Housing Price Prediction Project

by Christopher Garner, and Mai Dang

I. Introduction

Real estate markets are dynamic and complex with numerous factors influencing property prices, including but not limited to, the neighborhood, location, size, and the property conditions. As property sellers, Century 21, we want to know which features have the most significant impact on the housing prices. We also want to know whether we can accurately predict the price of a residential property based on which set of features, even across different neighborhoods in Ames, Iowa.

II. Data Description

- https://www.kaggle.com/c/house-prices-advanced-regression-techniques

Above is the link of our data source. The data for this Housing Price Prediction Project was obtained from Kaggle. The data consists of two main datasets: the training dataset and the test dataset. The train dataset contains 1,460 observations with a variety of property features and their corresponding actual sale prices. This dataset is used for model training and cross validation. The test dataset contains 1,459 observations with the same residential properties as those in the training set without the actual sale price. This dataset is used to analyze one city Ames, Iowa, across different neighborhoods and to make predictions of the unknown sale prices based on these same sets of features. This would mirror the actual experience of a real estate company, looking back at recent sale figures, gathering the pertinent data, and using it to predict future sale prices from the results.

III. Analysis Question 1:

A. Restatement of Problem

We want to check whether the sale price of a house under 4,000 square feet of above ground living area is related to the square footage of the living area of the house (GrLivArea) and the neighborhood the house located in, focusing solely on three distinct neighborhoods: Edwards, North Ames, and Brookside.

B. Build the Model

Before removing the outliers

After removing the outliers

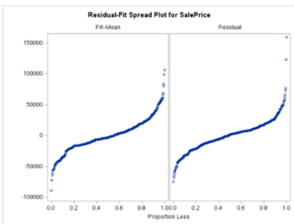
We felt that our data had some minor issues in assumptions but that we were best to leave it as linear- linear as discussed more thoroughly below. We also removed two properties that have above round living area larger than 4,000 square feet. From that decision, we constructed the initial equation below to represent out Sale Price equation:

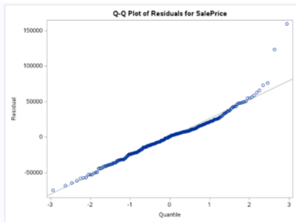
$$\hat{\mu}\{SalePrice\ |NEIGHBORHOOD, GrLivArea\} = \beta_0 + \beta_1 * Edwards + \beta_2 BrkSide + \beta_3\ GrLivArea + \beta_4 l\ GrLivArea * Edwards + \beta_5\ GrLivArea * BrkSide$$

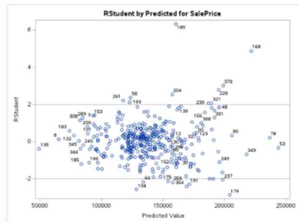
We then continued with our analysis:

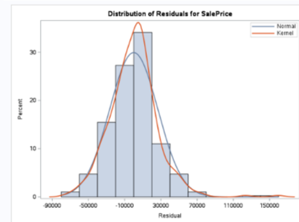
C. Checking Assumptions

i. Residual Plots



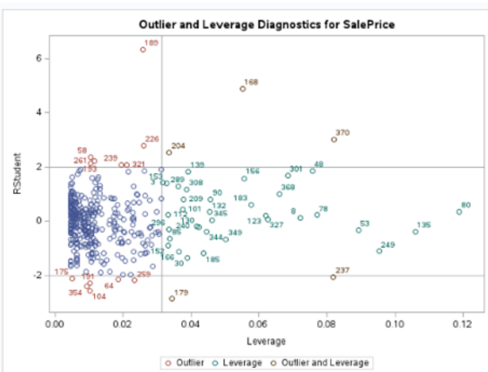


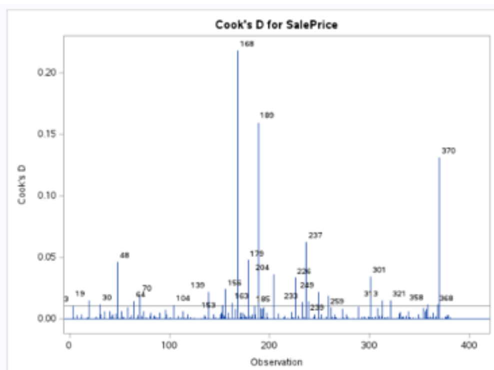




Above are the residual plots for sale prices of properties that have living area less than 4,000 square feet. The visual testes suggest the normal distribution of the residuals. They appear to be random cloud around the zero line.

ii. Influential point analysis (Cook's D and Leverage)





Even after two properties that have living area higher than 4,000 square feet are removed from the dataset, the leverage and the Cook's D plots for Sale Prices yield some concerns about the outliers that appears to be in the high residual and high leverage zone and an observation that has very low residual and high leverage. However, the scale of the leverage is relatively small, and these outliers all have leverage lower than 0.10. Therefore, they are not very concerning to our model. We will proceed to make our assumptions.

iii. Assumptions:

- **Independency:** We assume that the observations are independent although we do this with caution as the observations in the same neighborhood may have similar features and thus, similar sale prices than those from the other neighborhoods.
- **Linearity**: It is hard to assume the linearity with multiple variables. As seen in the scatter plot of the original data, there are two outliers that would influence the linearity between the sale price and the living area in the interest neighborhoods of interest. We removed these two observations that have the living area that are larger than 4,000 square feet. Based on the new scatter plot of the sale prices and the above ground living area of the houses that are under 4,000 square feet in the three neighborhoods (Names,

BrkSide, and Edwards), we can assume that there is adequate linearity between the property sale prices and their corresponding above ground living area.

- **<u>Constant variances:</u>** There is no evidence in the scatter plots of the residuals to suggest that they follow a certain pattern. Based on the random cloud of the residuals, we assume the equal variance of the variations at any size of the living area under 4,000 square feet.
- **<u>Normality:</u>** Two properties that have total living area higher than 4,000 squares also have high Cook' D and high influence on our model. Therefore, they were removed from our data and thus, resulted in our model consists only to the houses with total living area less than 4000 square feet. Judging from the qq-plot and the histograms of the residuals after the outliers are removed, is performed, there is no significant evidence to suggest that the residuals do not follow a normal distribution. According to the Cook's D and the leverage plot, there are some influential points and some observations that have leverage over 0.05. However, the visual tests do not yield a concern as the highest Cook's D observation is still under 0.2. We assume that the residuals are normally distributed. We will proceed fitting the model with only observations that have the total living area less than 4,000 square feet.

D. <u>Fit the Model:</u>

$$\hat{\mu}\{SalePrice\ |NEIGHBORHOOD,\ GrLivArea\} = 19972 + 11457 * Edwards + 54{,}705 NAmes$$
$$+87.162 GrLivArea - 11.186\ GrLivArea * Edwards - 32.847\ GrLivAre * NAmes$$

The R^2 of our fitted model is 0.5125 and the adjusted R^2 is 0.506.

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	5	2.836316E11	56726317745	78.83	<.0001
Error	375	2.698472E11	719592541		
Corrected Total	380	5.534788E11			

Root MSE	26825	R-Square	0.5125
Dependent Mean	137882	Adj R-Sq	0.5060
Coeff Var	19.45515		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|	95% Confidence Limits	
Intercept	1	19972	11604	1.72	0.0861	-2845.60572	42789
Edwards	1	11457	15343	0.75	0.4557	-18713	41627
NAmes	1	54705	13043	4.19	<.0001	29059	80351
GrLivArea	1	87.16253	9.19027	9.48	<.0001	69.09162	105.23345
int1	1	-11.18610	11.96277	-0.94	0.3504	-34.70861	12.33642
int2	1	-32.84667	10.16117	-3.23	0.0013	-52.82669	-12.86665

E. <u>Parameters</u>

<u>For houses in Brook Side neighborhood:</u>

- **<u>Intercept $\boldsymbol{\beta_0}$:</u>** There is not significant evidence suggest if the living area is neglected, the sale price of the properties in the Brook Side neighborhood is different than 0 (p-value =0.08606). The estimated 19,872 is an extrapolation as it may not make sense practically to have a house with above ground living area of 0 square feet. Therefore, the intercept in this case is more of a theoretically extrapolated value. It suggests a baseline for a linear relationship between the house prices in the Brook Side neighborhood and the living area. 95 % confidence interval of this estimated is ($-2,846, $42,789).
- **<u>Slope $\boldsymbol{\beta_3}$:</u>** There is strong evidence to suggest the association between the mean sale price and the living areas of the properties that have under 4,000 square feet of the above ground living area in the Brook Side neighborhood (p-value < 0.001). It estimated that for every 100 square feet increase in the above ground living area of those properties, the mean sale price increase $8,716$. A 95 % confidence interval of this estimated is ($6,909, $10,523).

<u>For houses in Edwards neighborhood:</u>

- **<u>Adjustment terms $\boldsymbol{\beta_1}$:</u>** There is not significant evidence to suggest a difference in sale prices between properties in the Edward neighborhoods and those in the Brook Side neighborhood (p-value =0.4557) if the above ground living area is 0. The estimated difference $11,457 is an extrapolation as it may not make sense practically to have a house with above ground living area of 0 square feet. Therefore, the intercept in this case is more of a theoretically extrapolated value. It suggests a baseline for a linear relationship between the house prices in the Edwards neighborhood and the living area. 95 % confidence interval of this estimated is ($-18,713, $41,627).

- **Interaction term β_4:** There not enough significant evidence to suggest a change from Brook Side to the Edwards neighborhood would affect the mean sale price of the properties under 4,000 square feet of the total living area, for every 100 square feet increase in the total living area of the properties (p-value = 0.35035). In another word, for every 100 square feet increase in the total above ground living area, the increase in the mean sale price of the properties in Edwards neighborhood may be no different than the increase in mean sale prices of the properties of the same above ground living area in the Brook Side neighborhood.

For houses in BrkSide neighborhood:

- **Adjustment terms β_2 :** A change from the Brook Side to the North Ames neighborhood appears to be significantly associated with $54,705 increasing in the mean sale prices of the properties of over ground living area of 0 square feet (p-value < 0.0001). Again, this is an extrapolation and may not practically make sense to have a total above ground living area of 0 square feet. A 95 % confidence interval of this estimated increase is ($29,059, $80,351).
- **Interaction term β_5:** There is significant evidence to suggest a change from Brook Side to the North Ames neighborhood also factor in the increase in the mean sale price of the properties under 4,000 square feet of the total living area for every 100 square feet increase in the total above ground living area of the properties (p-value = 0.0013). In another word, for every 100 square feet increase in the total above ground living area, the increase in the mean sale price of those properties in the North Ames neighborhood is approximately $3,284.67 less than the increase in the mean sale price of the properties in the North Ames neighborhood. A 95% confidence interval of this estimated is ($1,287, $5,283)

F. Conclusion

For the properties that have a total living area under 4,000 square feet in three neighborhoods, Names, BrkSide, and Edwards, there is an association between the property sale prices and its neighborhood as well as its total living area.

IV. R Shiny: Price v. Living Area Chart

Below are the hyperlinks to our R Shiny App.

- https://ctg-smu-data.shinyapps.io/Housing_Ames_Iowa/
- https://maidang.shinyapps.io/Rshiny/

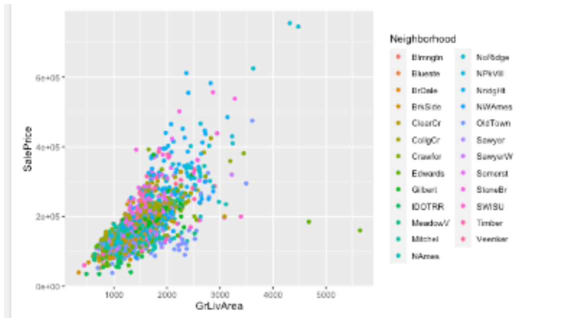
V. Analysis Question 2

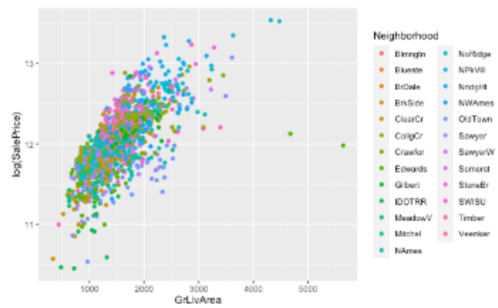
A. Restatement of Problem

From the previous analysis, we know that there is an association between the sale prices and the property neighborhood as well as its total living area. Therefore, we want to build the most predictive model for sale prices of homes in all of Ames, Iowa, including all neighborhoods.

B. Model Selection

Sale Price vs. GrLivArea by Neighborhood	Logged Sale Price vs. GrLivArea by Neighborhood

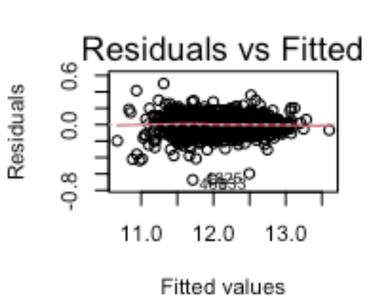


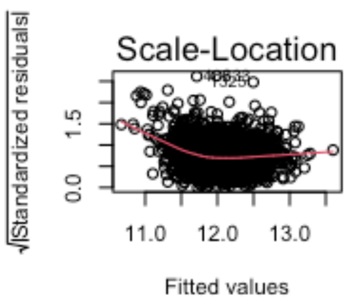


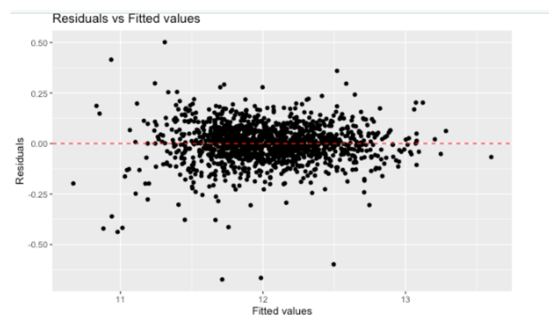
As seen in the scatter plots of the sale price and the total living area by the neighborhood before and after the sale price is log transformed, the linearity improves with the log transformation. We excluded some of the variables that do not have a strong impact to the sale price, such as Utilities, Alley, Id, etc. We fit the rest of the variables as predictors to predict the sale price using the three variable selections: forward, backward, and stepwise with the original sale prices as the response variable. We also fit the model with the logged sale price to check whether the log transformation would improve the house price predictions. Afterward, the backward model with the original dataset and the backward model using the logged sale prices yield different adjusted R^2 and RMSE but the same Kaggle score. We proceeded with the custom model using the logged sale prices and backward variable selection for the assumptions and the visual tests.

C. <u>Checking Assumptions</u>

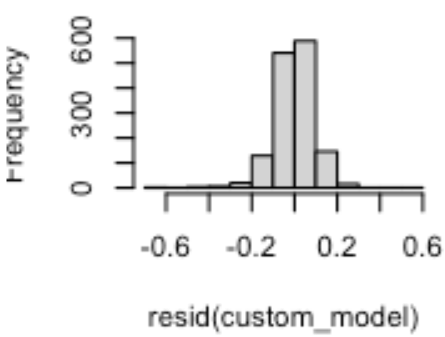
i. <u>Residual Plots</u>

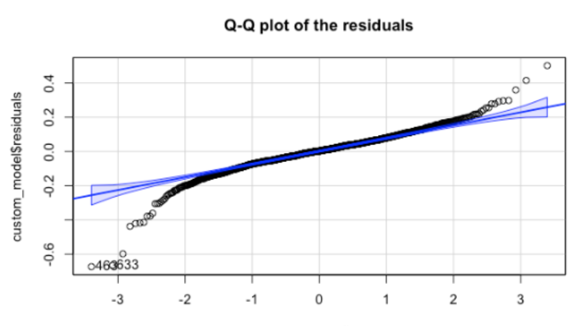






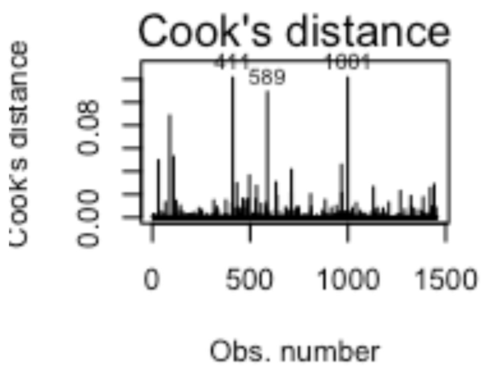
As seen in the scatter plots of the sale prices versus the above ground living area by neighborhood, there are some outliers. We removed three outliers in this dataset for two reasons. One outlier is a building, instead of residential properties. The other two outliers have above ground living area higher than 4,500 square feet. After removing the outliers, the residual plots appear to be a random cloud. We assume the equal variances between the observations in our model.

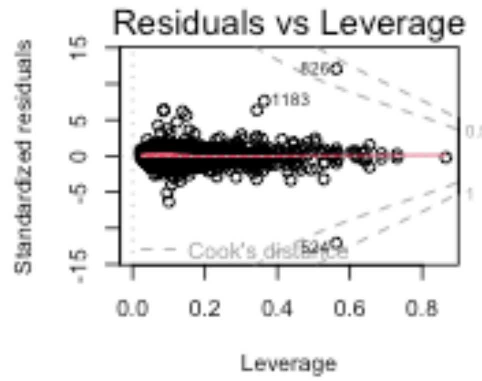




There is also not a lot of evidence to suggest that the residuals are not normally distributed. Therefore, we proceeded to make assumptions for our custom model.

ii. Influential point analysis (Cook’s D and Leverage)





Above are the Cook’sD and the leavrage plot which help us to determine if there are still high influential points and outliers that would make a significant impact on the linearity of our custom model. We will analyze the plots more throughout in the assumptions.

iii. Assumptions:

- **Independency:** We assume that the observations are independent although we do this with caution as the observations in the same neighborhood may have similar features and thus, similar sale prices than those from the other neighborhoods.
- **Linearity**: It is hard to assume the linearity with multiple variables. There are more than forty predictors that are selected using the backward selection by p-value. We will proceed with an assumption of the linearity between the logged sale prices and the variables selected in our custom model but thith caution as noted herein.
- **Constant variances:** There is no evidence in the scatter plots of the residuals to suggest that they follow a certain pattern. Based on the random cloud of the residuals, we assume the equal variance of the the house properties in our dataset.
- **Normality:** Three properties that have a total living area higher than 4,500 squares also have high Cook’s D and high leverage. They are removed from our data and thus, the results of our model pertains only to the properties of total above ground living area less than 4,500 square feet. Judging from the qq-plot and the histograms of the residuals after the outliers are removed, is performed, there is no significant evidence to suggest that the residuals do not follow a normal distribution. According to the Cook’s D and the leverage plot, there are some influential points and some observations that have leverage over 0.08. However, the visual tests do not yield a concern as the highest Cook’s D observation is still relatively small. We assume that the residuals are normally distributed. We will proceed fitting the model with only observations that have the total above ground living area less than 4,500 square feet.

D. Comparing Competing Models

Predictive Models	Adjusted R2	RMSE (from internal cross-validation)	Kaggle Score
Forward	.9154	46429	.14458
Backward	.9174	44766	.14442
Stepwise	.9154	46429	.14458
CUSTOM	.9359	0.19374(log scale)	.14442

VI. <u>Conclusion: A short summary of the analysis.</u>

Surprisingly, the custom model with log transformation and backward selection performs as well as the original data with only the back ward selection in the Kaggle competition as well as the adjusted R^2. We also implement the "LOOCV" cross-validation to evaluate the models. We believe the RMSE in the custom model is not accurate as well as the RMSE in the other models are too high compared to regular housing prices.

VII. <u>Links to Our GitHub Pages:</u>

- **Mai Dang: https://maedang.github.io/**
- **Christopher Garner: https://smu-datahub.github.io/**

<u>Appendix</u>

Codes for the Analysis 1

<table>
<tr>
<td>
/*Display the train dataset
proc print data = train;
run;

/* Part1: 3 GrLivArea vs Sale Price by Neighborhood*/

/* Filter for only 3 neighborhoods */
data filtered_train;
set train;
where Neighborhood in ('NAmes', 'Edwards', 'BrkSide');
run;

/* Keep only the 'Neighborhood', 'SalePrice', and 'GrLivArea' columns */
data reduced_train;
set filtered_train(keep=Neighborhood SalePrice GrLivArea);
run;

/* Add interactions and dummy variables */
data train2;
set reduced_train;
if Neighborhood = "Edwards" then Edwards = 1; else Edwards=0;
if Neighborhood = "NAmes" then NAmes= 1; else NAmes = 0;
int1 = Edwards*GrLivArea;
int2 = NAmes*GrLivArea;
run;

/* Remove outliers; */
data remove_outliers;
set train2;
if _n_ = 339 then delete;
if _n_ = 131 then delete;
/*Just to check what happen if more outliers are removed*/
/* if _n_ = 168 then delete;   */
/* if _n_ = 189 then delete;   */
run;

/* Plots */
symbol1 v= "B" C = black I = none;
symbol2 v = 'E' c = red I = none;
symbol3 v = "N" c = green I = none;
</td>
<td>
library(tidyverse)
library(ggplot2)
library(scales)
library(pwr)
library(agricolae)
library(huxtable)
library(lawstat)
library(lsmeans)
library(nCDunnett)
library(dplyr)
library(WDI)
library(investr)
library(multcomp)
library(pairwiseCI)
library(DescTools)
library(gridExtra)
library(car)
library(caret)
library(olsrr)
library(tidyverse)

# Load the file
# Load the data
df1 <- read.csv(file.choose(), header = TRUE)
head(df1)

# Removing all unnecessary columns and filter out the neighborhoods
# Subset the data frame to keep only the specified columns
df1 <- df1[, c("SalePrice", "GrLivArea", "Neighborhood")]
df1 <- df1 %>% filter(Neighborhood %in% c("NAmes", "Edwards", "BrkSide"))
head(df1)

#Scatter Plot of GrLivArea vs SalePrice by Neighborhood
#GrLiving area vs SalePrice by Neighborhood
ggplot(df1, aes(x=GrLivArea, y = SalePrice, color= Neighborhood)) + geom_point()

#Build the model with original data + Plots
# Fit the model
model <- lm(SalePrice ~ GrLivArea + Neighborhood, data = df1)
</td>
</tr>
</table>

```
title "Sale Prices vs Living Area by the
Neighborhood";


proc gplot data = remove_outliers;
plot SalePrice*GrLivArea= Neighborhood; run;


/* Matrix Plot */
proc sgscatter data = remove_outliers;
matrix SalePrice GrLivArea Edwards NAmes ;
run;

/* Models  */
proc reg data = remove_outliers plots(label) =
(CooksD all);
model SalePrice = Edwards NAmes GrLivArea
int1 int2  / clb;
run;
```

```
# Print the summary statistics of the model's
performance
summary(model)

# Create a plot of residuals vs fitted values
ggplot(df1, aes(x = model$fitted.values, y =
model$residuals)) +
  geom_point() +
  geom_hline(yintercept=0, color="red",
linetype="dashed") +
  labs(title="Residuals vs Fitted values", x="Fitted
values", y="Residuals")

# Create a Q-Q plot of the residuals
qqPlot(model$residuals, main="Q-Q plot of the
residuals")

# Compute Cook's distance
df1$CooksD <- cooks.distance(model)

# Create a histogram of Cook's distance
hist(df1$CooksD, main="Cook's Distance",
xlab="Cook's Distance")

# Remove the 2 highest GrLivArea
# Remove specific rows by index.  The high
square footage were outliers (row 339, 131).
rows_to_remove <- c(339, 131)
df1 <- df1[-rows_to_remove, ]

# Fit the model with interaction terms
fit <- lm(SalePrice ~ GrLivArea * Neighborhood,
data = df1)

# Print the summary
summary(fit)

# Compute VIF for the model
vif_model <- vif(fit, type=c("predictor"))
vif_model

# QQ plot of the residuals
qqnorm(residuals(fit))
qqline(residuals(fit))

# Diagnostic plots
par(mfrow = c(2, 3))

# Residuals vs Fitted Values
plot(fit, which = 1)
```

	# Scale-Location (also called Spread-Location)
	plot(fit, which = 3)
	
	# Cook's distance plot
	plot(fit, which = 4)
	
	# Residuals vs Leverage
	plot(fit, which = 5)
	
	# Histogram of residuals
	hist(resid(fit))
	
	#Confidence interval and predictional intervals of each slope
	ggplot(df1, aes(x=GrLivArea, y = SalePrice, color = Neighborhood)) + geom_point()
	+geom_smooth(method = "lm")

A. Codes for the Analysis 2

```
#Libraries
library(tidyverse)
library(ggplot2)
library(scales)
library(pwr)
library(agricolae)
library(huxtable)
library(lawstat)
library(lsmeans)
library(nCDunnett)
library(dplyr)
library(WDI)
library(investr)
library(multcomp)
library(pairwiseCI)
library(DescTools)
library(gridExtra)
library(car)
library(caret)
library(olsrr)
library(tidyverse)
library(corrplot)
library(MASS)

#Load csv files
train <- read.csv(file.choose(), header = TRUE)
test <- read.csv(file.choose(), header = TRUE)
#Add SalePrice in the test_dataset
```{r}
test$SalePrice <- NA
```
````

````
```
#Combine 2 dataset for cleaning
```{r}
# Combine the train and test sets
all_data <- rbind(train, test)
```
###Finding the NA values
```{r}
#Find the na values
columns_with_na <- names(all_data)[colSums(is.na(all_data)) > 0]
# Display the column names with NA values
print(columns_with_na)
```

#Replace the Na values
```{r}
# Step 1: Handle missing values for numeric variables
numeric_cols <- c("LotFrontage", "MasVnrArea", "GarageYrBlt", "BsmtFinSF1", "BsmtFinSF2",
         "BsmtUnfSF", "TotalBsmtSF", "BsmtFullBath", "BsmtHalfBath", "GarageCars",
"GarageArea")

# Replace missing values in numeric columns with the mean
all_data[numeric_cols] <- lapply(all_data[numeric_cols], function(x) ifelse(is.na(x), median(x, na.rm =
TRUE), x))

# Step 2: Handle missing values for categorical variables
categorical_cols <-
c("MSZoning","Alley","Utilities","Exterior1st","Exterior2nd","MasVnrType","BsmtQual",
          "BsmtCond","BsmtExposure","BsmtFinType1","BsmtFinType2", "Electrical",
"KitchenQual",
          "Functional", "FireplaceQu", "GarageType", "GarageFinish","GarageQual" ,
"GarageCond" ,
          "PoolQC","Fence","MiscFeature", "SaleType")
# Replace missing values in categorical columns with the mode
for (col in categorical_cols) {
  mode_val <- names(which.max(table(all_data[[col]])))
  all_data[[col]][is.na(all_data[[col]])] <- mode_val
}
```
#Factors all of the categorical variables
```{r}
all_data <- all_data %>% mutate_if(is.character, factor)
```

#Split the traina dn test set after cleaning process
```{r}
# Create train_data by removing rows with NA in SalePrice
train_data <- all_data[!is.na(all_data$SalePrice), ]
# Create test_data by keeping only rows with NA in SalePrice
test_data <- all_data[is.na(all_data$SalePrice), ]
````

````
```
#Split the train_data to train and test set
```{r}
#Divided the train_data to trai_set and test_set
# Calculate the number of rows for the train and test sets
num_rows <- nrow(train_data)
num_train_rows <- round(0.8 * num_rows)
# Generate random row indices for the train and test sets
train_indices <- sample(seq_len(num_rows), size = num_train_rows, replace = FALSE)
test_indices <- setdiff(seq_len(num_rows), train_indices)
# Create the train_set and test_set DataFrames
training_set <- train_data[train_indices, ]
testing_set <- train_data[test_indices, ]
```

#Visualization to check whether transformation is need
```{r}
#GrLiving area vs SalePrice by Neighborhood
ggplot(train_data, aes(x=GrLivArea, y = SalePrice, color= Neighborhood)) + geom_point()
#GrLiving area vs Logged SalePrice by Neighborhood
ggplot(train_data, aes(x=GrLivArea, y = log(SalePrice), color= Neighborhood)) + geom_point()

#GrLiving area vs SalePrice by Neighborhood
ggplot(train_data, aes(x= LotArea, y = SalePrice, color= OverallQual)) + geom_point()

#GrLiving area vs Logged SalePrice by Neighborhood
ggplot(train_data, aes(x= LotArea, y = log(SalePrice), color= OverallQual)) + geom_point()
```
#Corrplot
```{r}
# Select only numeric columns from the data frame
numeric_data <- select_if(train_data, is.numeric)

# Compute the correlation matrix for numeric variables
cor_matrix <- cor(numeric_data)

# Customize corrplot (e.g., change color, method, etc.)
# For more customization options, check the documentation: ?corrplot
corrplot(cor_matrix, method = "color", tl.cex = 0.7)
```
#Display the counts of each level of the categorial varibles
```{r}
for (col in names(train_data)) {
  if (is.factor(train_data[[col]])) {
    cat("Variable:", col, "\n")
    cat(table(train_data[[col]]), "\n\n")
  }
}
```

#Remove Utilies, PoolQC, MiscFeature, Id, and Alley
````
````

````
```{r}
train_data <- train_data[ , -c(1, 7, 10, 73, 75)]

```


#Fit the model with original data
```{r}
#Fit the model
fit = lm(SalePrice~., data = train_data)
summary(fit)

```


#Backward selection
```{r}
#Backwardselection
backward <- ols_step_backward_p(fit, prem = 0.05, details= FALSE)
summary(backward$model)
#Make prediction and unlog the logged sale price
prediction_bw <- predict(backward$model, newdata= test_data)
predictions_bw
#Write csv
submission_bw_path =
"/Users/maidang/Desktop/MSDS/Term1_Summer2023/MSDS_6371_Stat_Foundations/Project/Resources/submission_backward_nolog.csv"
#Create a new data frame with Id and SalePrice columns
submission_bw <- data.frame(Id = test_data$Id, SalePrice = predictions_bw)
#Save the new data frame to a CSV file
write.csv(submission_bw, submission_bw_path, row.names = FALSE)

```
#List all of the variables removed in the backward selection
```{r}
column_remove_bw <-backward$removed
column_remove_bw
```
#Cross Validation with the BAckward Selection
```{r}

# Load the caret package
library(caret)
#Define trainign control
train_control<- trainControl(method="LOOCV")

# train the model
model_bw <- train(SalePrice~ MiscVal + CentralAir + Electrical + Heating + GarageYrBlt +
BsmtHalfBath +
 + ExterCond + MSSubClass + Exterior2nd + FireplaceQu + EnclosedPorch + BsmtFullBath +
+ LotShape + GarageType + HalfBath + PavedDrive + OpenPorchSF + YrSold +
+ HeatingQC + SaleType + Foundation + X3SsnPorch + LotFrontage + GarageFinish +
+ FullBath + MoSold + MasVnrType + ScreenPorch + BsmtFinType2 + WoodDeckSF +
````
````

````
+ RoofStyle + BsmtCond + GarageArea + TotRmsAbvGrd
, data=train_data, trControl=train_control, method="lm")

#Model Result using the LOOVC trainingcontrol
model_bw

```


#Foward Selection
```{r}
#Forward selection
forward <- ols_step_forward_p(fit, penter = 0.05, details= FALSE)
forward$adjr
#Make prediction and unlog the logged sale price
prediction_fw <- predict(forward$model, newdata= test_data)
predictions_fw
#Write csv
submission_fw_path =
"/Users/maidang/Desktop/MSDS/Term1_Summer2023/MSDS_6371_Stat_Foundations/Project/Resour
ces/submission_fworward.csv"
#Create a new data frame with Id and SalePrice columns
submission_fw <- data.frame(Id = test_data$Id, SalePrice = predictions_fw)
#Save the new data frame to a CSV file
write.csv(submission_fw, submission_fw_path, row.names = FALSE)

#List all the variables selected for the forward selection
forward$predictors
# train the model
model_fw <- train(SalePrice~ OverallQual + GrLivArea + Neighborhood + BsmtQual + RoofMatl +
BsmtFinSF1 +
MSSubClass + BsmtExposure + KitchenQual + Condition2 + OverallCond + YearBuilt +
LotArea + SaleCondition + GarageArea + PoolArea + ExterQual + TotalBsmtSF +
Functional + BedroomAbvGr + BldgType + Exterior1st + MasVnrArea + Condition1 +
LandSlope + MSZoning + LandContour + LowQualFinSF + Street + ScreenPorch +
LotConfig + YearRemodAdd + KitchenAbvGr + Fireplaces
, data=train_data, trControl=train_control, method="lm")

#Model Result using the LOOVC trainingcontrol
train.control = trainControl(method = "cv", number = 20)
model_tr_result <- train(SalePrice~.,  data = train_data, method = "lm",
trControl = train.control)
model_tr_result$results
```


#Stepwise Selection
```{r}
#Stepwise selection
stepwise <- ols_step_both_p(fit, prem = 0.05, penter = 0.05, details= FALSE)
stepwise$adjr
#Make prediction and unlog the logged sale price
prediction_sw <- predict(stepwise$model, newdata= test_data)
````

```
predictions_sw
summary(stepwise)
#Write csv
submission_sw_path =
"/Users/maidang/Desktop/MSDS/Term1_Summer2023/MSDS_6371_Stat_Foundations/Project/Resources/submission_stepwise.csv"
#Create a new data frame with Id and SalePrice columns
submission_sw <- data.frame(Id = test_data$Id, SalePrice = predictions_sw)
#Save the new data frame to a CSV file
write.csv(submission_sw, submission_sw_path, row.names = FALSE)
#List all the variables are selected in the stepwise selection
stepwise$predictors

# train the model
model_sw <- train(SalePrice~  OverallQual + Neighborhood + GrLivArea + GarageArea + OverallCond +
RoofMatl + TotalBsmtSF + YearBuilt + Condition1 + Condition2 + MSZoning + BsmtFinSF1 + LotArea + ScreenPorch + Fireplaces + BsmtExposure + Exterior1st + YearRemodAdd +LandSlope + GarageArea + LotConfig + BsmtQual + RoofMatl + MSSubClass + KitchenQual + SaleCondition + PoolArea + ExterQual + Functional + BedroomAbvGr + BldgType + MasVnrArea + LandContour + LowQualFinSF + Street + YearRemodAdd + KitchenAbvGr, data=train_data, trControl=train_control, method="lm")
#Model Result using the LOOVC trainingcontrol
model_sw

```
#Fit the model with log transformation
```{r}
#Log transformations to get logPrice
train_data$logprice <- log(train_data$SalePrice)
#Fit the model
log_fit = lm(logprice~.-SalePrice, data = train_data)
summary(log_fit)

```
#Remove outliers
```{r}
rows_to_remove <- c(1183,826,524)
train_data<- train_data[-rows_to_remove, ]
```

```{r}
#Backwardselection
backward_log <- ols_step_backward_p(log_fit, prem = 0.05, details= FALSE)
summary(backward_log$model)
#Make prediction and unlog the logged sale price
logged_predictions_bw_log<- predict(backward_log$model, newdata= test_data)
prdeictions_bw_log <- exp(logged_predictions_bw_log)
#Write csv
submission_bw_log_path =
"/Users/maidang/Desktop/MSDS/Term1_Summer2023/MSDS_6371_Stat_Foundations/Project/Resources/submission_backward_log.csv"
```

```
#Create a new data frame with Id and SalePrice columns
submission_bw_log <- data.frame(Id = test_data$Id, SalePrice = predictions_bw_log)
#Save the new data frame to a CSV file
write.csv(submission_bw_log, submission_bw_log_path, row.names = FALSE)
```
#List all of the variables removed in the backward selection
```{r}
column_remove_bw <-backward_log$removed
column_remove_bw
```
#Cross Validation with the BAckward Selection
```{r}

# Load the caret package
library(caret)
#Define trainign control
train_control<- trainControl(method="LOOCV")

# train the model
model_bw_log <- train(logprice~ MSZoning + LotFrontage + LotArea + Street + LotConfig + LandSlope + Neighborhood + Condition1 + BldgType + OverallQual + OverallCond + YearBuilt + YearRemodAdd + RoofMatl + Exterior1st + ExterCond + Foundation + BsmtExposure + BsmtFinSF1 + BsmtFinSF2 + BsmtUnfSF + TotalBsmtSF + Heating + HeatingQC + CentralAir + X1stFlrSF + X2ndFlrSF + LowQualFinSF + GrLivArea + BsmtFullBath + KitchenAbvGr + KitchenQual + Functional + Fireplaces + GarageCars + GarageArea + GarageQual + GarageCond + WoodDeckSF + OpenPorchSF + EnclosedPorch + ScreenPorch + SaleType + SaleCondition
  , data=train_data, trControl=train_control, method="lm")

#Model Result using the LOOVC trainingcontrol
model_bw_log
```
#Residual Plots
```{r}
#Fit the model
custom_model = lm(logprice~ MSZoning + LotFrontage + LotArea + Street + LotConfig + LandSlope + Neighborhood + Condition1 + BldgType + OverallQual + OverallCond + YearBuilt + YearRemodAdd + RoofMatl + Exterior1st + ExterCond + Foundation + BsmtExposure + BsmtFinSF1 + BsmtFinSF2 + BsmtUnfSF + TotalBsmtSF + Heating + HeatingQC + CentralAir + X1stFlrSF + X2ndFlrSF + LowQualFinSF + GrLivArea + BsmtFullBath + KitchenAbvGr + KitchenQual + Functional + Fireplaces + GarageCars + GarageArea + GarageQual + GarageCond + WoodDeckSF + OpenPorchSF + EnclosedPorch + ScreenPorch + SaleType + SaleCondition
  , data=train_data)
# Create a plot of residuals vs fitted values
ggplot(train_data, aes(x = custom_model$fitted.values, y = custom_model$residuals)) +
  geom_point() +
  geom_hline(yintercept=0, color="red", linetype="dashed") +
  labs(title="Residuals vs Fitted values", x="Fitted values", y="Residuals")

# Create a Q-Q plot of the residuals
qqPlot(custom_model$residuals, main="Q-Q plot of the residuals")
```

````
```
#Residual Plots
```{r}
# Diagnostic plots
par(mfrow = c(2, 3))

# Residuals vs Fitted Values
plot(custom_model, which = 1)

# Scale-Location (also called Spread-Location)
plot(custom_model, which = 3)

# Cook's distance plot
plot(custom_model, which = 4)

# Residuals vs Leverage
plot(custom_model, which = 5)

# Histogram of residuals
hist(resid(custom_model))
```
````
````