417
F-statistic: 862.7 on 2 and 1456 DF,  p-value: < 2.2e-16

> print(anova(mlr1))
Analysis of Variance Table

Response: SalePrice
            Df     Sum Sq    Mean Sq  F value    Pr(>F)
GrLivArea    1 4.8473e+12 4.8473e+12 1674.845 < 2.2e-16 ***
FullBath     1 1.4627e+11 1.4627e+11   50.538 1.824e-12 ***
Residuals 1456 4.2139e+12 2.8942e+09
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

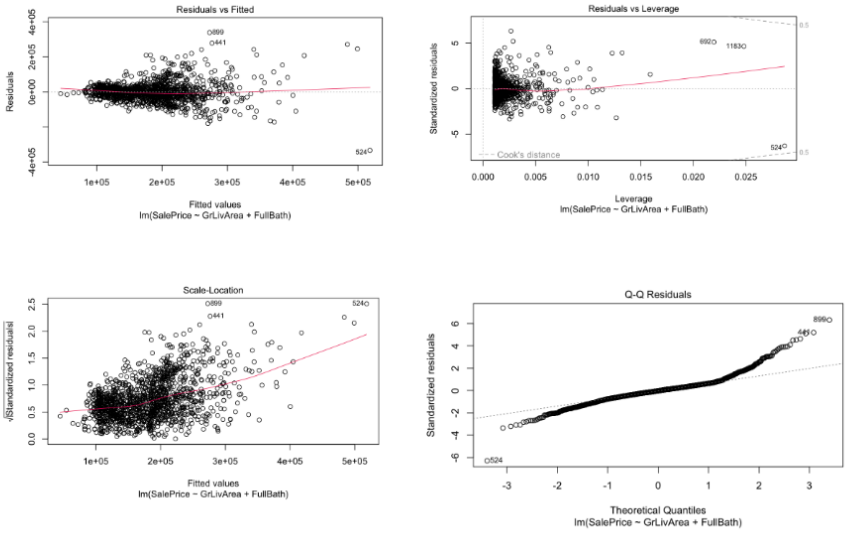


Figure B2 – Multiple Linear Regression 1: Sans Outlier

```
Call:
lm(formula = SalePrice ~ GrLivArea * interaction + LogOverallQual,
    data = df_3_cut)

Residuals:
    Min      1Q  Median      3Q     Max
-321905  -24223   -2678   18180  299040

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)           -1.109e+05  3.707e+04  -2.991 0.002830 **
GrLivArea             -6.343e+00  2.879e+01  -0.220 0.825687
interactionFV         -1.260e+05  4.567e+04  -2.759 0.005878 **
interactionRH         -3.723e+04  4.526e+04  -0.823 0.410880
interactionRL         -7.981e+04  3.709e+04  -2.152 0.031581 *
interactionRM         -5.061e+04  3.783e+04  -1.338 0.181195
LogOverallQual         1.459e+05  5.855e+03  24.923  < 2e-16 ***
GrLivArea:interactionFV 1.103e+02  3.317e+01   3.325 0.000906 ***
GrLivArea:interactionRH 3.315e+01  3.281e+01   1.010 0.312471
GrLivArea:interactionRL 8.362e+01  2.881e+01   2.903 0.003758 **
GrLivArea:interactionRM 3.934e+01  2.933e+01   1.342 0.179943
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 42570 on 1448 degrees of freedom
Multiple R-squared:  0.7151,    Adjusted R-squared:  0.7131
F-statistic: 363.4 on 10 and 1448 DF,  p-value: < 2.2e-16

> print(anova(mlr2))
Analysis of Variance Table

Response: SalePrice
                     Df     Sum Sq    Mean Sq  F value    Pr(>F)
GrLivArea             1 4.8473e+12 4.8473e+12 2675.269 < 2.2e-16 ***
interaction           4 4.2709e+11 1.0677e+11   58.929 < 2.2e-16 ***
LogOverallQual        1 1.1888e+12 1.1888e+12  656.113 < 2.2e-16 ***
GrLivArea:interaction 4 1.2067e+11 3.0168e+10   16.650 2.383e-13 ***
Residuals          1448 2.6236e+12 1.8119e+09
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

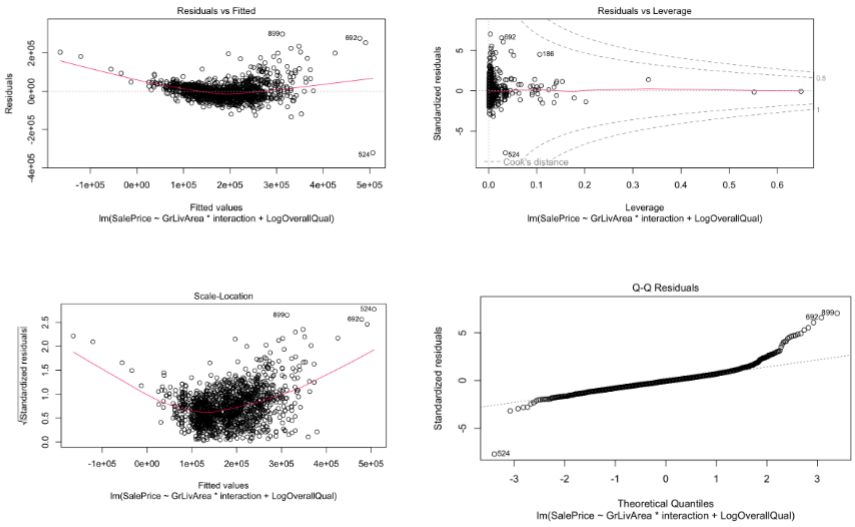


Figure B3 – Multiple Linear Regression 2: Sans Outlier

```
Call:
lm(formula = SalePrice ~ GrLivArea + LogOverallQual, data = df_3_cut)

Residuals:
    Min      1Q  Median      3Q     Max
-301741  -24132   -1709   19249  304992

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -2.062e+05  8.915e+03  -23.12   <2e-16 ***
GrLivArea     7.061e+01  2.759e+00   25.59   <2e-16 ***
LogOverallQual  1.574e+05  5.888e+03   26.74   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 44810 on 1456 degrees of freedom
Multiple R-squared:  0.6824,    Adjusted R-squared:  0.682
F-statistic:  1564 on 2 and 1456 DF,  p-value: < 2.2e-16

> print(anova(mlr3))
Analysis of Variance Table

Response: SalePrice
               Df     Sum Sq    Mean Sq F value    Pr(>F)
GrLivArea       1 4.8473e+12 4.8473e+12 2413.55 < 2.2e-16 ***
LogOverallQual  1 1.4360e+12 1.4360e+12  715.01 < 2.2e-16 ***
Residuals    1456 2.9242e+12 2.0084e+09
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

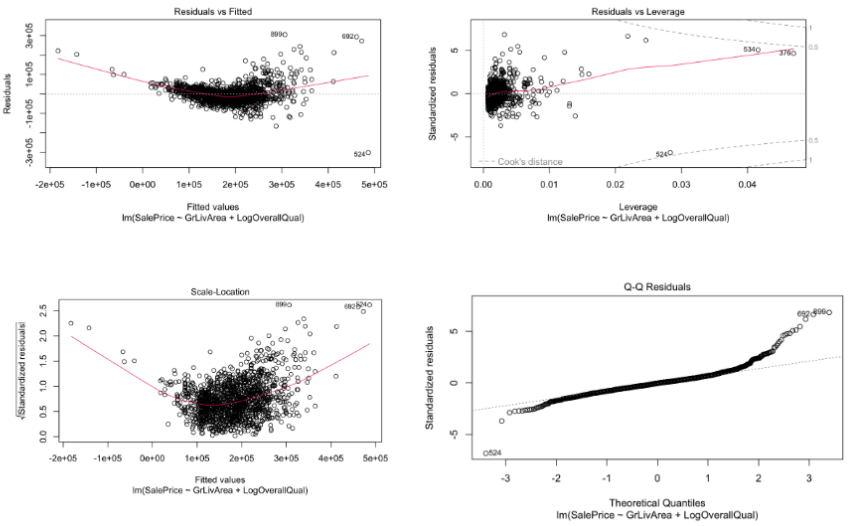


Figure B4 – Mutiple Linear Regression 3: Sans Outlier

CV Error SLR

```{r}
set.seed(444)
folds <- sample(1:10, size = nrow(df_clean), replace = TRUE)
cv_error_slr <- numeric(length(unique(folds)))

for(f in unique(folds)) {
  test_indices <- which(folds == f)
  train_indices <- setdiff(1:nrow(df_clean), test_indices)
  model_cv <- lm(SalePrice ~ LotArea, data = df_clean, subset = train_indices)
  predicted_values <- predict(model_cv, newdata = df_clean[test_indices, ])
  cv_error_slr[f] <- mean((predicted_values - df_clean$SalePrice[test_indices])^2)
}
```

CV Error MLR1

```{r}
set.seed(444)
folds <- sample(1:10, size = nrow(df_clean), replace = TRUE)
cv_error_mlr1 <- numeric(length(unique(folds)))

for(f in unique(folds)) {
  test_indices <- which(folds == f)
  train_indices <- setdiff(1:nrow(df_clean), test_indices)
  model_cv <- lm(SalePrice ~ GrLivArea + FullBath, data = df_clean, subset = train_indices)
  predicted_values <- predict(model_cv, newdata = df_clean[test_indices, ])
  cv_error_mlr1[f] <- mean((predicted_values - df_clean$SalePrice[test_indices])^2)
}
```

CV Error MLR2

```{r}
set.seed(444)
folds <- sample(1:10, size = nrow(df_3_cut), replace = TRUE)
cv_error_mlr2 <- numeric(length(unique(folds)))

for(f in unique(folds)) {
  test_indices <- which(folds == f)
  train_indices <- setdiff(1:nrow(df_3_cut), test_indices)
  model_cv <- lm(SalePrice ~ GrLivArea * interaction  +  LogOverallQual, data = df_3_cut, subset = train_indices)
  predicted_values <- predict(model_cv, newdata = df_3_cut[test_indices, ])
  cv_error_mlr2[f] <- mean((predicted_values - df_3_cut$SalePrice[test_indices])^2)
}
```

CV Error MLR3

```{r}
set.seed(444)
folds <- sample(1:10, size = nrow(df_3_cut), replace = TRUE)
cv_error_mlr3 <- numeric(length(unique(folds)))

for(f in unique(folds)) {
  test_indices <- which(folds == f)
  train_indices <- setdiff(1:nrow(df_3_cut), test_indices)
  model_cv <- lm(SalePrice ~ GrLivArea  +  LogOverallQual, data = df_3_cut, subset = train_indices)
  predicted_values <- predict(model_cv, newdata = df_3_cut[test_indices, ])
  cv_error_mlr3[f] <- mean((predicted_values - df_3_cut$SalePrice[test_indices])^2)
}
```

Fig B5 – CV Press for each model