group15

Pamela Kamdefwere

2025-05-08

```
# Load necessary packages
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.3.3
```

```
## Warning: package 'readr' was built under R version 4.3.3
```

```
## Warning: package 'forcats' was built under R version 4.3.3
```

```
## Warning: package 'lubridate' was built under R version 4.3.3
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.0     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.3.3
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 4.3.3
```

```
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
##
## Loaded glmnet 4.1-8
```

```r
# Step 1: Load dataset
data <- read.csv("C:/Users/user/Downloads/house.csv", stringsAsFactors = FALSE)

# Step 2: Clean and convert 'Price' column
data$Price <- as.character(data$Price) # Convert to character
data$Price <- gsub(",", "", data$Price) # Remove commas
data$Price <- gsub("\\$", "", data$Price) # Remove dollar signs (if any)
data$Price <- as.numeric(data$Price) # Convert to numeric
data <- data[!is.na(data$Price), ] # Drop rows with NA prices

# Step 3: Create log-transformed target
data$LogPrice <- log(data$Price)

# Step 4: Keep only numeric predictors
numeric_data <- data %>% select(where(is.numeric))

# Step 5: Remove near-zero variance features
nzv <- nearZeroVar(numeric_data)
if (length(nzv) > 0) {
numeric_data <- numeric_data[, -nzv]
}

# Step 6: Train-test split
set.seed(123)
train_index <- createDataPartition(numeric_data$LogPrice, p = 0.8, list = FALSE)
train <- numeric_data[train_index, ]
test <- numeric_data[-train_index, ]

# Step 7: Fit linear model
lm_model <- lm(LogPrice ~ ., data = train)
summary(lm_model)
```

```
##
## Call:
## lm(formula = LogPrice ~ ., data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.98155 -0.13441  0.06253  0.19874  0.27007
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.167e+01  3.325e-01  35.088   <2e-16 ***
## Id          -1.092e-05  1.054e-05  -1.037    0.300
## Area        -4.341e-06  4.677e-06  -0.928    0.353
## Bedrooms     6.183e-03  4.241e-03   1.458    0.145
## Bathrooms   -2.465e-04  5.457e-03  -0.045    0.964
## Floors       1.553e-03  7.484e-03   0.208    0.836
## YearBuilt    4.938e-06  1.687e-04   0.029    0.977
## Price        2.456e-06  2.193e-08 111.980   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2424 on 1592 degrees of freedom
```

```
## Multiple R-squared:  0.8879, Adjusted R-squared:  0.8874
## F-statistic:  1801 on 7 and 1592 DF,  p-value: < 2.2e-16
```

```
# Step 8: Predict and evaluate
preds <- predict(lm_model, newdata = test)
rmse <- sqrt(mean((preds - test$LogPrice)^2))
mae <- mean(abs(preds - test$LogPrice))
r2 <- cor(preds, test$LogPrice)^2

cat("Model Evaluation:\n")
```

```
## Model Evaluation:
```

```
cat("RMSE:", round(rmse, 3), "\n")
```

```
## RMSE: 0.221
```

```
cat("MAE:", round(mae, 3), "\n")
```

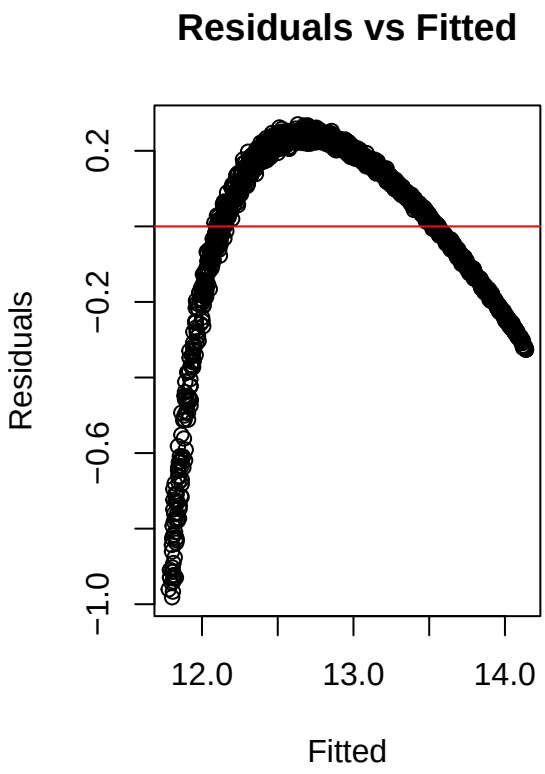
```
## MAE: 0.178
```

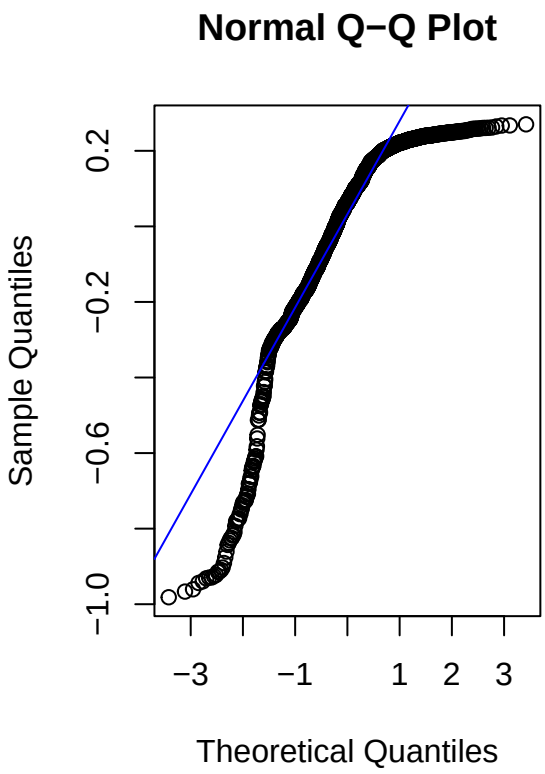
```
cat("R-squared:", round(r2, 4), "\n")
```

```
## R-squared: 0.8971
```

```
# Step 9: Plot residuals
res <- residuals(lm_model)
fitted <- fitted(lm_model)

par(mfrow = c(1, 2))
plot(fitted, res, main = "Residuals vs Fitted", xlab = "Fitted", ylab = "Residuals")
abline(h = 0, col = "red")
qqnorm(res)
qqline(res, col = "blue")
```


Residuals vs Fitted
Residuals
Fitted
−1.0
−0.6
−0.2
0.2
12.0
13.0
14.0


Normal Q−Q Plot
Sample Quantiles
Theoretical Quantiles
−1.0
−0.6
−0.2
0.2
−3
−1
1
2
3