Yves Hughes & Xavier Mojica

SMU | MSDS  DS6371 Spring 23

real estate in ames iowa

## Introduction

This case study will focus on the analysis of the price of homes, and on creating predictive models for the final sale price of homes in Ames, Iowa.

The first analysis question asks us to look into the relationship between the sales price of homes and the square footage of those homes per three neighborhoods of interest.

As for the second analysis question, we will use regression modeling and other techniques in order to select the best explanatory variables for the optimal model that will predict any home’s sale price.

## Data Description

The open-source data were provided by Kaggle’s site for the House Prices - Advanced Regression Techniques ***https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques/data****.* The data set for the training data is made up of 1460 observations, while the test data set is comprised of 1459 observations with 79 explanatory variables. Since the data are readily available on Kaggle’s website, one can find the data set description, training, and testing data sets under the tab labeled Data.

# Analysis Question 1

## Restatement of Problem

As part of our 1st analysis, we will be examining how the sale price of homes is related to the square footage of those homes with respect to 3 neighborhoods of interest. Those neighborhoods happen to be Edwards, North Ames (NAmes), and Brookside (BrkSide).

## Preparation

We start by reading in the dataset and looking at the structure of both the training and testing datasets. Then, we round the living area (GrLivArea) by increments of 100 sq. ft. and visualize the data to get an initial understanding of the relationship between SalePrice and RoundedLivArea. Also, we filter the data to include only the three neighborhoods of interest: NAmes, Edwards and BrkSide.

## Build and Fit the Model

We built a linear regression model using the original training dataset and checked for assumption violations. We found that the residuals were not normally distributed with an unequal variance so we performed a log transformation on both SalePrice and RoundedLivArea. See Fig.1.

After the transformation, the assumptions seem to be met. However, there were two outliers that needed to be removed to improve the model. See Fig. 2

## Transformation

By having performed a log transformation of the data from Fig.1, we will address some concerns of unequal variance that appear to be present in Fig.1. As we see from Fig.1, from each data set based on neighborhood, the overall spread for each neighborhood is shown to start off large and tapper off to smaller variances. These types of violations will be addressed through a log-log transformation. Upon having implemented the log transformation that we can see in Fig.2, we are able to perceive visually strong evidence of linearity and improved constant variance. In other words, we can see a relationship between the log of Sale Price and the log of Rounded Living Area with respect to the Neighborhoods (BkrSide, Edwards, NAmes) of interest.

## Nota Bene

It is critical to mention that the data pre- and post-log transformation do contain some observations of interest. Those observations—or houses of interest—appear to be outliers that have high leverage with a possibility of high or low influence. We will address those concerns later on in our study.

Furthermore, we would like to address that though the log transformation does appear to create a slight skewness in the data, the concern is appeased due to our sampling size (n) which is indeed > 30 and thus we would rely on Central Limit Theorem.

## Check Assumptions

In reviewing the residual plots for the log transformed data—see Fig.2—the relationship between the predictor variable and the target variable appears to be linear. Each observation appears to be independent. The variance of each observation is consistent across all the values of the predictor variables. The observations appear to be normally distributed.

Influential point analysis (Cook’s D and Leverage)

There are two outliers that are uniquely set apart from the rest of the data; their relationships between Sale Price and Living Area (pre- and post-log transformation) do not behave likewise to the rest of the observations shown within Fig.1 and Fig.2. Upon a closer examination of those two potential outliers, it was found that both of the outliers had a high living area in terms of sq.ft. (>3800 sq.ft. in the original scale or about 8.25 on the log scale), which wouldn’t truly represent the rest of the data.

## Comparing Competing Models

We will start by building various models using different combinations of predictors and interactions. Then, we shall calculate R2 and Adjusted R2 for each model, which provides insights into how well the model fit, while penalizing it for complexity. The higher the R2, the better the fit, taking into account the number of predictors in the model from the Adjusted R2. The R2 and adjustedR2 are higher in the Model with interactions which can be seen in Fig.3b and Fig.4.

To improve our model and avoid overfitting, we will utilize Internal Cross Validation PRESS. The Internal CV PRESS value was 14.22, indicating a reasonably good predictive performance when compared to alternative models. The value suggests that our model can effectively generalize to new observations while minimizing the risk of overfitting.

Referencing Fig.3b and Fig.4 , we will have to compare both models, the one without interactions and the one with interactions, to find out what the coefficients are and how significant each is for their respective model, with a significance level of 0.05.

There is sufficient evidence to suggest, with a significance level of 0.05, that the full model is significantly better than the reduced model (p-value = 0.0002749). We would have to reject our null hypothesis in favor of our alternative hypothesis, and therefore conduct our study using the full model.

## Assumptions

Linearity: Upon examining the scatter plot, we can see that there appears to exist a strong linear trend between Sale Price and Living Area. The means of those distributions appear to follow a linear pattern.

Normality: By examining the QQ Plot, we can see that overall—given that the sample size (n) is greater than 30—normal distribution albeit a slight skewness.

Equal Standard Deviation: It is apparent that there is significant evidence for homoscedasticity.

Independence: As far as the information provided, and the fact that there are house Identifications to go along with Sale Prices, independence will be assumed for each observation.

## Nota Bene

It is important to note that, from the scatter plot, the Residuals v. Fitted shows signs of a randomized cloud; this event favors the usage of linear modeling.

## Parameters

See Figures 3b – 6 for parameters

Interpretation

**Intercept Interpretation:**

We estimate that the median sale price for a house with a living area of 0 sq.ft. in the Brookside neighborhood to be associated with 27.454= $175.34 (p-value < 2e-16). We are 95% confident that the sales price is between 26.98787039 = $126.93 and 27.92093889 = $242.35.

**Slope Interpretation:**

We estimate that for the Edwards neighborhood, there is an 2-0.01601 = 0.99 times multiplicative decrease in the median sale price for a house (p-value = 0.616). This is with respect to the Brookside neighborhood. We are 95% confident that the multiplicative decrease is within the interval of 0.95 = 2-0.07865432 and 1.03 = 20.04663353 or an 5% decrease to 3% increase in comparison to Brookside.

We estimate that for the North Ames neighborhood, there is an 2e0.12543 = 1.09 times multiplicative increase in the median sale price for a house (p-value = 1.37e-05). This is with respect to the Brookside neighborhood. We are 95% confident that the multiplicative increase is within the interval of 1.05= 20.06946555 and 1.13 = 2 0.18139252 or between 5% and 13% increase in comparison to Brookside.

Lastly, for Brookside we estimate that there is an 20.60079 = 1.52 times multiplicative increase in the median sale price for a house (p-value < 2e-16) in Brookside. We are 95% confident that the true multiplicative increase is between 20.53483012 = 1.45 and 20.66675200 =1.60 or between 45% and 60% increase.

## Conclusion

In conclusion, our study found strong statistical evidence that the neighborhood where a home is located not only affects the sale price of the home, but also influences the relationship between living area and sale price.

# Analysis Question 2

## Restatement of Problem

For the second analysis, we aim to build the most predictive model for house sale prices in Ames, Iowa, considering all neighborhoods. The objective is to create four different models using various feature selection methods: forward selection, backward elimination, stepwise selection, and a custom model. The custom model can be any of the previous three models or a new one built by adding or subtracting variables as desired. The models will be evaluated using adjusted R2, Predicted Residual Sum of Squares, and Kaggle Score to determine which model performs best in predicting future sale prices of homes in Ames, Iowa.

## Preparation

Cleaning up the data is foremost and the most crucial step of our study for analysis two. Upon reviewing the data, the data underwent a thorough process:  
  
Data Preprocessing: We began by loading the required libraries and reading the training data. We then removed the 'Id' column, adjusted a column name, and converted columns with missing values to double type.

Handling Missing Values: We addressed missing values in both numeric and categorical variables. For numeric variables, we used mean imputation, while for categorical variables, we replaced missing values with "NA."

Creating Dummy Variables: To deal with categorical variables, we generated dummy variables for each category.

Correlation Analysis: We computed the correlation matrix for numeric variables and identified and eliminated pairs of variables that were highly correlated.

Exploring the Data: We examined the distribution of our target variable, SalePrice, using histograms and density plots. To normalize its distribution, we applied a log transformation to SalePrice.

Tackling Multicollinearity: We produced a user-friendly correlation table and removed additional highly correlated variables. We then calculated the Variance Inflation Factor (VIF) for predictors in the refined dataset. Finally, we eliminated predictors with high VIF values and recalculated VIF values for the updated dataset.

## Model Selection

**Backward Elimination**: The automation of the algorithm allows us to start with all candidate variables. It eliminates each variable one at a time until the AIC reaches its threshold where then it cannot remove another variable.

**Forward Selection:** This process involves starting with no variables in the model and then, through testing, adds another variable so long as it meets a criterion. This process is repeated until AIC cannot be lowered through the addition of a new variable.

**Stepwise**: In itself, this process is a combination of the two selection processes mentioned above. It initiates by adding the first most significant variable and then removes or ads one variable at a time till the AIC criterion is met.

**Custom**: After the results of the aforementioned models, the custom model uses variables that were selected based on significant levels and on industry knowledge. With those two criteria, we were able to craft our model.

## Type of Selection

The custom model is chosen as the model to move forward with in the study. This decision is made on the premise of domain knowledge along with statistically favorable variables.

## Checking Assumptions

We checked assumptions on the Residuals vs. Fitted, Normal Q-Q, Scale-Location, and Residuals vs. Leverage to check assumptions. (fig 2.5 – 2.10)

**Linearity**: Upon close examination of the residual plots, it is possible to see that the data are strongly penchant for a linear relationship. As there is a randomized cloud, the relationship between sale price in dollars and living area in terms of sq.ft. is reflected within the data, which can be seen in Fig A2.13.

**Normality**: This foremost assumption—quite pivotal, alongside linearity, is that upon examining the residuals, there are favorable indications for normality. The sample size is quite large, greater than 30, and we can say that the Central Limit Theorem would address any slight concern against normality. From the custom model's QQ plot, Fig A.12, we can see that the data are slightly skewed.

**Equal Standard Deviation:** Upon examination of the residual plots, there seems to be a penchant for homoscedasticity as seen per Fig A2.13.

**Independence:** As far as the information provided, and the fact that there are house Identifications to go along with Sale Prices, independence will be assumed for each observation.

## Comparing Competing Models

|  |  |  |  |
| --- | --- | --- | --- |
| **Predictive Models** | **Adjusted R2** | **CV PRESS** | **Kaggle Score** |
| Forward | 0.747 | 5.657 | TBD |
| Backward | 0.748 | 5.311 | TBD |
| Stepwise | 0.748 | 5.1 | TBD |
| CUSTOM | 0.447 | 7.5 | TBD |

## Conclusion

Despite the Custom Model's lower performance metrics compared to the Forward, Backward, and Stepwise models, it is important to consider domain knowledge. Yves is a licensed realtor, and allows us to identify variables that are not only statistically significant but also practically meaningful in predicting property prices. The Custom model has been carefully tailored and we can ensure that our predictive model reflects the realities of the industry today.

While the model's Adjusted R2, CV PRESS, and Kaggle Score may not be the highest among the competing model, it delivers a more comprehensive understanding of the factors that drive property prices. This additional context can lead to more accurate predictions in the long run.

Taking all of this into account, the Custom model stands out as the preferred choice for its ability to provide valuable insights and a deeper understanding of the real estate market.

# Appendix A: Analysis 1

|  |  |
| --- | --- |
|  | Fig. 1: Scatterplots depict original data without transformation of Sale Price vs. Rounded Living Area per Neighborhood |

|  |  |
| --- | --- |
|  | Fig. 2: Scatterplots depict log-log transformation of Sale Price vs. Rounded Living Area per Neighborhood |

|  |  |
| --- | --- |
|  | Fig.3a : PRESS Value |

|  |  |
| --- | --- |
|  |  |
| Fig.3b Model without Interactions | Fig.4 Model with Interactions |

|  |  |
| --- | --- |
|  | Fig.5 Full Model and Reduced Model |

|  |  |
| --- | --- |
|  | Fig.6: Lack of Fit Test (Full Model and Reduced Model) |

|  |  |
| --- | --- |
| Chart, scatter chart  Description automatically generated | Chart, scatter chart  Description automatically generated |
| Chart, line chart  Description automatically generated | Chart, scatter chart  Description automatically generated |
| Chart, scatter chart  Description automatically generated | Chart  Description automatically generated with low confidence |
| Fig.7: Residuals Analysis | |

|  |
| --- |
| Chart, scatter chart  Description automatically generated |
| Chart, scatter chart  Description automatically generated |
| Chart, scatter chart  Description automatically generated |
| Fig.8 Regression models with Intervals of Log Sale Prive v. Log Rounded Living Area by Neighborhood |

|  |  |
| --- | --- |
| Table  Description automatically generated | Fig. 9: This captures the estimates,  standard errors, t-values, and p-values  of our variables used in our model.  This also includes our Confidence Intervals. |
| Text  Description automatically generated |

# Appendix B: Analysis 2

|  |  |
| --- | --- |
| Chart, histogram  Description automatically generated | Chart  Description automatically generated |
| Fig A2.1 | Fig A2.2 |

|  |  |
| --- | --- |
| Chart, histogram  Description automatically generated | Chart, histogram  Description automatically generated |
| Fig A2.3 | Fig A2.4 |

|  |  |
| --- | --- |
| Graphical user interface, application  Description automatically generated | Graphical user interface, application, table  Description automatically generated |
| Fig A2.5 – Forward Model | Fig A2.6 – Forward Model |

|  |  |
| --- | --- |
| Chart  Description automatically generated | Chart  Description automatically generated |
| Fig A2.7 – Backward Model | Fig A2.8 – Backward Model |

|  |  |
| --- | --- |
| Graphical user interface  Description automatically generated | Graphical user interface, table  Description automatically generated with medium confidence |
| Fig A2.9 – Stepwise Model | Fig A2.10 – Stepwise Model |

|  |  |
| --- | --- |
| Chart, scatter chart  Description automatically generated | Chart, line chart  Description automatically generated |
| Fig A2.11 - Custom Model | Fig A2.12 - Custom Model |

|  |  |
| --- | --- |
| Chart, scatter chart  Description automatically generated | Chart, line chart  Description automatically generated |
| Fig A2.13 - Custom Model | Fig A2.14 – Custom Model |

|  |  |
| --- | --- |
| A picture containing text, receipt  Description automatically generated | A picture containing text, receipt  Description automatically generated |
| Forward Model Fig A2.14 | Backward Model Fig A2.15 |

|  |  |
| --- | --- |
| Table  Description automatically generated | Table  Description automatically generated |
| Stepwise Model Fig A2.16 | Custom Model Fig A2.17 |

---

title: "Final Project - Housing Price Analysis"

author: "Yves & Xavier"

date: "2023-04-08"

output: html\_document

---

```{r setup, include=FALSE}

#libraries

knitr::opts\_chunk$set(echo = TRUE)

library(tidyverse)

library(tidyr)

library(ggplot2)

library(MASS)

library(caret)

library(knitr)

library(rmarkdown)

options(rgl.useNULL=TRUE)

library(qpcR)

library(scales)

library(dplyr)

library(scales)

library(stringr)

library(olsrr)

```

```{r}

#Read in Data Sets, Training, Testing

training = read.csv("/Users/xaviermojica/Desktop/StatsIntro/house-prices-advanced-regression-techniques/train.csv", header = TRUE)

testing = read.csv("/Users/xaviermojica/Desktop/StatsIntro/house-prices-advanced-regression-techniques/test.csv", header = TRUE)

```

```{r}

#Columns Needed: SalePrice,GrLivArea, Neighborhood = ("NAmes, Edwards, BrkSide")

#Need to Address: Living area = GrLivArea must be in increaments of 100sq ft.

#Assumptions are met and outliers/influential obs have been id and addressed

#Build Model

#Provide Estimate and CI for any Estimate

```

```{r}

# Information about Dataset

str(training)

str(testing)

```

```{r}

#Realtors prefer to talk about living area in increments of 100 sq. ft.

#Rounding Living Area Square Footage

training = training %>% mutate(RoundedLivArea = round(GrLivArea, digits = -2))

testing = testing %>% mutate(RoundedLivArea = round(GrLivArea, digits = -2))

#Checking Training and Testing's Rounded Up data

training$RoundedLivArea

testing$RoundedLivArea

#Making Neighborhoods a factor

training$Neighborhood = as.factor(training$Neighborhood)

testing$Neighborhood = as.factor(testing$Neighborhood)

```

```{r}

# Visualizing Selected Data for Training Data

training %>% filter(Neighborhood %in% c("NAmes","BrkSide","Edwards")) %>% ggplot(aes(x = RoundedLivArea, y = SalePrice, col = Neighborhood)) + geom\_point() + geom\_jitter() + labs(title = "Rounded Living Area vs. Sale Price", x = "Rouned Living Area", y = "Sale Price")

```

```{r}

#Visualizing Neighborhoods Separately for Training Data

#Plotting Rounded Living Area vs. Sale Price by Neighborhood

training %>% filter(Neighborhood %in% c("NAmes","BrkSide","Edwards")) %>% ggplot(aes(x = RoundedLivArea, y = SalePrice, col = Neighborhood)) + geom\_point() + geom\_jitter() + facet\_wrap(~Neighborhood, scales = "free") +xlab("Rounded Living Area") + ylab("Sale Price in Dollars") + ggtitle("Rounded Living Area vs. Sale Price by Neighborhood")

```

```{r}

#Checking data for assumptions violations

training1 = training %>%

filter(Neighborhood %in% c("NAmes","Edwards","BrkSide")) %>%

dplyr::select(SalePrice,Neighborhood,RoundedLivArea)

#Exploring Observations. Validation of assumptions.

mod\_orginal\_data = lm(SalePrice~Neighborhood +RoundedLivArea, training1)

res = resid(mod\_orginal\_data)

#Producing Residuals vs. Fitted Plot

plot(fitted(mod\_orginal\_data), res)

#Adding horizontal line at 0

abline(0,0)

#Creating a QQ Plot for Residuals

qqnorm(res)

#Adding diagonal line

qqline(res)

#Histogram of Residuals with Curve

ggplot(training1, aes(x=mod\_orginal\_data$residuals)) + geom\_histogram(fill = "steelblue", bins = 39) + labs(title = "Histogram of Residuals", x = "Residuals", y = "Frequency")

```

```{r}

#Transformation due to Skewness

#Log Transformation

logSalePrice = log(training1$SalePrice)

logRoundedLivArea = log(training1$RoundedLivArea)

logTraining1 = data.frame(logSalePrice,logRoundedLivArea, training1$Neighborhood)

head(logTraining1) #checking log transformations

#Renaming Column training1$Neighborhood to Neighborhood

colnames(logTraining1)[3] = "Neighborhood"

head(logTraining1)

```

```{r}

#Plotting logSalePrice vs. logRoundedLivArea

logTraining1 %>% ggplot(aes(x = logRoundedLivArea, y = logSalePrice, col = Neighborhood)) + geom\_point() + geom\_jitter() + facet\_wrap(~Neighborhood, scales = "free") +xlab("Log Rounded Living Area") + ylab("Log Sale Price in Dollars") + ggtitle("Log Rounded Living Area vs. Log Sale Price by Neighborhood")

```

```{r}

#Checking data for assumptions violations for Log Data

#Exploring Observations. Validation of assumptions.

modLogData = lm(logSalePrice~Neighborhood +logRoundedLivArea, logTraining1)

res = resid(modLogData)

#Producing Residuals vs. Fitted Plot

plot(fitted(modLogData), res)

#Adding horizontal line at 0

abline(0,0)

#Creating a QQ Plot for Residuals

qqnorm(res)

#Adding diagonal line

qqline(res)

#Histogram of Residuals with Curve

ggplot(logTraining1, aes(x=modLogData$residuals)) + geom\_histogram(fill = "steelblue", bins = 39) + labs(title = "Histogram of Residuals", x = "Residuals", y = "Frequency")

#Studentized Residual

ols\_plot\_resid\_stud\_fit(modLogData)

#Cook's D

ols\_plot\_cooksd\_chart(modLogData)

```

```{r}

#Removing Outliers. Looking at both the Original and Log data set

#The two same outliers are present. We will be restricting the domain

#of the Log data set D:[0,8.25]

x = data.frame(logTraining1$logRoundedLivArea < 8.25)

x # using boolean to see which rows fit criterion

logTraining1\_noOutliers = logTraining1[-c(339,131),]

logTraining1\_noOutliers

```

```{r}

#Plotting logSalePrice vs. logRoundedLivArea without 2 Outliers

logTraining1\_noOutliers %>% ggplot(aes(x = logRoundedLivArea, y = logSalePrice, col = Neighborhood)) + geom\_point() + geom\_jitter() + facet\_wrap(~Neighborhood, scales = "free") +xlab("Log Rounded Living Area") + ylab("Log Sale Price") + ggtitle("Log Rounded Living Area vs. Log Sale Price by Neighborhood")

```

```{r}

#Checking data for assumptions violations for Log Data without 2 Outliers. Restricted Domain less than 8.25 logRoundedLivArea

#Exploring Observations. Validation of assumptions.

modLogDataNoOut = lm(logSalePrice~Neighborhood +logRoundedLivArea, logTraining1\_noOutliers)

res = resid(modLogDataNoOut)

#Producing Residuals vs. Fitted Plot

plot(fitted(modLogDataNoOut), res)

#Adding horizontal line at 0

abline(0,0)

#Creating a QQ Plot for Residuals

qqnorm(res)

#Adding diagonal line

qqline(res)

#Histogram of Residuals with Curve

ggplot(logTraining1\_noOutliers, aes(x=modLogDataNoOut$residuals)) + geom\_histogram(fill = "steelblue", bins = 39) + labs(title = "Histogram of Residuals", x = "Residuals", y = "Frequency")

#Studentized Residual

ols\_plot\_resid\_stud\_fit(modLogDataNoOut)

#Cook's D

ols\_plot\_cooksd\_chart(modLogDataNoOut)

```

```{r}

#Building Model without Interactions

summary(modLogDataNoOut)

plot(modLogDataNoOut)

pressmodLogDataNoOut = PRESS(modLogDataNoOut)

pressmodLogDataNoOut$stat

```

```{r}

#Plotting Model

########################Need work help on this

cf = data.frame(predict(modLogDataNoOut,logTraining1\_noOutliers, interval = "confidence"))

p = data.frame(predict(modLogDataNoOut,logTraining1\_noOutliers, interval = "prediction"))

both = cf %>% bind\_cols(p[,2:3], modLogDataNoOut)

both %>% ggplot(aes(x =logRoundedLivArea,y = logSalePrice)) + geom\_point(aes(color=Neighborhood)) +

geom\_line(aes(y=(fit),color=Neighborhood)) +

geom\_ribbon(aes(ymin=(lwr),ymax=(upr),fill=Neighborhood),alpha=.3) +

geom\_line(aes(y=(lwr),col=Neighborhood)) +

geom\_line(aes(y=(upr),col=Neighborhood)) +

facet\_wrap(~Neighborhood) +

labs(title = "Log Living Area vs Log Sale Price",

x = "Log of Living Area",

y = "Log of Sale Price")+ scale\_y\_continuous(labels=dollar)

logTraining1\_noOutliers %>% ggplot(aes(x = logRoundedLivArea, y = logSalePrice, col = Neighborhood)) + geom\_point() + geom\_jitter() + facet\_wrap(~Neighborhood, scales = "free") +xlab("Log Rounded Living Area") + ylab("Log Sale Price") + ggtitle("Log Rounded Living Area vs. Log Sale Price by Neighborhood")

```

```{r}

#Model with Interactions

fit = lm(logSalePrice~Neighborhood\*logRoundedLivArea, logTraining1\_noOutliers)

summary(fit)

confint(fit)

plot(fit)

pressfit = PRESS(fit)

pressfit$stat

```

```{r}

#Plotting Interactions

```

```{r}

#Lack of Fit Test

##Full and Reduced Model

anova(fit,modLogDataNoOut)

```

```{r}

#Testing our predictions using the full model

#In order to restrict domain and remove two outliers, we filtered out GrLivArea using exp(8.25) which gives us 4000

#making testing logged

logSalePriceTest = log(testing$SalePrice)

logRoundedLivAreaTest = log(testing$RoundedLivArea)

logGrLivArea = log(testing$GrLivArea)

logTesting1 = data.frame(testing$Id,testing$Neighborhood,logRoundedLivAreaTest, logGrLivArea)

testing1 = logTesting1 %>%

filter(Neighborhood %in% c("NAmes","Edwards","BrkSide"), logGrLivArea< 8.25) %>%

dplyr::select(Id,Neighborhood, GrLivArea,RoundedLivArea)

```

---

title: "Final Project - Housing Price Analysis"

author: "Yves & Xavier "

date: "2023-04-08"

output: html\_document

---

```{r setup, include=FALSE}

knitr::opts\_chunk$set(echo = TRUE)

```

```{r}

# Load necessary libraries

library(tidyverse)

library(caret)

library(olsrr)

library(corrplot)

library(dplyr)

library(glmnet)

# Read in train data

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

# Remove 'Id' column

train\_data <- train\_data[, -which(names(train\_data) == "Id")]

# Rename the column if it contains an unintentional space

colnames(train\_data)[colnames(train\_data) == "Exterior2ndWd Sdng"] <- "Exterior2ndWdSdng"

# Convert columns with missing values to double type

train\_data$LotFrontage <- as.double(train\_data$LotFrontage)

train\_data$MasVnrArea <- as.double(train\_data$MasVnrArea)

train\_data$GarageYrBlt <- as.double(train\_data$GarageYrBlt)

# Impute missing values for numeric variables (LotFrontage,MasVnrArea,GarageYrBlt)

train\_data$LotFrontage <- replace\_na(train\_data$LotFrontage, mean(train\_data$LotFrontage, na.rm = TRUE))

train\_data$MasVnrArea <- replace\_na(train\_data$MasVnrArea, mean(train\_data$MasVnrArea, na.rm = TRUE))

train\_data$GarageYrBlt <- replace\_na(train\_data$GarageYrBlt, mean(train\_data$GarageYrBlt, na.rm = TRUE))

# Impute missing values for categorical variables

train\_data$Electrical <- replace\_na(train\_data$Electrical, "NA")

train\_data$Exterior1st <- replace\_na(train\_data$Exterior1st, "NA")

train\_data$Exterior2nd <- replace\_na(train\_data$Exterior2nd, "NA")

train\_data$KitchenQual <- replace\_na(train\_data$KitchenQual, "NA")

train\_data$SaleType <- replace\_na(train\_data$SaleType, "NA")

train\_data$Functional <- replace\_na(train\_data$Functional, "NA")

train\_data$Utilities <- replace\_na(train\_data$Utilities, "NA")

train\_data$MSZoning <- replace\_na(train\_data$MSZoning, "NA")

train\_data$MasVnrType <- replace\_na(train\_data$MasVnrType, "NA")

train\_data$BsmtFinType1 <- replace\_na(train\_data$BsmtFinType1, "NA")

train\_data$BsmtFinType2 <- replace\_na(train\_data$BsmtFinType2, "NA")

train\_data$BsmtQual <- replace\_na(train\_data$BsmtQual, "NA")

train\_data$BsmtCond <- replace\_na(train\_data$BsmtCond, "NA")

train\_data$BsmtExposure <- replace\_na(train\_data$BsmtExposure, "NA")

train\_data$GarageType <- replace\_na(train\_data$GarageType, "NA")

train\_data$GarageCond <- replace\_na(train\_data$GarageCond, "NA")

train\_data$GarageFinish <- replace\_na(train\_data$GarageFinish, "NA")

train\_data$GarageQual <- replace\_na(train\_data$GarageQual, "NA")

train\_data$FireplaceQu <- replace\_na(train\_data$FireplaceQu, "NA")

train\_data$Fence <- replace\_na(train\_data$Fence, "NA")

train\_data$Alley <- replace\_na(train\_data$Alley, "NA")

train\_data$MiscFeature <- replace\_na(train\_data$MiscFeature, "NA")

train\_data$PoolQC <- replace\_na(train\_data$PoolQC, "NA")

# Create dummy variables for categorical variables

categorical\_vars <- train\_data %>% select(where(is.character))

train\_data\_dummies <- dummyVars(~ ., data = categorical\_vars, fullRank = T)

train\_data\_transformed <- predict(train\_data\_dummies, newdata = categorical\_vars)

numeric\_vars <- train\_data %>% select(where(is.numeric))

combined\_data <- bind\_cols(as.data.frame(numeric\_vars), as.data.frame(train\_data\_transformed))

print(combined\_data)

# Calculate the correlation matrix of numeric variables

numeric\_vars <- train\_data %>% select(where(is.numeric))

cor\_matrix <- cor(numeric\_vars, use = "complete.obs")

# Print correlation matrix

print(cor\_matrix)

# Identify highly correlated variable pairs (e.g., correlation greater than 0.8)

highly\_correlated <- findCorrelation(cor\_matrix, cutoff = 0.8)

highly\_correlated\_vars <- colnames(numeric\_vars)[highly\_correlated]

# Print highly correlated variable names

print(highly\_correlated\_vars)

# Remove highly correlated variables if necessary

numeric\_vars\_cleaned <- numeric\_vars %>% select(-one\_of(highly\_correlated\_vars))

# Combine cleaned numeric variables with the transformed categorical variables

combined\_data\_cleaned <- bind\_cols(as.data.frame(numeric\_vars\_cleaned), as.data.frame(train\_data\_transformed))

# Print the cleaned combined dataset

print(combined\_data\_cleaned)

```

```{r}

# Histogram of SalePrice

ggplot(train\_data, aes(x = SalePrice)) +

geom\_histogram(binwidth = 25000, fill = "steelblue", color = "black", alpha = 0.8) +

labs(title = "Histogram of SalePrice", x = "Sale Price", y = "Frequency") +

theme\_minimal()

# Density plot of SalePrice

ggplot(train\_data, aes(x = SalePrice)) +

geom\_density(fill = "steelblue", alpha = 0.8) +

labs(title = "Density Plot of SalePrice", x = "Sale Price", y = "Density") +

theme\_minimal()

```

```{r}

# Apply log transformation to SalePrice

train\_data$LogSalePrice <- log(train\_data$SalePrice)

# Histogram of Log-transformed SalePrice

ggplot(train\_data, aes(x = LogSalePrice)) +

geom\_histogram(binwidth = 0.1, fill = "steelblue", color = "black", alpha = 0.8) +

labs(title = "Histogram of Log-transformed SalePrice", x = "Log Sale Price", y = "Frequency") +

theme\_minimal()

# Density plot of Log-transformed SalePrice

ggplot(train\_data, aes(x = LogSalePrice)) +

geom\_density(fill = "steelblue", alpha = 0.8) +

labs(title = "Density Plot of Log-transformed SalePrice", x = "Log Sale Price", y = "Density") +

theme\_minimal()

```

```{r}

# Fit an initial model including all predictors

initial\_model <- lm(SalePrice ~ ., data = combined\_data\_cleaned)

# Check for aliased coefficients

aliased\_initial <- alias(initial\_model)

print(aliased\_initial)

# Identify aliased coefficients

aliased\_initial <- alias(initial\_model)

# Print aliased coefficients

print(aliased\_initial)

# Remove one variable from each pair of aliased coefficients

vars\_to\_remove <- rownames(aliased\_initial$Complete)[1:length(aliased\_initial$Complete)]

combined\_data\_cleaned <- combined\_data\_cleaned %>% select(-one\_of(vars\_to\_remove))

# Print the updated dataset

print(combined\_data\_cleaned)

```

```{r}

# Fit an initial model including all predictors

initial\_model <- lm(LogSalePrice ~ ., data = train\_data)

```

```{r}

### Address Multicollinearity

# Load the reshape2 package

library(reshape2)

# Generate a readable correlation table

cor\_matrix\_readable <- cor\_matrix

rownames(cor\_matrix\_readable) <- colnames(cor\_matrix)

cor\_matrix\_readable[upper.tri(cor\_matrix\_readable)] <- NA

cor\_matrix\_long <- as.data.frame(melt(cor\_matrix\_readable, na.rm = TRUE))

colnames(cor\_matrix\_long) <- c("var1", "var2", "correlation")

cor\_matrix\_long\_sorted <- cor\_matrix\_long[order(-abs(cor\_matrix\_long$correlation)),]

# Print the sorted correlation table

print(cor\_matrix\_long\_sorted)

# Remove even more highly correlated variables

selected\_vars <- c("GarageArea", "TotRmsAbvGrd", "X1stFlrSF", "OverallQual", "GarageYrBlt", "GrLivArea", "X2ndFlrSF", "BedroomAbvGr", "BsmtFinSF1")

numeric\_vars\_cleaned <- numeric\_vars %>% select(-one\_of(selected\_vars))

# Combine cleaned numeric variables with the transformed categorical variables

combined\_data\_cleaned <- bind\_cols(as.data.frame(numeric\_vars\_cleaned), as.data.frame(train\_data\_transformed))

# Print the cleaned combined dataset

print(combined\_data\_cleaned)

# Calculate the VIF for predictors in the cleaned dataset

# Load the required libraries

library(car)

# Define a function to remove aliased coefficients iteratively

remove\_aliased <- function(data) {

model <- lm(SalePrice ~ ., data = data)

aliased <- alias(model)

aliased\_vars <- row.names(aliased$Complete)

if (length(aliased\_vars) > 0) {

data <- dplyr::select(data, -dplyr::one\_of(aliased\_vars[1]))

return(remove\_aliased(data))

} else {

return(data)

}

}

# Remove aliased coefficients

combined\_data\_cleaned <- remove\_aliased(combined\_data\_cleaned)

# Calculate VIF for predictors in the cleaned dataset

vif\_values <- vif(lm(SalePrice ~ ., data = combined\_data\_cleaned))

print(vif\_values)

# Set a VIF threshold (5 or 10, for example)

vif\_threshold <- 5

# Remove predictors with VIF values greater than the threshold

high\_vif\_vars <- names(vif\_values)[vif\_values > vif\_threshold]

combined\_data\_cleaned <- dplyr::select(combined\_data\_cleaned, -dplyr::one\_of(high\_vif\_vars))

# Remove variables with perfect multicollinearity

combined\_data\_cleaned <- combined\_data\_cleaned %>%

select(-one\_of(c("BsmtCondNA", "BsmtFinType1NA", "Exterior2ndCBlock", "BsmtQualNA")))

# Recalculate VIF values for the updated dataset

vif\_values <- vif(lm(SalePrice ~ ., data = combined\_data\_cleaned))

print(vif\_values)

```

```{r, warning=FALSE}

### Model Selection

# Fit an initial model including all predictors after removing aliased coefficients

initial\_model <- lm(SalePrice ~ ., data = combined\_data\_cleaned)

```

```{r}

# Forward model selection

forward\_model <- ols\_step\_forward\_p(initial\_model)

summary(forward\_model)

plot(forward\_model)

```

```{r}

# Backward model selection

backward\_model <- ols\_step\_backward\_p(initial\_model)

summary(backward\_model)

plot(backward\_model)

```

```{r}

# Stepwise model selection

stepwise\_model <- ols\_step\_both\_p(initial\_model)

summary(stepwise\_model)

plot(stepwise\_model)

# Using cat function

cat("Adjusted R-squared (Stepwise Model):", adj\_r\_squared\_stepwise, "\n")

# Using print function

print(paste("Adjusted R-squared (Stepwise Model):", adj\_r\_squared\_stepwise))

```

```{r}

# Custom Model

custom\_model = lm(SalePrice ~ OverallCond + GarageCars + ScreenPorch + NeighborhoodBlueste + NeighborhoodBrDale + NeighborhoodClearCr + NeighborhoodMeadowV + NeighborhoodNPkVill + NeighborhoodStoneBr + NeighborhoodSWISU + NeighborhoodTimber + NeighborhoodVeenker, data = combined\_data\_cleaned)

summary(custom\_model)

plot(custom\_model)

```

```{r}

# Convert your data frame to a matrix

X <- model.matrix(forward\_formula, data = combined\_data\_cleaned)

y <- combined\_data\_cleaned$SalePrice

# Perform k-fold cross-validation using glmnet

k <- 10

cv\_results <- cv.glmnet(X, y, nfolds = k, alpha = 0)

# Calculate the mean squared error

cv\_mean\_squared\_error <- mean(cv\_results$cvm)

# Calculate CV PRESS

cv\_press\_forward <- cv\_mean\_squared\_error \* nrow(combined\_data\_cleaned)

# Create a table with the forward model's Adjusted R2 and CV PRESS

forward\_table <- data.frame(

Predictive.Models = "Forward",

Adjusted.R2 = summary(forward\_selected\_model)$adj.r.squared,

CV.PRESS = cv\_press\_forward

)

print(forward\_table)

```

```{r}

# Backward

# Fit the full model

full\_model <- lm(SalePrice ~ ., data = combined\_data\_cleaned)

# Perform backward stepwise regression with p-value thresholding

backward\_p\_model <- ols\_step\_backward\_p(full\_model, prem = 0.1)

# Get the variable names of the selected predictors

selected\_predictors <- names(coef(backward\_p\_model)[-1])

# Check if selected predictors is empty or contains invalid variable names

if (length(selected\_predictors) == 0) {

stop("No variables selected for the model.")

}

if (any(!selected\_predictors %in% colnames(combined\_data\_cleaned))) {

stop("Invalid variable names in the selected predictors.")

}

# Create the formula object for the selected predictors

formula <- as.formula(paste("SalePrice ~", paste(selected\_predictors, collapse = "+")))

# Print the formula object

cat("Formula: ", deparse(formula), "\n")

# Convert your data frame to a matrix

X <- model.matrix(formula, data = combined\_data\_cleaned)

y <- combined\_data\_cleaned$SalePrice

# Perform k-fold cross-validation using glmnet

k <- 10

cv\_results <- cv.glmnet(X, y, nfolds = k, alpha = 0)

# Calculate the mean squared error

cv\_mean\_squared\_error <- mean(cv\_results$cvm)

# Calculate CV PRESS

cv\_press\_backward\_p <- cv\_mean\_squared\_error \* nrow(combined\_data\_cleaned)

# Create a table with the backward p-value model's Adjusted R2 and CV PRESS

backward\_p\_table <- data.frame(

Predictive.Models = "Backward with p-value thresholding",

Adjusted.R2 = summary(backward\_p\_model)$adj.r.squared,

CV.PRESS = cv\_press\_backward\_p

)

print(backward\_p\_table)

```

```{r}

# Stepwise

# Fit the full model

full\_model <- lm(SalePrice ~ ., data = combined\_data\_cleaned)

# Perform stepwise regression with p-value thresholding for variable entry and exit

stepwise\_model <- ols\_step\_both\_p(full\_model, pent = 0.1, prem = 0.1)

# Get the variable names of the selected predictors

selected\_predictors <- names(coef(stepwise\_model)[-1])

# Check if selected predictors is empty or contains invalid variable names

if (length(selected\_predictors) == 0) {

stop("No variables selected for the model.")

}

if (any(!selected\_predictors %in% colnames(combined\_data\_cleaned))) {

stop("Invalid variable names in the selected predictors.")

}

# Create the formula object for the selected predictors

formula <- as.formula(paste("SalePrice ~", paste(selected\_predictors, collapse = "+")))

# Print the formula object

cat("Formula: ", deparse(formula), "\n")

# Convert your data frame to a matrix

X <- model.matrix(formula, data = combined\_data\_cleaned)

y <- combined\_data\_cleaned$SalePrice

# Perform k-fold cross-validation using glmnet

k <- 10

cv\_results <- cv.glmnet(X, y, nfolds = k, alpha = 0)

# Calculate the mean squared error

cv\_mean\_squared\_error <- mean(cv\_results$cvm)

# Calculate CV PRESS

cv\_press\_stepwise <- cv\_mean\_squared\_error \* nrow(combined\_data\_cleaned)

# Create a table with the stepwise model's Adjusted R2 and CV PRESS

stepwise\_table <- data.frame(

Predictive.Models = "Stepwise",

Adjusted.R2 = summary(stepwise\_model)$adj.r.squared,

CV.PRESS = cv\_press\_stepwise

)

print(stepwise\_table)

```

```{r}

# Custom

selected\_predictors <- c("OverallCond", "GarageCars", "ScreenPorch", "NeighborhoodBlueste", "NeighborhoodBrDale", "NeighborhoodClearCr", "NeighborhoodMeadowV", "NeighborhoodNPkVill", "NeighborhoodStoneBr", "NeighborhoodSWISU", "NeighborhoodTimber", "NeighborhoodVeenker")

# Create the formula object for the selected predictors

formula <- as.formula(paste("SalePrice ~", paste(selected\_predictors, collapse = "+")))

# Convert your data frame to a matrix

X <- model.matrix(formula, data = combined\_data\_cleaned)

y <- combined\_data\_cleaned$SalePrice

# Perform k-fold cross-validation using glmnet

k <- 10

cv\_results <- cv.glmnet(X, y, nfolds = k, alpha = 0)

# Calculate the mean squared error

cv\_mean\_squared\_error <- mean(cv\_results$cvm)

# Calculate CV PRESS

cv\_press\_custom <- cv\_mean\_squared\_error \* nrow(combined\_data\_cleaned)

# Create a table with the custom model's Adjusted R2 and CV PRESS

custom\_table <- data.frame(

Predictive.Models = "Custom",

Adjusted.R2 = summary(custom\_model)$adj.r.squared,

CV.PRESS = cv\_press\_custom

)

print(custom\_table)

```