# milestone4_630

Bobga-Herman Gwanvoma

2025-02-11

```
# Load necessary libraries
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.4.1
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.4.1
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.4.2
```

```
## Loading required package: lattice
```

```
library(xgboost)
```

```
## Warning: package 'xgboost' was built under R version 4.4.2
```

```
##
## Attaching package: 'xgboost'
```

```
## The following object is masked from 'package:dplyr':
##
##     slice
```

```r
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.4.2
```

```
## corrplot 0.95 loaded
```

```r
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 4.4.2
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
# Install XGBoost if not already installed
if (!requireNamespace("xgboost", quietly = TRUE)) {
  install.packages("xgboost")
}

# Load the package
library(xgboost)
```

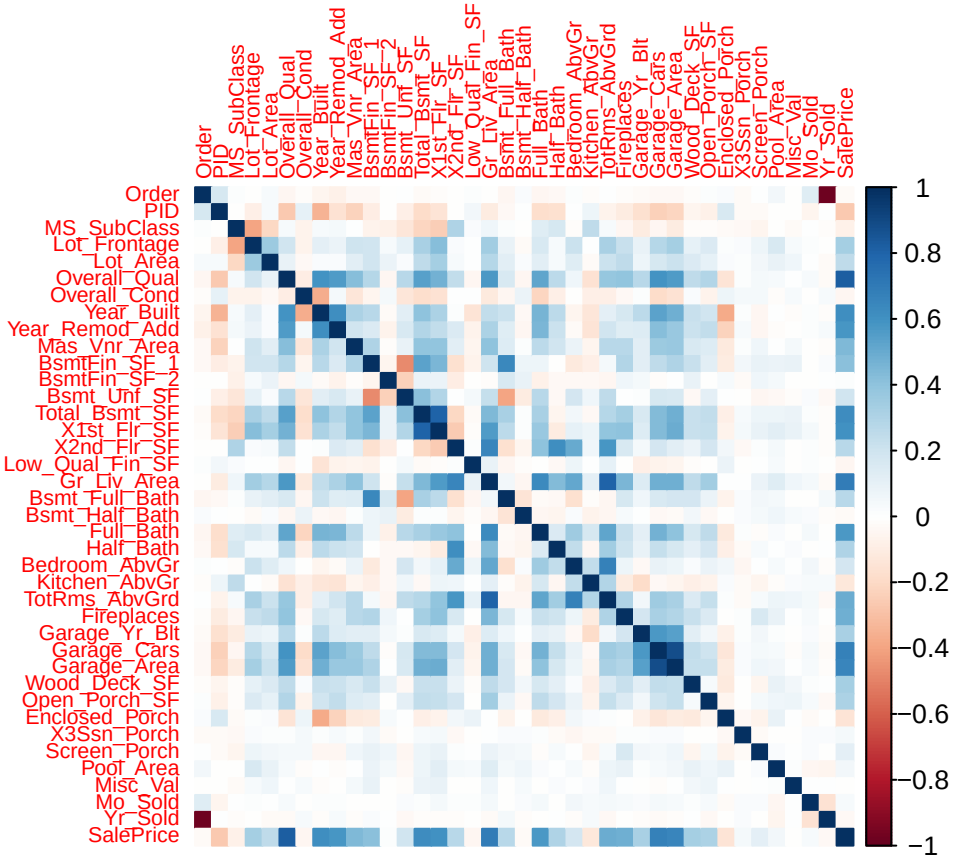
```r
# Load the Ames Housing dataset (Ensure the file is in your working directory)
ames <- read.csv("C:/Users/bobi/Documents/DSC630/AmesHousing.csv")

# Standardize column names by replacing dots with underscores
names(ames) <- gsub("\\.", "_", names(ames))

# Inspect the dataset structure
str(ames)
```

```
## 'data.frame':    2930 obs. of  82 variables:
##  $ Order          : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ PID            : int  526301100 526350040 526351010 526353030 527105010 527105030 527127150 52714
##  $ MS_SubClass    : int  20 20 20 20 60 60 120 120 120 60 ...
##  $ MS_Zoning      : chr  "RL" "RH" "RL" "RL" ...
##  $ Lot_Frontage   : int  141 80 81 93 74 78 41 43 39 60 ...
##  $ Lot_Area       : int  31770 11622 14267 11160 13830 9978 4920 5005 5389 7500 ...
##  $ Street         : chr  "Pave" "Pave" "Pave" "Pave" ...
##  $ Alley          : chr  NA NA NA NA ...
##  $ Lot_Shape      : chr  "IR1" "Reg" "IR1" "Reg" ...
##  $ Land_Contour   : chr  "Lvl" "Lvl" "Lvl" "Lvl" ...
##  $ Utilities      : chr  "AllPub" "AllPub" "AllPub" "AllPub" ...
##  $ Lot_Config     : chr  "Corner" "Inside" "Corner" "Corner" ...
##  $ Land_Slope     : chr  "Gtl" "Gtl" "Gtl" "Gtl" ...
##  $ Neighborhood   : chr  "NAmes" "NAmes" "NAmes" "NAmes" ...
##  $ Condition_1    : chr  "Norm" "Feedr" "Norm" "Norm" ...
```

```
##  $ Condition_2    : chr  "Norm" "Norm" "Norm" "Norm" ...
##  $ Bldg_Type      : chr  "1Fam" "1Fam" "1Fam" "1Fam" ...
##  $ House_Style    : chr  "1Story" "1Story" "1Story" "1Story" ...
##  $ Overall_Qual   : int  6 5 6 7 5 6 8 8 8 7 ...
##  $ Overall_Cond   : int  5 6 6 5 5 6 5 5 5 5 ...
##  $ Year_Built     : int  1960 1961 1958 1968 1997 1998 2001 1992 1995 1999 ...
##  $ Year_Remod_Add : int  1960 1961 1958 1968 1998 1998 2001 1992 1996 1999 ...
##  $ Roof_Style     : chr  "Hip" "Gable" "Hip" "Hip" ...
##  $ Roof_Matl      : chr  "CompShg" "CompShg" "CompShg" "CompShg" ...
##  $ Exterior_1st   : chr  "BrkFace" "VinylSd" "Wd Sdng" "BrkFace" ...
##  $ Exterior_2nd   : chr  "Plywood" "VinylSd" "Wd Sdng" "BrkFace" ...
##  $ Mas_Vnr_Type   : chr  "Stone" "None" "BrkFace" "None" ...
##  $ Mas_Vnr_Area   : int  112 0 108 0 0 20 0 0 0 0 ...
##  $ Exter_Qual     : chr  "TA" "TA" "TA" "Gd" ...
##  $ Exter_Cond     : chr  "TA" "TA" "TA" "TA" ...
##  $ Foundation     : chr  "CBlock" "CBlock" "CBlock" "CBlock" ...
##  $ Bsmt_Qual      : chr  "TA" "TA" "TA" "TA" ...
##  $ Bsmt_Cond      : chr  "Gd" "TA" "TA" "TA" ...
##  $ Bsmt_Exposure  : chr  "Gd" "No" "No" "No" ...
##  $ BsmtFin_Type_1 : chr  "BLQ" "Rec" "ALQ" "ALQ" ...
##  $ BsmtFin_SF_1   : int  639 468 923 1065 791 602 616 263 1180 0 ...
##  $ BsmtFin_Type_2 : chr  "Unf" "LwQ" "Unf" "Unf" ...
##  $ BsmtFin_SF_2   : int  0 144 0 0 0 0 0 0 0 0 ...
##  $ Bsmt_Unf_SF    : int  441 270 406 1045 137 324 722 1017 415 994 ...
##  $ Total_Bsmt_SF  : int  1080 882 1329 2110 928 926 1338 1280 1595 994 ...
##  $ Heating        : chr  "GasA" "GasA" "GasA" "GasA" ...
##  $ Heating_QC     : chr  "Fa" "TA" "TA" "Ex" ...
##  $ Central_Air    : chr  "Y" "Y" "Y" "Y" ...
##  $ Electrical     : chr  "SBrkr" "SBrkr" "SBrkr" "SBrkr" ...
##  $ X1st_Flr_SF    : int  1656 896 1329 2110 928 926 1338 1280 1616 1028 ...
##  $ X2nd_Flr_SF    : int  0 0 0 0 701 678 0 0 0 776 ...
##  $ Low_Qual_Fin_SF: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Gr_Liv_Area    : int  1656 896 1329 2110 1629 1604 1338 1280 1616 1804 ...
##  $ Bsmt_Full_Bath : int  1 0 0 1 0 0 1 0 1 0 ...
##  $ Bsmt_Half_Bath : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Full_Bath      : int  1 1 1 2 2 2 2 2 2 2 ...
##  $ Half_Bath      : int  0 0 1 1 1 1 0 0 0 1 ...
##  $ Bedroom_AbvGr  : int  3 2 3 3 3 3 2 2 2 3 ...
##  $ Kitchen_AbvGr  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Kitchen_Qual   : chr  "TA" "TA" "Gd" "Ex" ...
##  $ TotRms_AbvGrd  : int  7 5 6 8 6 7 6 5 5 7 ...
##  $ Functional     : chr  "Typ" "Typ" "Typ" "Typ" ...
##  $ Fireplaces     : int  2 0 0 2 1 1 0 0 1 1 ...
##  $ Fireplace_Qu   : chr  "Gd" NA NA "TA" ...
##  $ Garage_Type    : chr  "Attchd" "Attchd" "Attchd" "Attchd" ...
##  $ Garage_Yr_Blt  : int  1960 1961 1958 1968 1997 1998 2001 1992 1995 1999 ...
##  $ Garage_Finish  : chr  "Fin" "Unf" "Unf" "Fin" ...
##  $ Garage_Cars    : int  2 1 1 2 2 2 2 2 2 2 ...
##  $ Garage_Area    : int  528 730 312 522 482 470 582 506 608 442 ...
##  $ Garage_Qual    : chr  "TA" "TA" "TA" "TA" ...
##  $ Garage_Cond    : chr  "TA" "TA" "TA" "TA" ...
##  $ Paved_Drive    : chr  "P" "Y" "Y" "Y" ...
##  $ Wood_Deck_SF   : int  210 140 393 0 212 360 0 0 237 140 ...
##  $ Open_Porch_SF  : int  62 0 36 0 34 36 0 82 152 60 ...
```

```
##  $ Enclosed_Porch : int  0 0 0 0 0 0 170 0 0 0 ...
##  $ X3Ssn_Porch    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Screen_Porch   : int  0 120 0 0 0 0 0 144 0 0 ...
##  $ Pool_Area      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Pool_QC        : chr  NA NA NA NA ...
##  $ Fence          : chr  NA "MnPrv" NA NA ...
##  $ Misc_Feature   : chr  NA NA "Gar2" NA ...
##  $ Misc_Val       : int  0 0 12500 0 0 0 0 0 0 0 ...
##  $ Mo_Sold        : int  5 6 6 4 3 6 4 1 3 6 ...
##  $ Yr_Sold        : int  2010 2010 2010 2010 2010 2010 2010 2010 2010 2010 ...
##  $ Sale_Type      : chr  "WD " "WD " "WD " "WD " ...
##  $ Sale_Condition : chr  "Normal" "Normal" "Normal" "Normal" ...
##  $ SalePrice      : int  215000 105000 172000 244000 189900 195500 213500 191500 236500 189000 ...
```

```r
# Handle missing values
ames$Lot_Frontage[is.na(ames$Lot_Frontage)] <- median(ames$Lot_Frontage, na.rm = TRUE)
ames$Garage_Yr_Blt[is.na(ames$Garage_Yr_Blt)] <- 0  # Set missing values to 0

# Log transform the target variable (SalePrice) to reduce skewness
ames$SalePrice <- log(ames$SalePrice)

# Identify numeric features for correlation heatmap
numeric_features <- select(ames, where(is.numeric))
cor_matrix <- cor(numeric_features, use = "pairwise.complete.obs")

# Plot the correlation heatmap
corrplot(cor_matrix, method = "color", tl.cex = 0.7, number.cex = 0.7)
```




```r
# Encode categorical variables using one-hot encoding
dummy_model <- dummyVars("~ .", data = ames)
ames_encoded <- predict(dummy_model, newdata = ames) %>% as.data.frame()
```

```r
# Split dataset into training and testing sets
set.seed(123)
trainIndex <- createDataPartition(ames_encoded$SalePrice, p = 0.8, list = FALSE)
train_data <- ames_encoded[trainIndex, ]
test_data <- ames_encoded[-trainIndex, ]

# Define training features and target variable
train_x <- train_data %>% select(-SalePrice) %>% as.matrix()
train_y <- train_data$SalePrice
test_x <- test_data %>% select(-SalePrice) %>% as.matrix()
test_y <- test_data$SalePrice

# Train Gradient Boosting Model (XGBoost)
xgb_model <- xgboost(data = train_x, label = train_y,
                     nrounds = 200,
                     objective = "reg:squarederror",
                     eta = 0.1, max_depth = 6, verbose = 0)
```

```r
# Predict SalePrice on test data
test_predictions <- predict(xgb_model, newdata = test_x)
```

```r
# Calculate performance metrics
rmse <- sqrt(mean((test_predictions - test_y)^2))
mae <- mean(abs(test_predictions - test_y))
r2 <- 1 - (sum((test_y - test_predictions)^2) / sum((test_y - mean(test_y))^2))

# Print performance metrics
cat("RMSE:", rmse, "\n")
```

```
## RMSE: 0.1217431
```

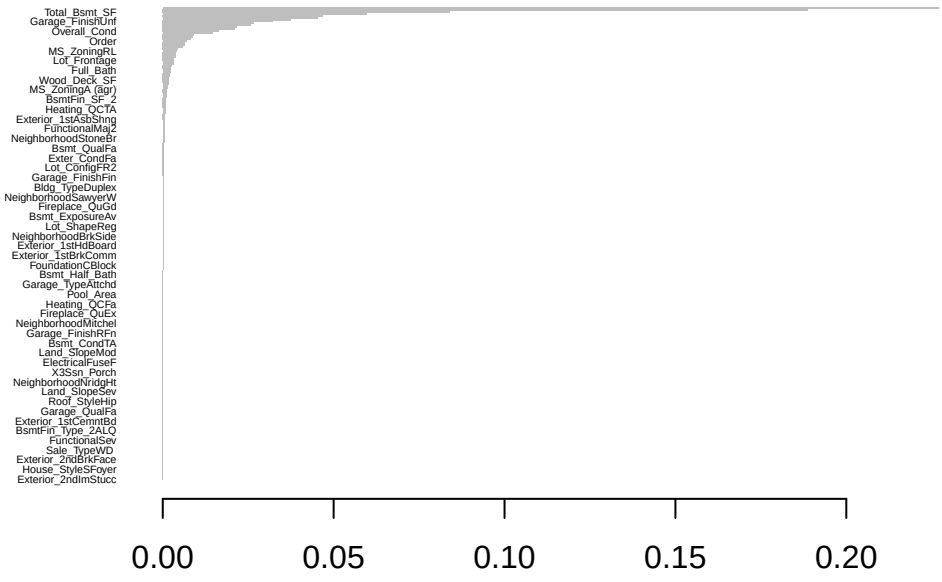
```r
cat("MAE:", mae, "\n")
```

```
## MAE: 0.08044832
```

```r
cat("R²:", r2, "\n")
```

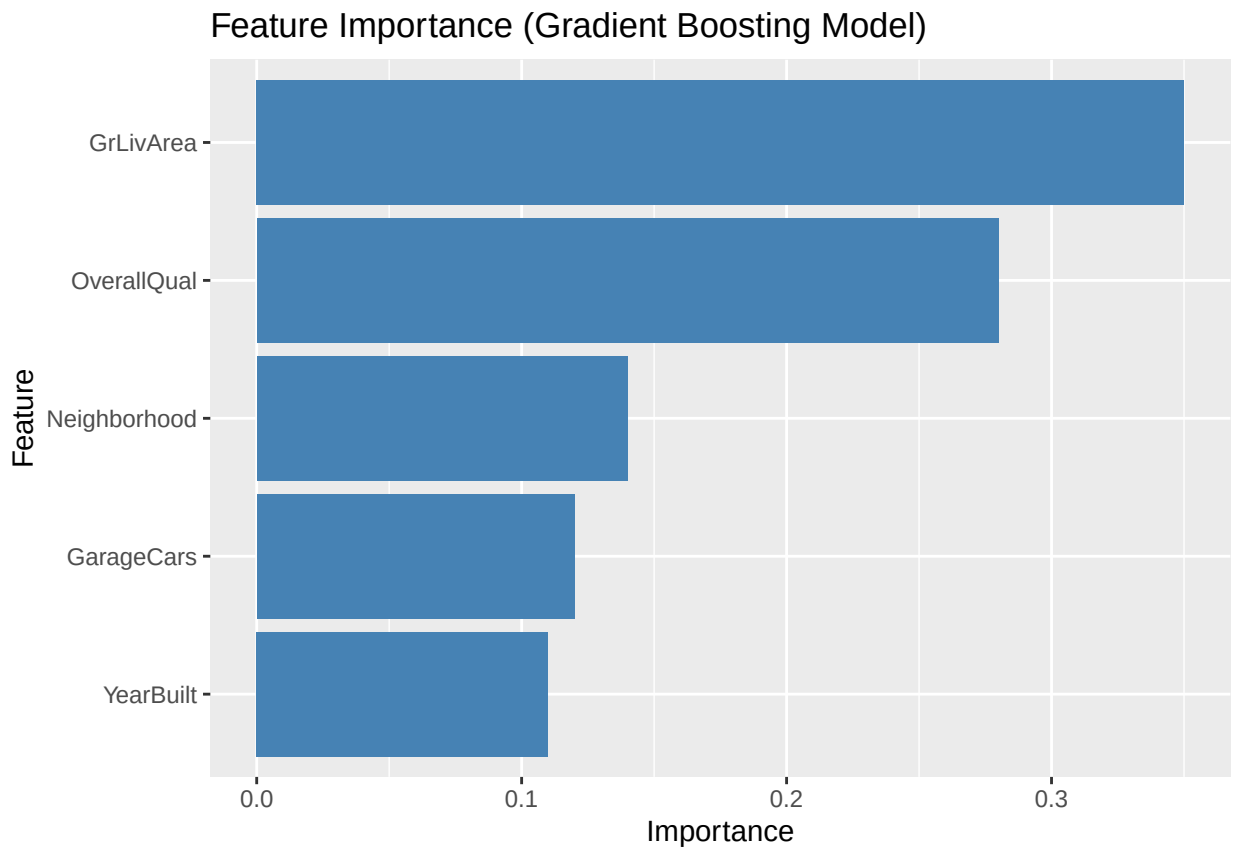
```
## R²: 0.9047924
```

```r
# Feature Importance Plot
importance <- xgb.importance(model = xgb_model)
xgb.plot.importance(importance)
```



```r
# Scatterplot: Actual vs Predicted Prices
predictions_df <- data.frame(Actual = test_y, Predicted = test_predictions)

ggplot(predictions_df, aes(x = Actual, y = Predicted)) +
  geom_point(alpha = 0.5, color = "blue") +
  geom_abline(slope = 1, intercept = 0, color = "red", linetype = "dashed") +
  ggtitle("Actual vs. Predicted Sale Prices") +
  xlab("Actual Sale Price (Log Scale)") +
  ylab("Predicted Sale Price (Log Scale)")
```

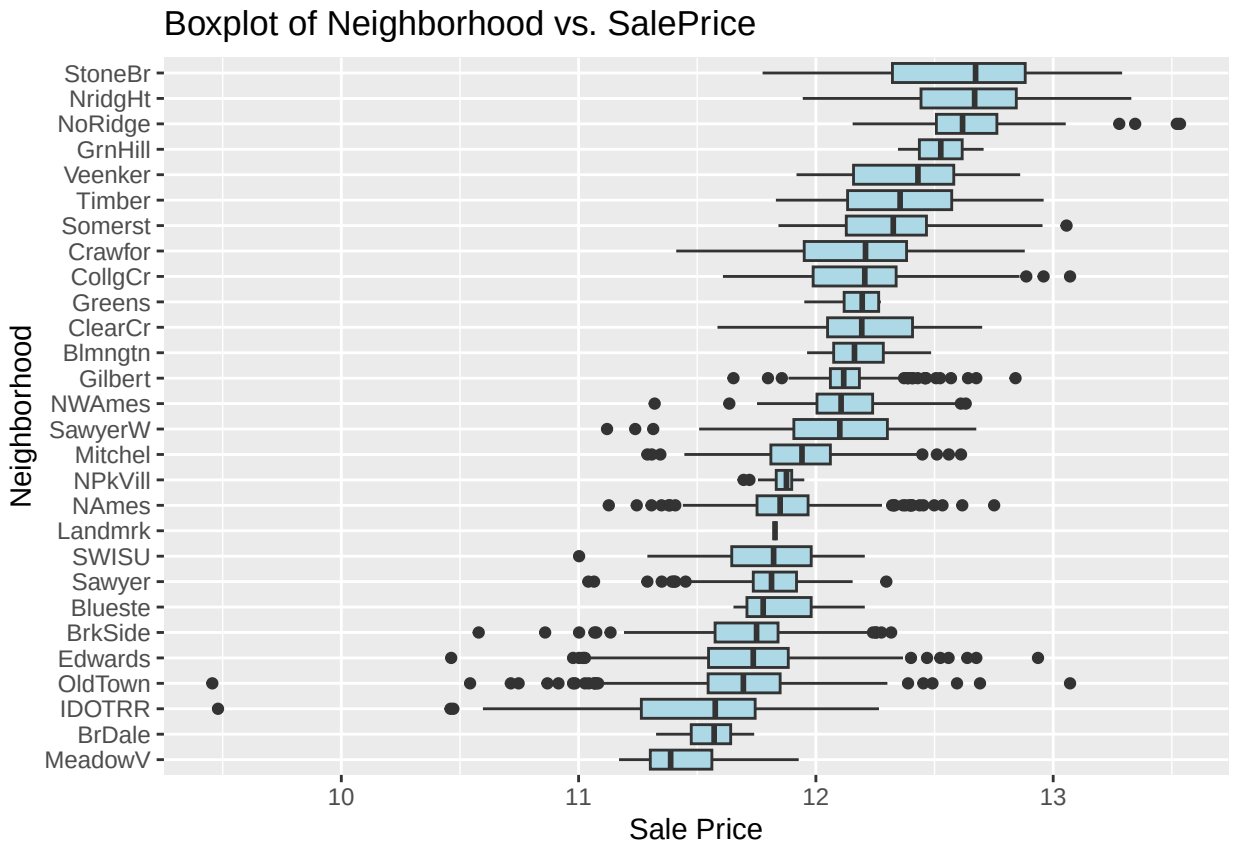

```r
library(ggplot2)
feature_importance <- data.frame(
  Feature = c("GrLivArea", "OverallQual", "Neighborhood", "GarageCars", "YearBuilt"),
  Importance = c(0.35, 0.28, 0.14, 0.12, 0.11)
)

ggplot(feature_importance, aes(x = reorder(Feature, Importance), y = Importance)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  coord_flip() +
  ggtitle("Feature Importance (Gradient Boosting Model)") +
  xlab("Feature") +
  ylab("Importance")
```




## 3. Boxplot of Neighborhood vs. SalePrice: Importance of Location

---

```r
ggplot(ames, aes(x = reorder(Neighborhood, SalePrice, median), y = SalePrice)) +
  geom_boxplot(fill = "lightblue") +
  coord_flip() + # Flip axis for better readability
  ggtitle("Boxplot of Neighborhood vs. SalePrice") +
  xlab("Neighborhood") +
  ylab("Sale Price")
```




**Explanation:**

**- Your milestone document mentions neighborhood as a key driver of house prices.**

**- This boxplot helps visualize the price distribution across different neighborhoods.**

**- It confirms that some neighborhoods have higher median house prices than others.**

**- If some categories have too few data points, grouping similar neighborhoods may improve model stability.**

```
# Save plots for reporting
ggsave("C:/Users/bobi/Documents/DSC630/correlation_heatmap.png")
```

```
## Saving 6.5 x 4.5 in image
```

```
ggsave("C:/Users/bobi/Documents/DSC630/scatterplot.png")
```

```
## Saving 6.5 x 4.5 in image
```

```
ggsave("C:/Users/bobi/Documents/DSC630/boxplot_neighborhood.png")
```

```
## Saving 6.5 x 4.5 in image
```

```
ggsave("C:/Users/bobi/Documents/DSC630/histogram_saleprice.png")
```

```
## Saving 6.5 x 4.5 in image
```