DATA SCIENCE PROGRAMMING
LAB (L3+L4)

ASSESSMENT 1

LINEAR REGRESSION IN R

Name: SWATHI D

Reg. no.: 22MID0035

AIM:

To develop a linear regression model in R to predict a continuous variable (e.g., house rent) based on one or more independent variables.

Procedure

1. **Problem Definition:**

   Objective:
   Predict **house sale price (SalePrice)** using property features.

   Target Variable:
   SalePrice (numeric)

   Features:
   - LotArea
   - MSSubClass
   - OverallCond
   - YearBuilt
   - YearRemodAdd
   - BldgType
   - LotConfig
   - Exterior1st
   - TotalBsmtSF
   - BsmtFinSF2

   Goal:
   Build a linear regression model to estimate house prices based on these characteristics.

2. Import Libraries and Dataset

```r
# Load dataset
data <- read.csv("C:/Users/HP/Downloads/hpp.csv")
```

```
> # Load dataset
> data <- read.csv("C:/Users/HP/Downloads/hpp.csv")
```

```r
#First 10 entries of the dataset
head(data)
```

```
> head(data)
  Id MSSubClass MSZoning LotArea LotConfig BldgType OverallCond YearBuilt YearRemodAdd Exterior1st
1  0         60       RL    8450    Inside     1Fam           5      2003         2003     VinylSd
2  1         20       RL    9600       FR2     1Fam           8      1976         1976     MetalSd
3  2         60       RL   11250    Inside     1Fam           5      2001         2002     VinylSd
4  3         70       RL    9550    Corner     1Fam           5      1915         1970     Wd Sdng
5  4         60       RL   14260       FR2     1Fam           5      2000         2000     VinylSd
6  5         50       RL   14115    Inside     1Fam           5      1993         1995     VinylSd
  BsmtFinSF2 TotalBsmtSF SalePrice
1          0         856    208500
2          0        1262    181500
3          0         920    223500
4          0         756    140000
5          0        1145    250000
6          0         796    143000
```

3. Exploratory Data Analysis (EDA)

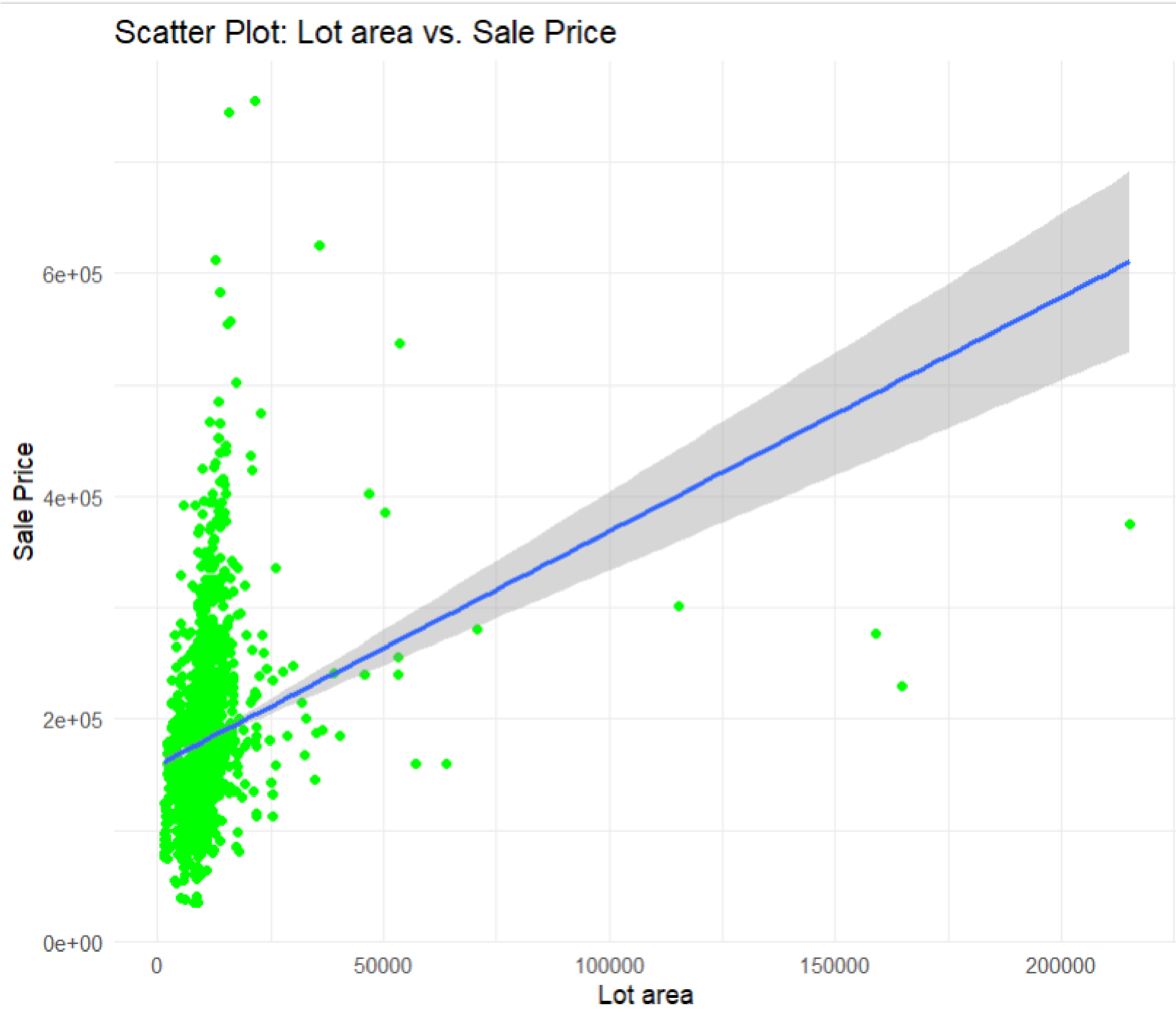
```
# View dimension and summary
dim(data)
summary(data)
```

```
> # View dimension and summary
> dim(data)
[1] 2919   13
```

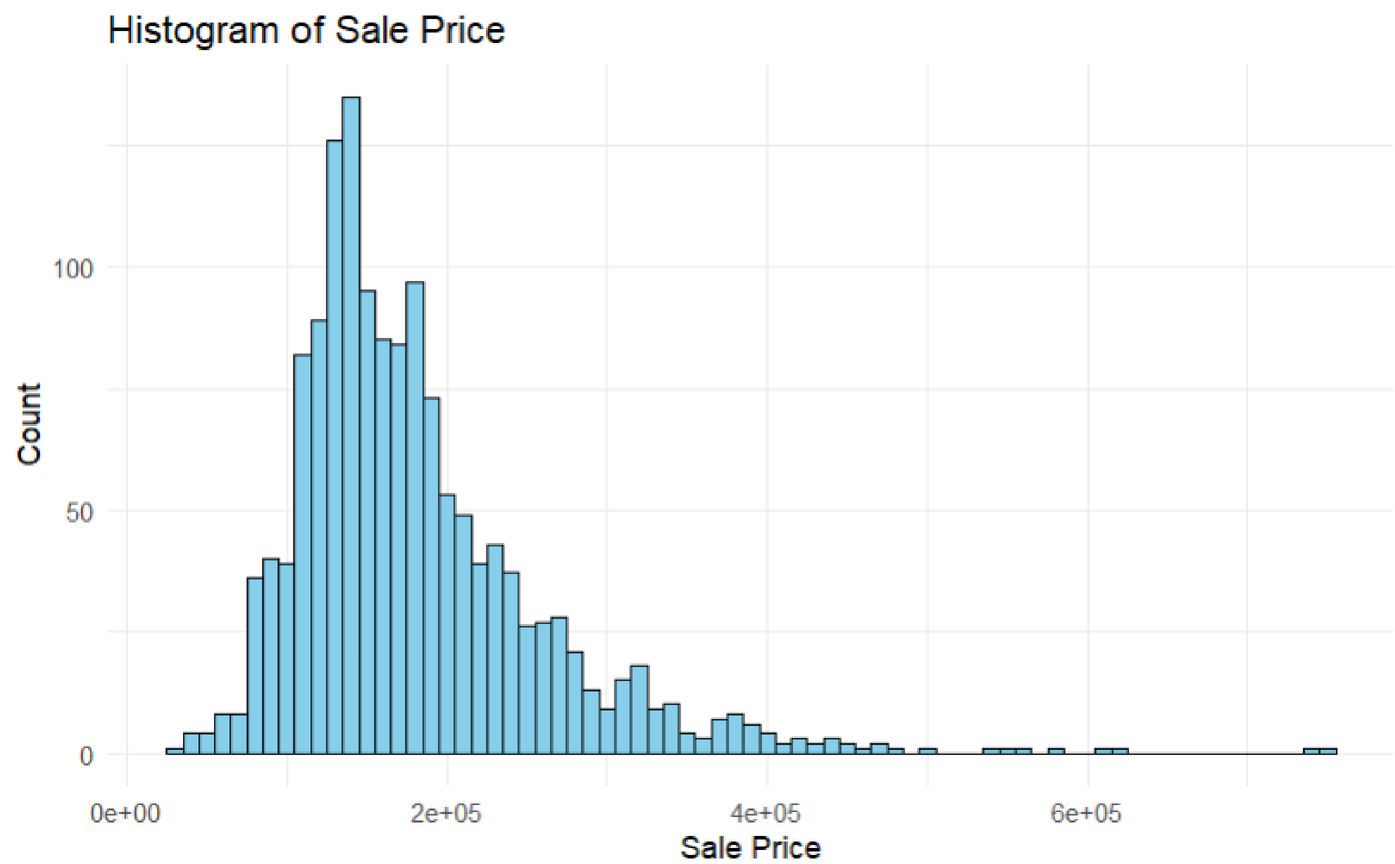
```
> summary(data)
       Id            MSSubClass        MSZoning             LotArea         LotConfig        
 Min.   :   0.0   Min.   : 20.00   Length:2919        Min.   :  1300   Length:2919       
 1st Qu.: 729.5   1st Qu.: 20.00   Class :character   1st Qu.:  7478   Class :character  
 Median :1459.0   Median : 50.00   Mode  :character   Median :  9453   Mode  :character  
 Mean   :1459.0   Mean   : 57.14                      Mean   : 10168                     
 3rd Qu.:2188.5   3rd Qu.: 70.00                      3rd Qu.: 11570                     
 Max.   :2918.0   Max.   :190.00                      Max.   :215245                     
```

```
   BldgType          OverallCond      YearBuilt     YearRemodAdd  Exterior1st       
 Length:2919        Min.   :1.000   Min.   :1872   Min.   :1950   Length:2919       
 Class :character   1st Qu.:5.000   1st Qu.:1954   1st Qu.:1965   Class :character  
 Mode  :character   Median :5.000   Median :1973   Median :1993   Mode  :character  
                    Mean   :5.565   Mean   :1971   Mean   :1984                     
                    3rd Qu.:6.000   3rd Qu.:2001   3rd Qu.:2004                     
                    Max.   :9.000   Max.   :2010   Max.   :2010                     

   BsmtFinSF2        TotalBsmtSF       SalePrice     
 Min.   :   0.00   Min.   :   0.0   Min.   : 34900  
 1st Qu.:   0.00   1st Qu.: 793.0   1st Qu.:129975  
 Median :   0.00   Median : 989.5   Median :163000  
 Mean   :  49.58   Mean   :1051.8   Mean   :180921  
 3rd Qu.:   0.00   3rd Qu.:1302.0   3rd Qu.:214000  
 Max.   :1526.00   Max.   :6110.0   Max.   :755000  
 NA's   :1         NA's   :1        NA's   :1459    
```

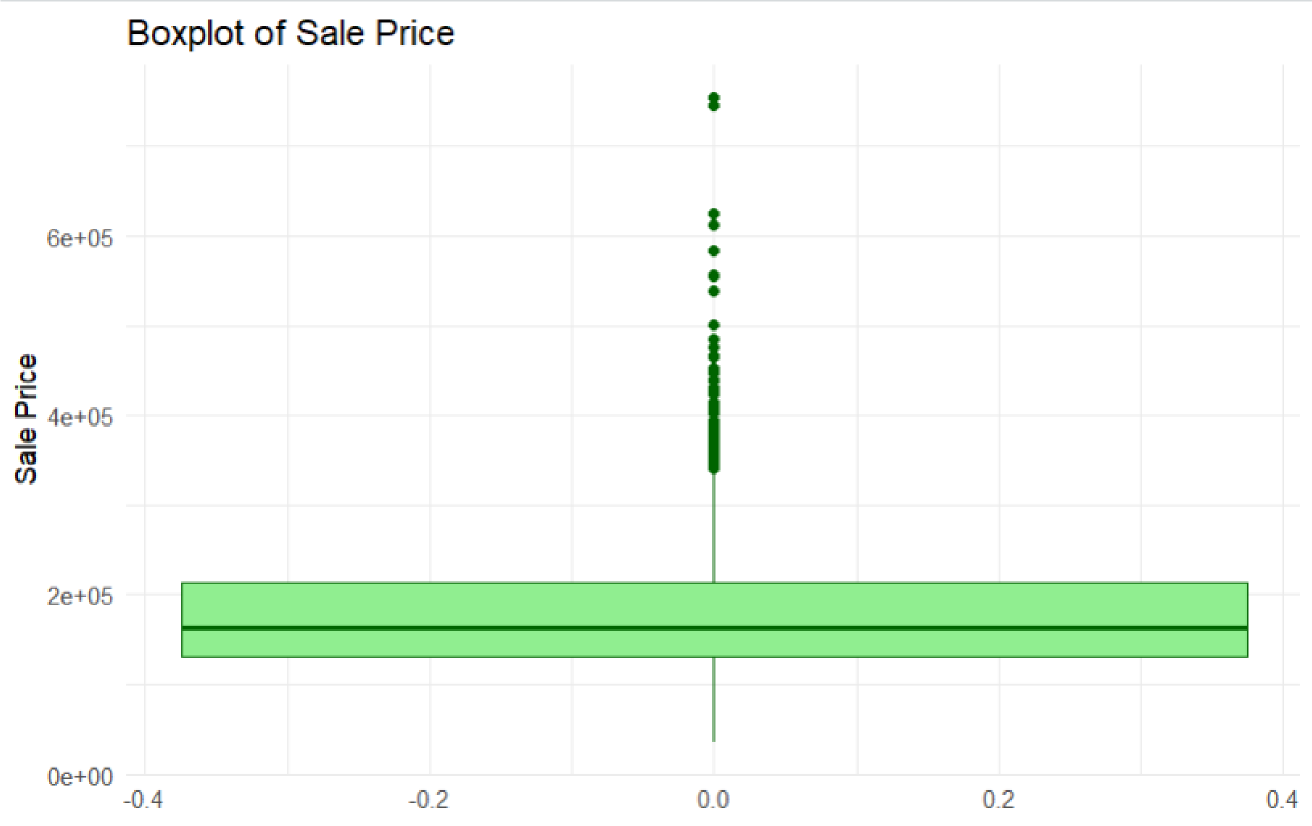
```r
#Scatter plot
ggplot(data, aes(x = LotArea, y = SalePrice)) +
  geom_point(color = "red", alpha = 0.6) +
  labs(title = "Scatter Plot: Lot Area vs Sale Price",
       x = "Lot Area (sq ft)",
       y = "Sale Price") +
  theme_minimal()
```



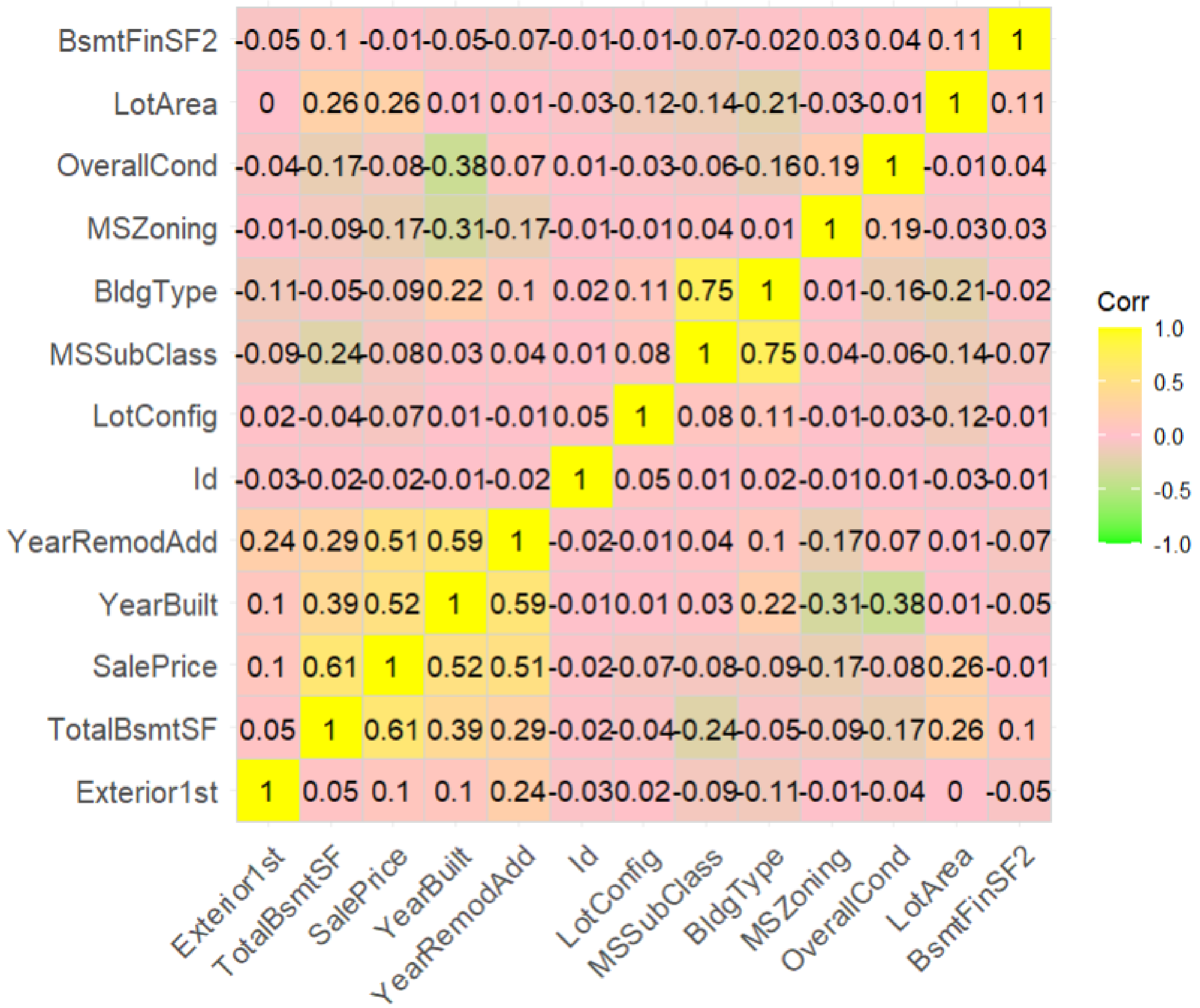
```
#Histogram for sale price
library(ggplot2)

ggplot(data, aes(x = SalePrice)) +
  geom_histogram(binwidth = 10000, fill = "skyblue", color = "black") +
  labs(title = "Histogram of Sale Price", x = "Sale Price", y = "Count") +
  theme_minimal()
```



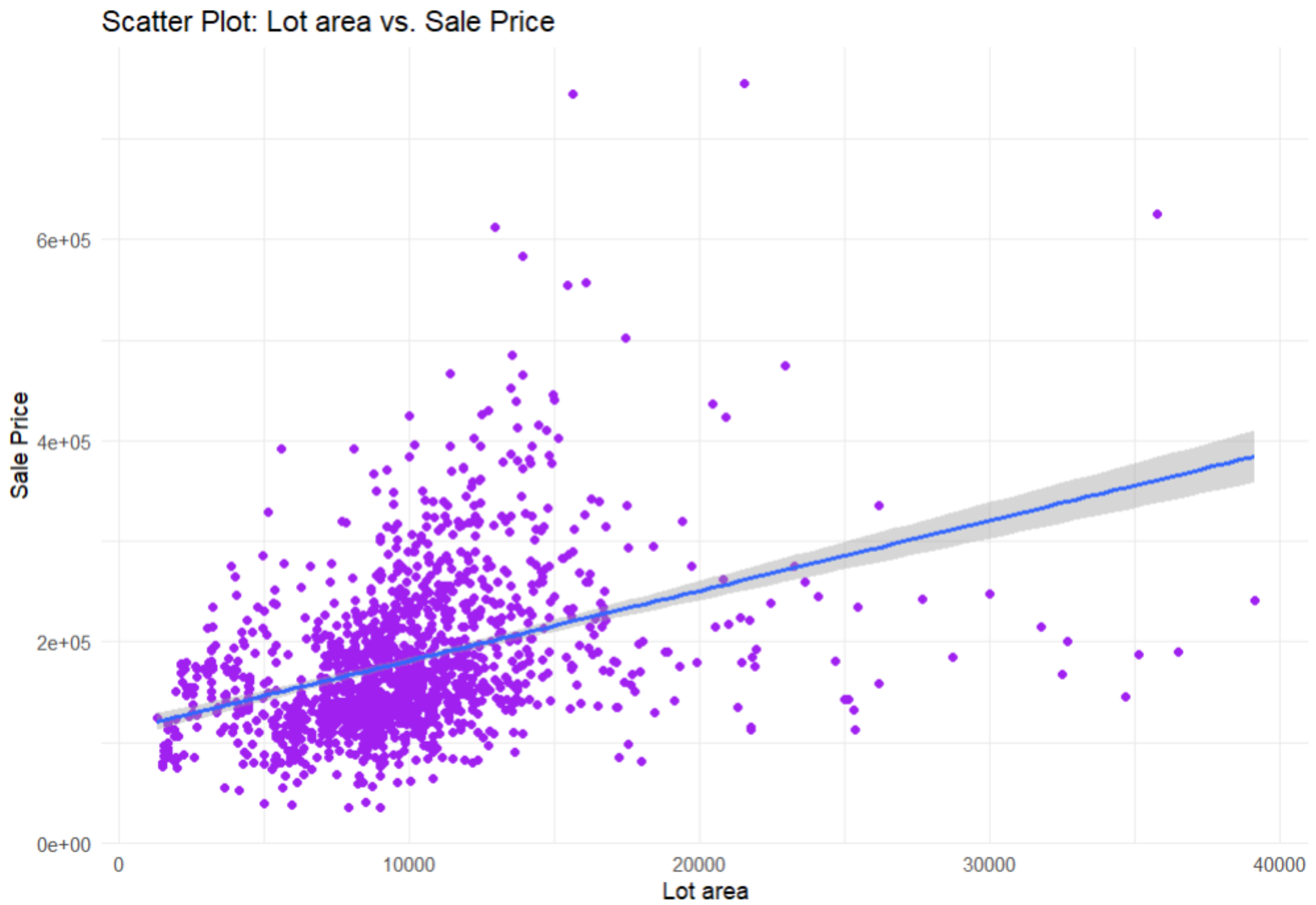
```r
#Box plot for sales price
ggplot(data, aes(y = SalePrice)) +
  geom_boxplot(fill = "lightgreen", color = "darkgreen") +
  labs(title = "Boxplot of Sale Price", y = "Sale Price") +
  theme_minimal()
```



```r
library(ggplot2)
library(ggcorrplot)

# Calculate the correlation matrix
correlation_matrix <- cor(data)

# Create the correlation plot with custom colors
ggcorrplot(
  correlation_matrix,
  hc.order = TRUE,
  lab = TRUE,
  colors = c("green", "pink", "yellow"),
  outline.col = "lightgray",
  ggtheme = ggplot2::theme_minimal()
)
```



```
# Removing rows with Lot Area < 40000
data <- data[data$LotArea < 40000, ]

# Create a scatter plot between "square_feet" and "price" with aesthetic colors
ggplot(data, aes(x = LotArea, y = SalePrice)) +
  geom_point(color = "purple") +
  labs(title = "Scatter Plot: Lot area vs. Sale Price", x = "Lot area", y = "Sale Price") +
  theme_minimal()+
  geom_smooth(method = lm)
```



4. Data Preprocessing:

```
# Check missing values
colSums(is.na(data))
```

```
> # Check missing values
> colSums(is.na(data))
         Id  MSSubClass    MSZoning     LotArea   LotConfig    BldgType OverallCond 
          0           0           0           0           0           0           0 
  YearBuilt YearRemodAdd Exterior1st  BsmtFinSF2 TotalBsmtSF   SalePrice 
          0           0           0           1           1        1459 
```

```
# Remove missing values
data<-na.omit(data)
sum(is.na(data))
```

```
> # Remove missing values
> data<-na.omit(data)
> sum(is.na(data))
[1] 0
```

```
# Performing label encoding
data$MSZoning <- as.numeric(factor(data$MSZoning))
data$LotConfig <- as.numeric(factor(data$LotConfig))
data$BldgType <- as.numeric(factor(data$BldgType))
data$Exterior1st <- as.numeric(factor(data$Exterior1st))
```

```
> # Performing label encoding
> data$MSZoning <- as.numeric(factor(data$MSZoning))
> data$LotConfig <- as.numeric(factor(data$LotConfig))
> data$BldgType <- as.numeric(factor(data$BldgType))
> data$Exterior1st <- as.numeric(factor(data$Exterior1st))
>
```

5. **Split the Dataset**

```
# Splitting data into train and test
set.seed(123)

# Define the proportion for the training set (e.g., 70% for training, 30% for testing)
train_proportion <- 0.7

# Generate a random sample of row indices for the training set
train_indices <- sample(1:nrow(data),
                        size = round(train_proportion * nrow(data)))

# Create the training and testing datasets
train_data <- data[train_indices, ]
test_data <- data[-train_indices, ]

dim(train_data)

dim(test_data)
```

```
> # Splitting data into train and test
> set.seed(123)
> # Define the proportion for the training set (e.g., 70% for training, 30% for testing)
> train_proportion <- 0.7
> # Generate a random sample of row indices for the training set
> train_indices <- sample(1:nrow(data),
+                         size = round(train_proportion * nrow(data)))
> # Create the training and testing datasets
> train_data <- data[train_indices, ]
> test_data <- data[-train_indices, ]
> dim(train_data)
[1] 1012   13
> dim(test_data)
[1] 434  13
```

6. Build the Linear Regression Model

```
# Train a Multiple Linear Regression model on the train data
model <- lm(SalePrice ~ ., data = train_data)

# Printing summary of the model
summary(model)
```

```
> # Train a Multiple Linear Regression model on the train data
> model <- lm(SalePrice ~ ., data = train_data)
> # Printing summary of the model
> summary(model)

Call:
lm(formula = SalePrice ~ ., data = train_data)

Residuals:
    Min      1Q  Median      3Q     Max 
-161895  -25558   -5578   20650  271209 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -3.048e+06  1.561e+05 -19.530  < 2e-16 ***
Id            1.877e+00  3.405e+00   0.551  0.58167    
MSSubClass    5.718e+02  5.608e+01  10.196  < 2e-16 ***
MSZoning     -1.385e+03  2.325e+03  -0.596  0.55141    
LotArea       4.061e+00  3.878e-01  10.474  < 2e-16 ***
LotConfig     4.642e+02  9.061e+02   0.512  0.60851    
BldgType     -1.811e+04  2.092e+03  -8.656  < 2e-16 ***
OverallCond   5.904e+03  1.500e+03   3.936 8.86e-05 ***
YearBuilt     7.667e+02  7.375e+01  10.396  < 2e-16 ***
YearRemodAdd  7.839e+02  9.858e+01   7.952 4.95e-15 ***
Exterior1st  -8.116e+02  4.669e+02  -1.738  0.08246 .  
BsmtFinSF2   -2.340e+01  8.665e+00  -2.701  0.00703 ** 
TotalBsmtSF   8.976e+01  4.264e+00  21.053  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 45290 on 999 degrees of freedom
Multiple R-squared:  0.6674,	Adjusted R-squared:  0.6634 
F-statistic: 167.1 on 12 and 999 DF,  p-value: < 2.2e-16
```

```r
#Model Prediction
predictions <- predict(model, newdata = test_data)

head(predictions)
```

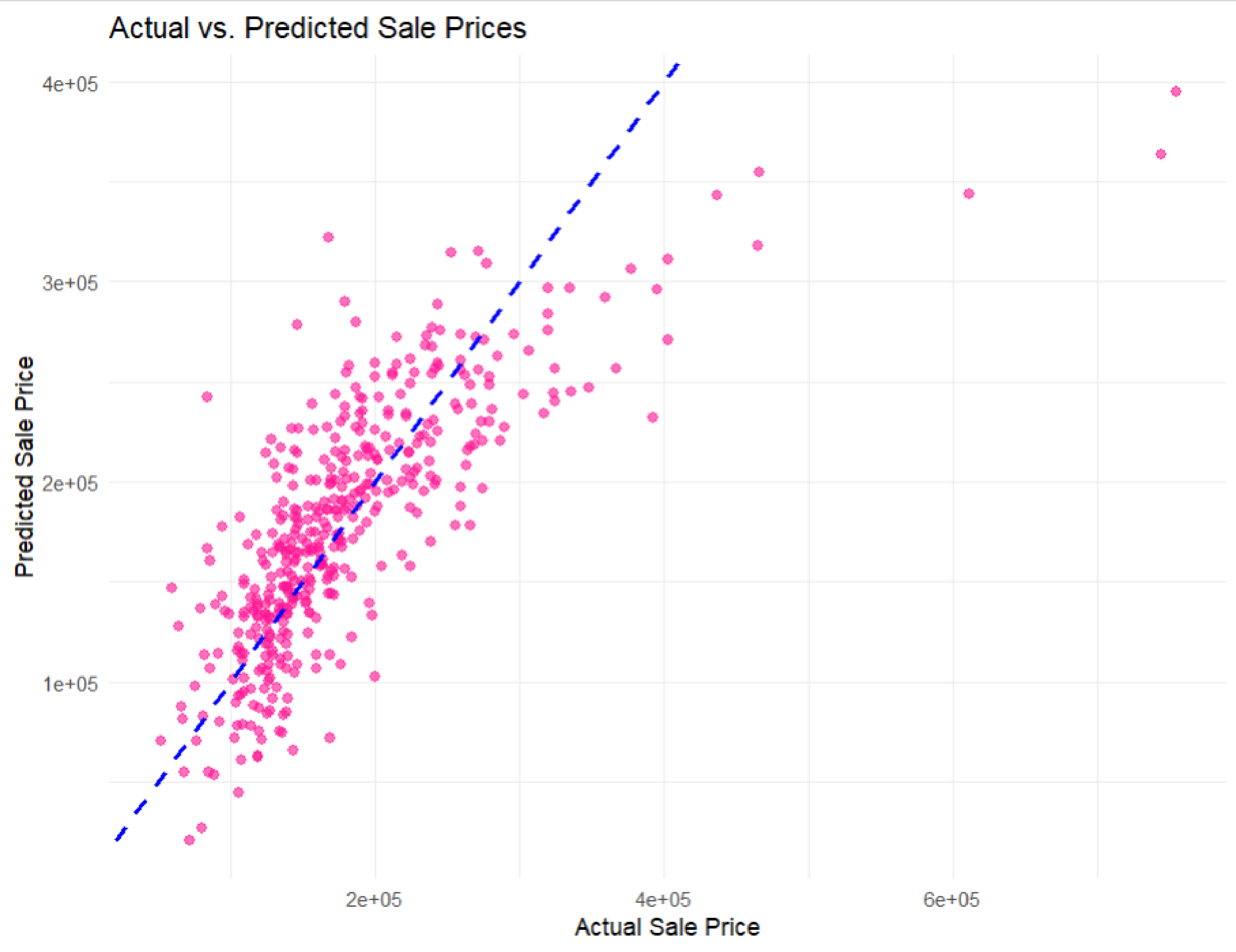
```
> #Model Prediction
> predictions <- predict(model, newdata = test_data)
> head(predictions)
       1        3        7       14       15       21 
200771.2 215574.6 261383.2 249570.3 158306.5 253359.3 
>
```

7. **Model Evaluation**

```r
# Sample test case
test_case <- data.frame(MSSubClass = 60,MSZoning = 4,LotArea = 8450,
                        LotConfig = 5, BldgType = 1, OverallCond = 5,
                        YearBuilt = 2003, YearRemodAdd = 2003,
                        Exterior1st = 13, BsmtFinSF2 = 0,
                        TotalBsmtSF = 856
)

# Make predictions using the trained model
predicted_price <- predict(model, newdata = test_case)

# View the predicted price
print(predicted_price)
```

```
> # Sample test case
> train_data <- data[ , !(names(data) %in% "Id")]
> # Train the model
> model <- lm(SalePrice ~ ., data = train_data)
> test_case <- data.frame(MSSubClass = 60,MSZoning = 4,LotArea = 8450,
+                         LotConfig = 5, BldgType = 1, OverallCond = 5,
+                         YearBuilt = 2003, YearRemodAdd = 2003,
+                         Exterior1st = 13, BsmtFinSF2 = 0,
+                         TotalBsmtSF = 856
+ )
> # Make predictions using the trained model
> predicted_price <- predict(model, newdata = test_case)
> # View the predicted price
> print(predicted_price)
       1 
200630.3 
```

```r
library(ggplot2)
results_df <- data.frame(Actual = actual,
                         Predicted = predictions)

# Plot
ggplot(results_df, aes(x = Actual, y = Predicted)) +
  geom_point(color = "deeppink", alpha = 0.6, size = 2) +
  geom_abline(intercept = 0, slope = 1, color = "blue", linetype = "dashed", size = 1) +
  labs(title = "Actual vs. Predicted Sale Prices",
       x = "Actual Sale Price",
       y = "Predicted Sale Price") +
  theme_minimal()
```

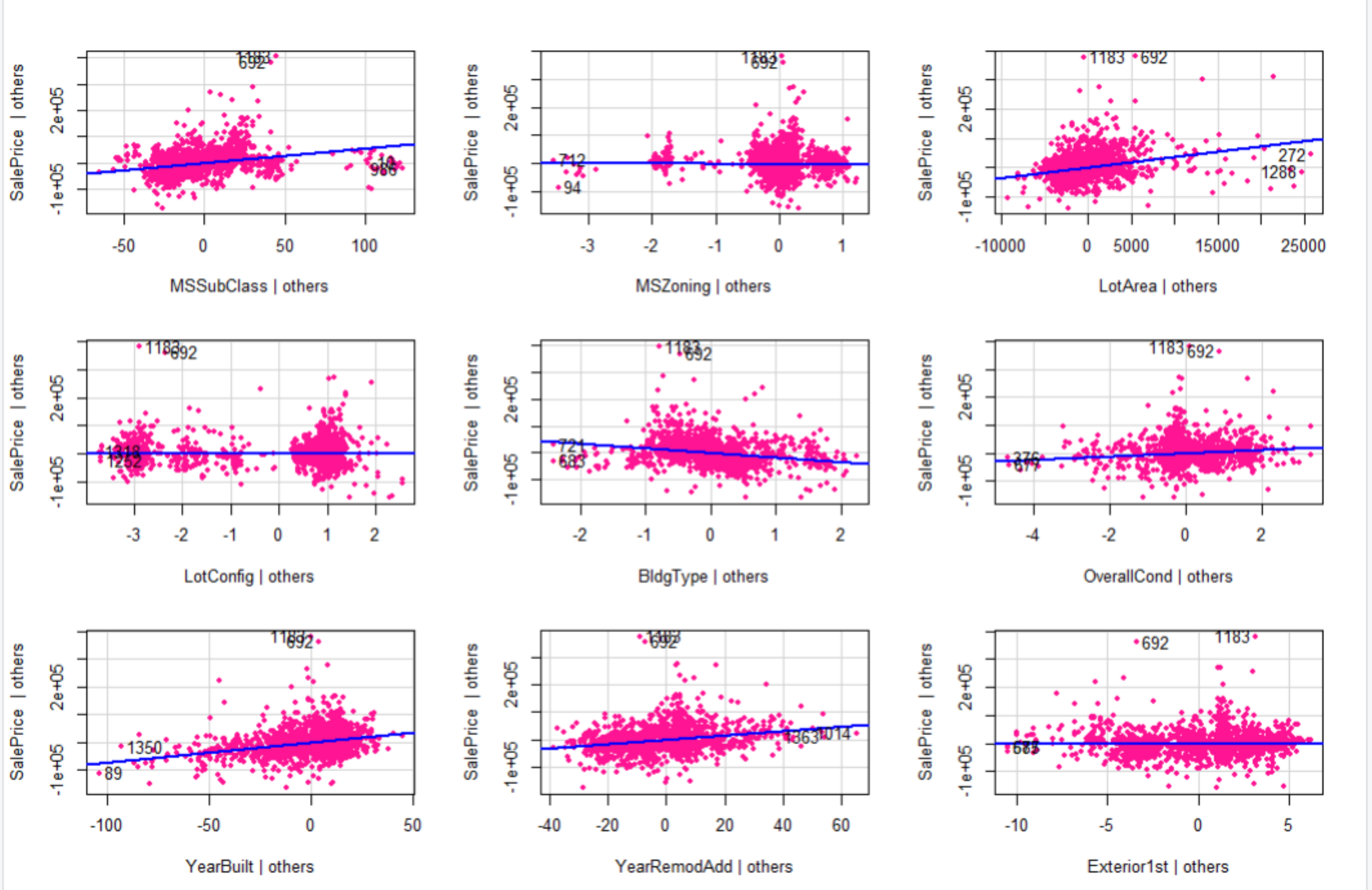


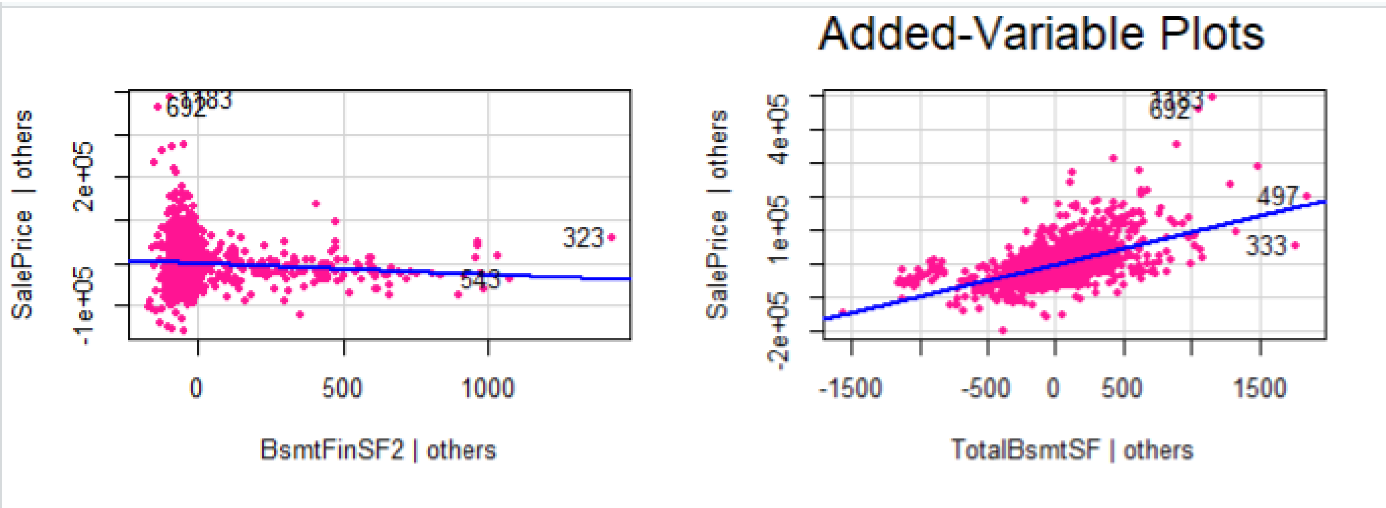
```r
# Predict on test set
predictions <- predict(model, newdata = test_data)

# Actual values
actual <- test_data$SalePrice

# Calculate metrics
MAE <- mean(abs(predictions - actual))
MSE <- mean((predictions - actual)^2)
RMSE <- sqrt(MSE)
R2 <- summary(model)$r.squared

# Print results
cat("Model Evaluation Metrics:\n")
cat("R-squared (R²):", round(R2, 4), "\n")
cat("Mean Absolute Error (MAE):", round(MAE, 2), "\n")
cat("Mean Squared Error (MSE):", round(MSE, 2), "\n")
cat("Root Mean Squared Error (RMSE):", round(RMSE, 2), "\n")
```

```
> # Predict on test set
> predictions <- predict(model, newdata = test_data)
> # Actual values
> actual <- test_data$SalePrice
> # Calculate metrics
> MAE <- mean(abs(predictions - actual))
> MSE <- mean((predictions - actual)^2)
> RMSE <- sqrt(MSE)
> R2 <- summary(model)$r.squared
> # Print results
> cat("Model Evaluation Metrics:\n")
Model Evaluation Metrics:
> cat("R-squared (R²):", round(R2, 4), "\n")
R-squared (R²): 0.6416
> cat("Mean Absolute Error (MAE):", round(MAE, 2), "\n")
Mean Absolute Error (MAE): 34338.82
> cat("Mean Squared Error (MSE):", round(MSE, 2), "\n")
Mean Squared Error (MSE): 2600316481
> cat("Root Mean Squared Error (RMSE):", round(RMSE, 2), "\n")
Root Mean Squared Error (RMSE): 50993.3
```

8. Visualization

```r
library(car)

# Added-variable plots
avPlots(model,
        col = "deeppink",
        col.lines = "blue",
        pch = 19,
        ask = FALSE)
```





9. Conclusion

A linear regression model was built to predict house prices based on property features. The model performed reasonably well, with factors like **LotArea**, **Overall Condition**, and **Total Basement Area** having strong influence. Visualizations and metrics confirmed that the model can make useful price estimates, though further tuning could improve accuracy.