Research Proposal and Data Introduction

A Preliminary Multiple Linear Regression Model

Alyna Qi Heidi Wang John Zhang

December 06, 2024

Contents

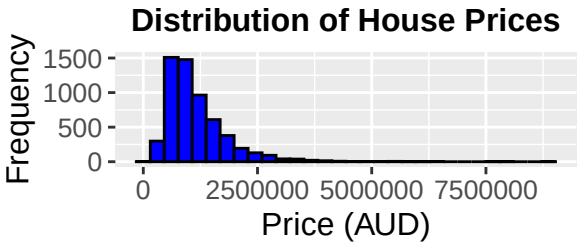
1 Data Description

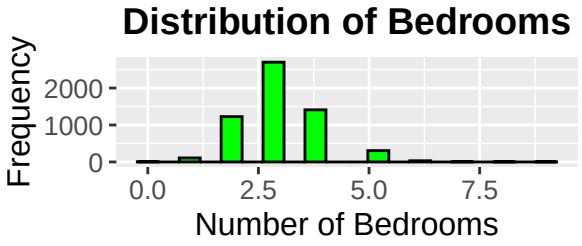
Table 1: Summary Statistics for Numeric Variables

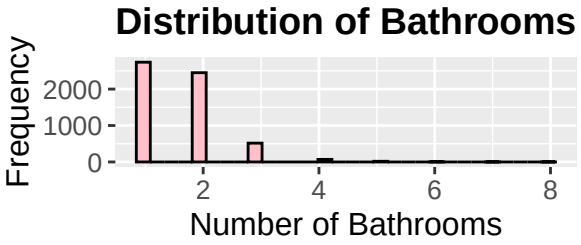
	Value
Price_Min	131000.00
Price_Mean	1156080.72
Price_Median	965000.00
Price_Max	9000000.00
Bedroom2_Min	0.00
Bedroom2_Mean	3.12
Bedroom2_Median	3.00
Bedroom2_Max	9.00
Bathroom_Min	1.00
Bathroom_Mean	1.65
Bathroom_Median	2.00
Bathroom_Max	8.00
Distance_Min	0.70
Distance_Mean	10.80
Distance_Median	10.10
Distance_Max	47.40
Landsize_Min	1.00
Landsize_Mean	572.83
Landsize_Median	509.00
Landsize_Max	37000.00

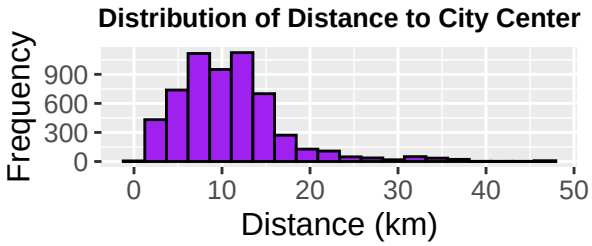
Table 2: Frequency Table for House Type

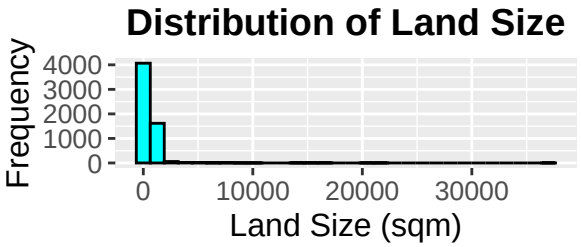
House Type	Frequency
h	4570
t	554
u	673

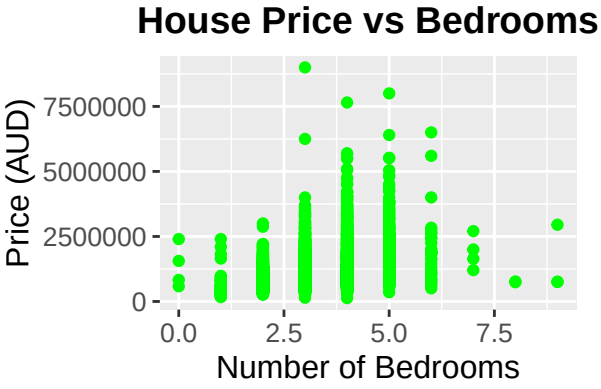

Distribution of House Prices
Frequency
1500
1000
500
0
0
2500000
5000000
7500000
Price (AUD)

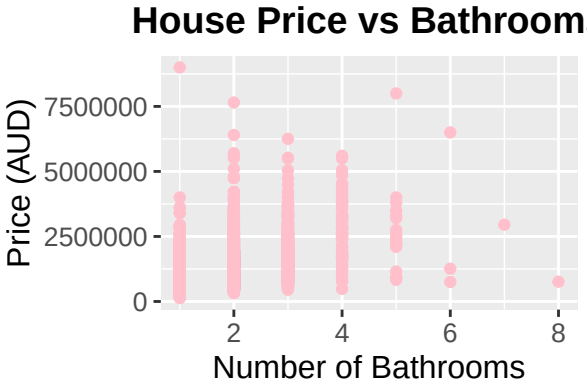

Distribution of Bedrooms
Frequency
2000
1000
0
0.0
2.5
5.0
7.5
Number of Bedrooms

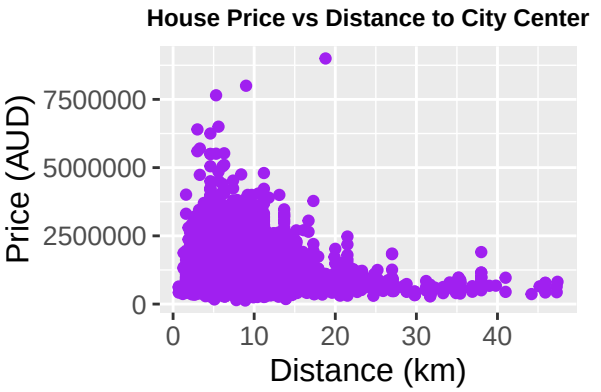

Distribution of Bathrooms
Frequency
2000
1000
0
2
4
6
8
Number of Bathrooms

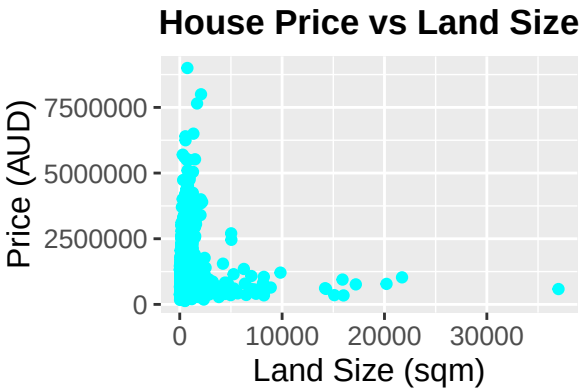

Distribution of Distance to City Center
Frequency
900
600
300
0
0
10
20
30
40
50
Distance (km)


Distribution of Land Size
Frequency
4000
3000
2000
1000
0
0
10000
20000
30000
Land Size (sqm)


House Price vs Bedrooms
Price (AUD)
7500000
5000000
2500000
0
0.0
2.5
5.0
7.5
Number of Bedrooms


House Price vs Bathrooms
Price (AUD)
7500000
5000000
2500000
0
2
4
6
8
Number of Bathrooms


House Price vs Distance to City Center
Price (AUD)
7500000
5000000
2500000
0
0
10
20
30
40
Distance (km)

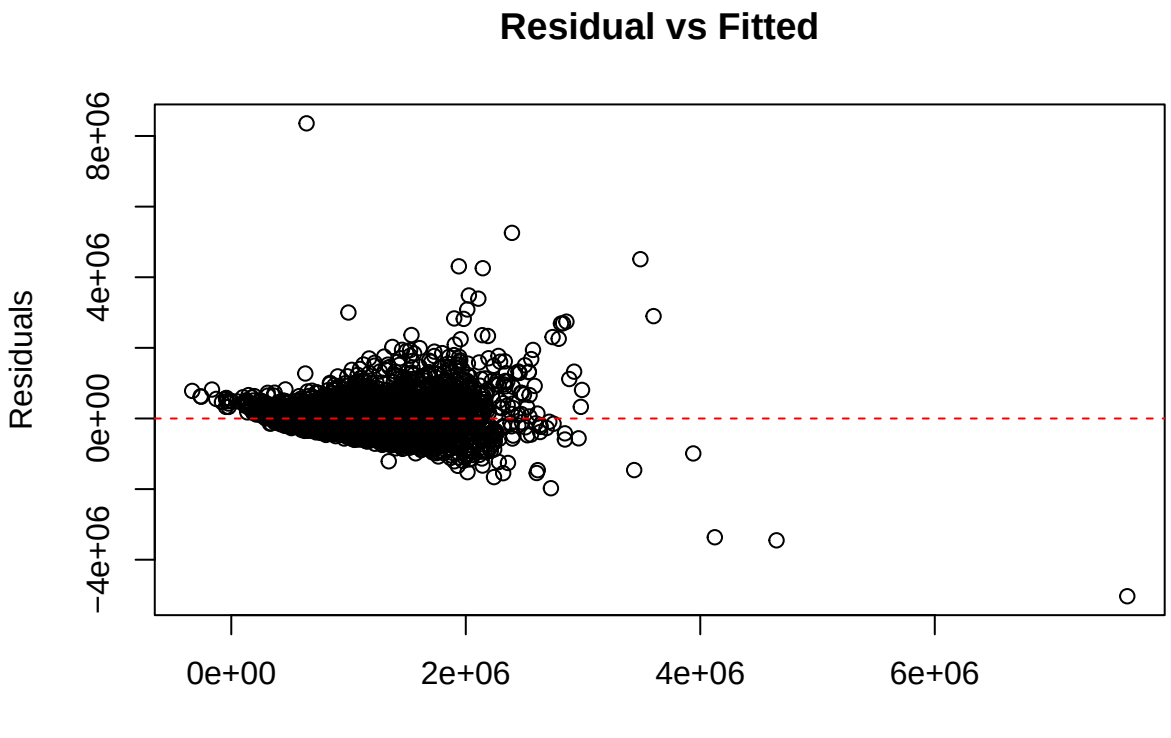

House Price vs Land Size
Price (AUD)
7500000
5000000
2500000
0
0
10000
20000
30000
Land Size (sqm)

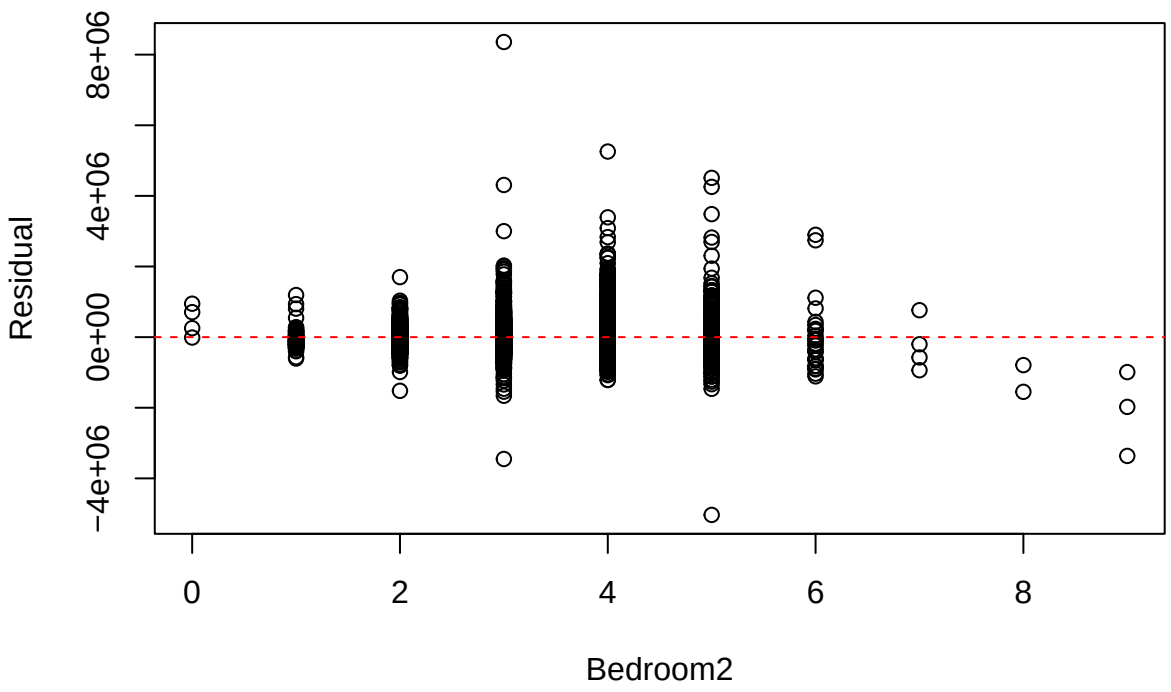
2 Train and Test Split

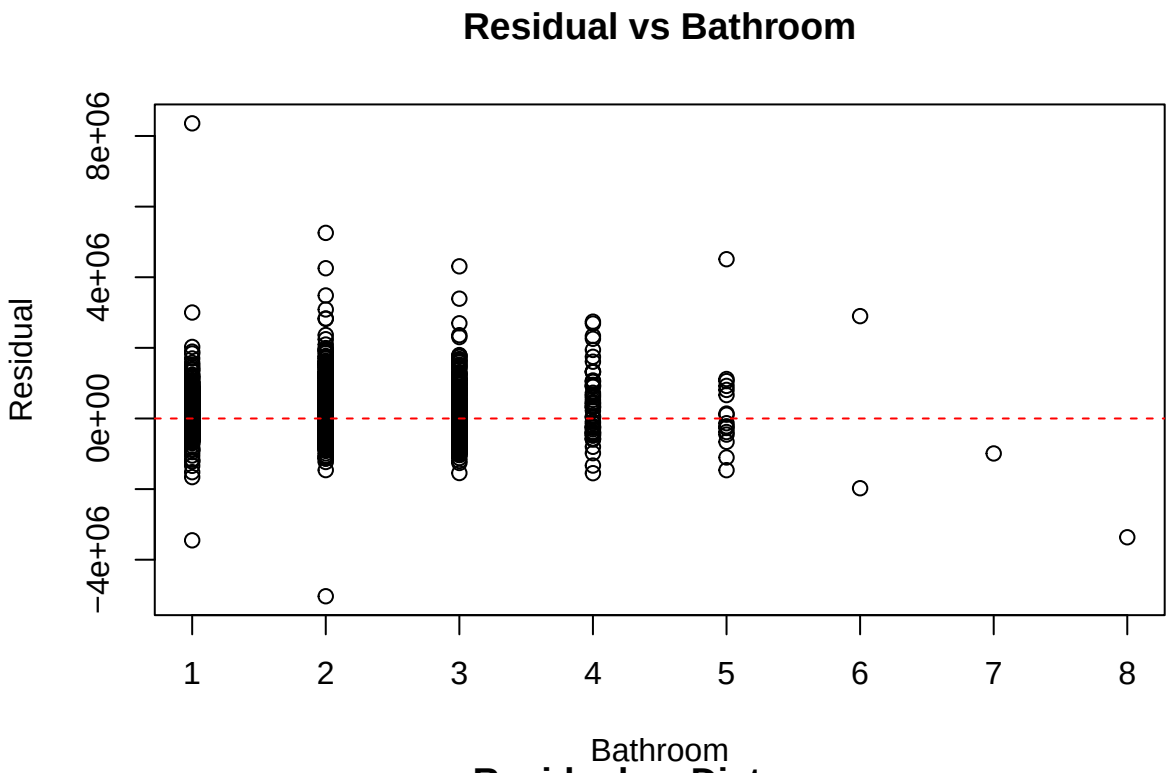
Housing Price = β_0+β_1·Type+β_2·Method+β_3·Distance+β_4·Bedroom+β_5·Bathroom+β_6·Landsize+β_7·Car+β_8·BuildingArea-

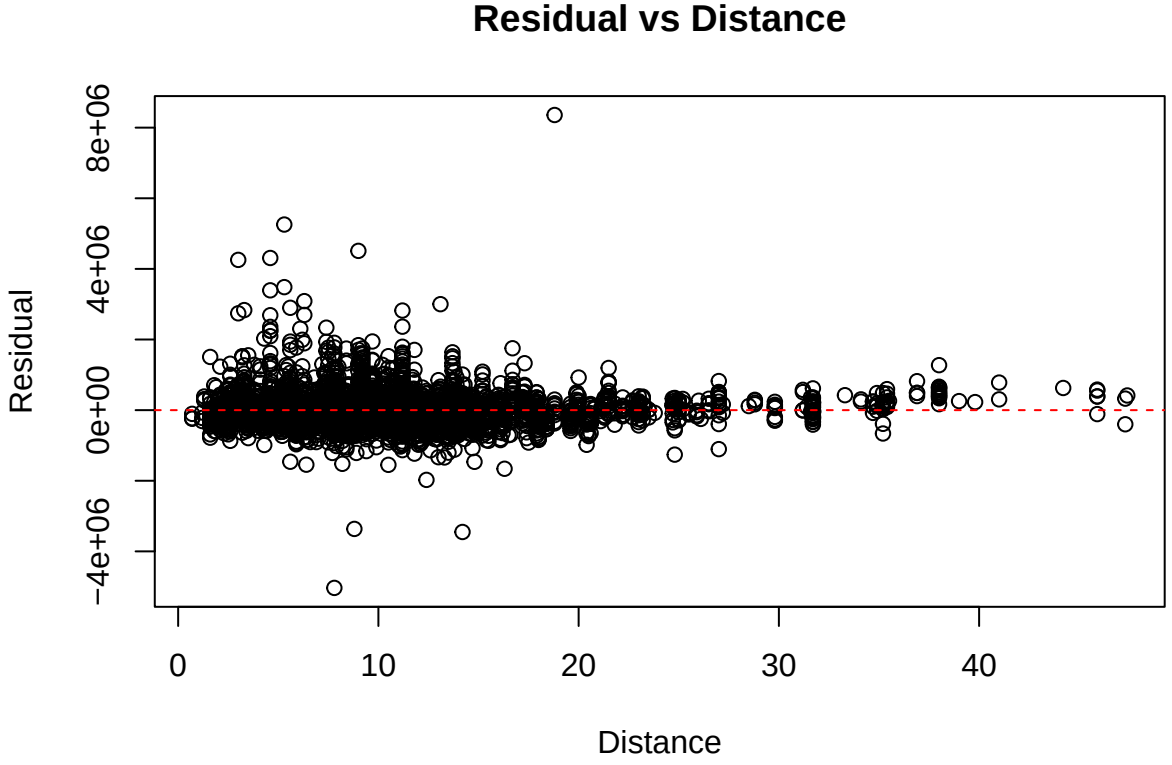
3 Model 1

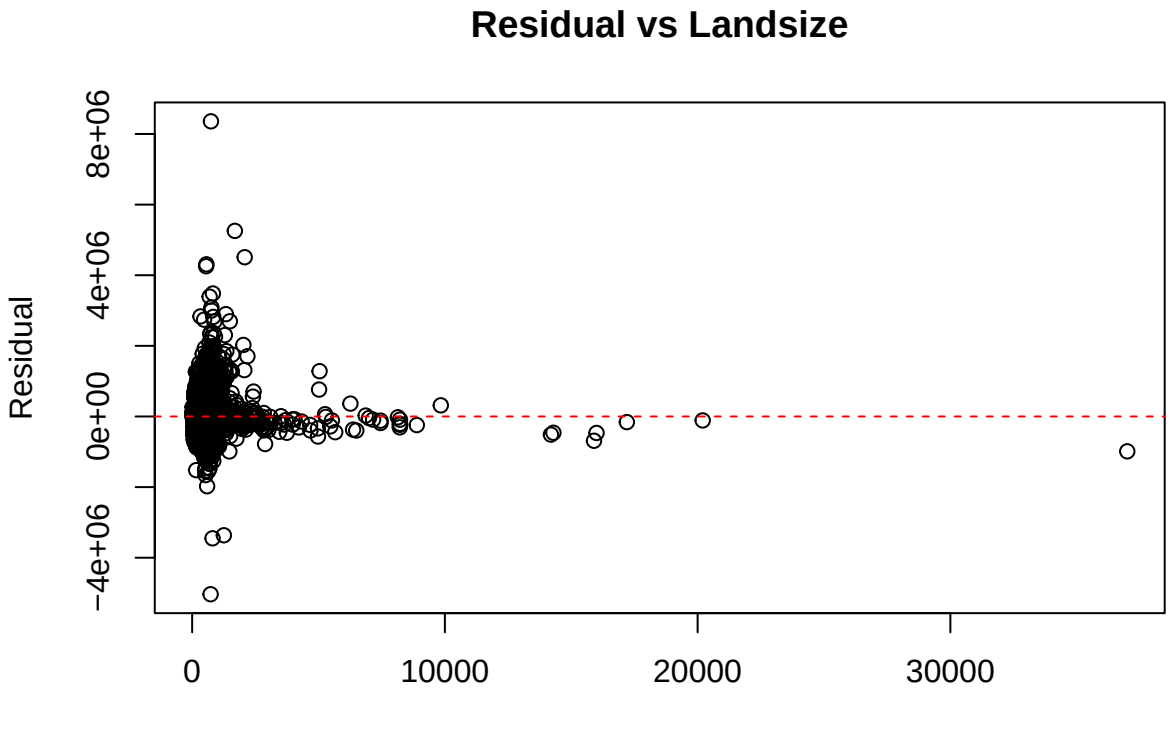
```
##
## Call:
## lm(formula = Price ~ Type + Method + Distance + Bedroom2 + Bathroom +
##     Landsize + Car + BuildingArea + Propertycount + BuildingAge,
##     data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5034076  -259705   -59885   180742  8358065
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.048e+05  4.734e+04   2.215  0.02682 *
## Typet         -8.731e+04  2.782e+04  -3.138  0.00171 **
## Typeu         -1.460e+05  2.691e+04  -5.425 6.10e-08 ***
## MethodS        2.680e+04  2.282e+04   1.174  0.24033
## MethodSA      -5.472e+04  9.356e+04  -0.585  0.55863
## MethodSP      -6.721e+04  2.868e+04  -2.343  0.01916 *
## MethodVB       4.900e+04  3.353e+04   1.461  0.14399
## Distance      -3.015e+04  1.321e+03 -22.822  < 2e-16 ***
## Bedroom2       7.938e+04  1.163e+04   6.825 9.94e-12 ***
## Bathroom       2.721e+05  1.327e+04  20.508  < 2e-16 ***
## Landsize       1.726e+01  7.507e+00   2.299  0.02154 *
## Car            5.479e+04  8.220e+03   6.665 2.95e-11 ***
## BuildingArea   1.956e+03  9.816e+01  19.923  < 2e-16 ***
## Propertycount -1.371e+00  1.643e+00  -0.834  0.40417
## BuildingAge    4.796e+03  2.278e+02  21.055  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 490300 on 4624 degrees of freedom
## Multiple R-squared:  0.4966, Adjusted R-squared:  0.4951
## F-statistic: 325.8 on 14 and 4624 DF,  p-value: < 2.2e-16
```

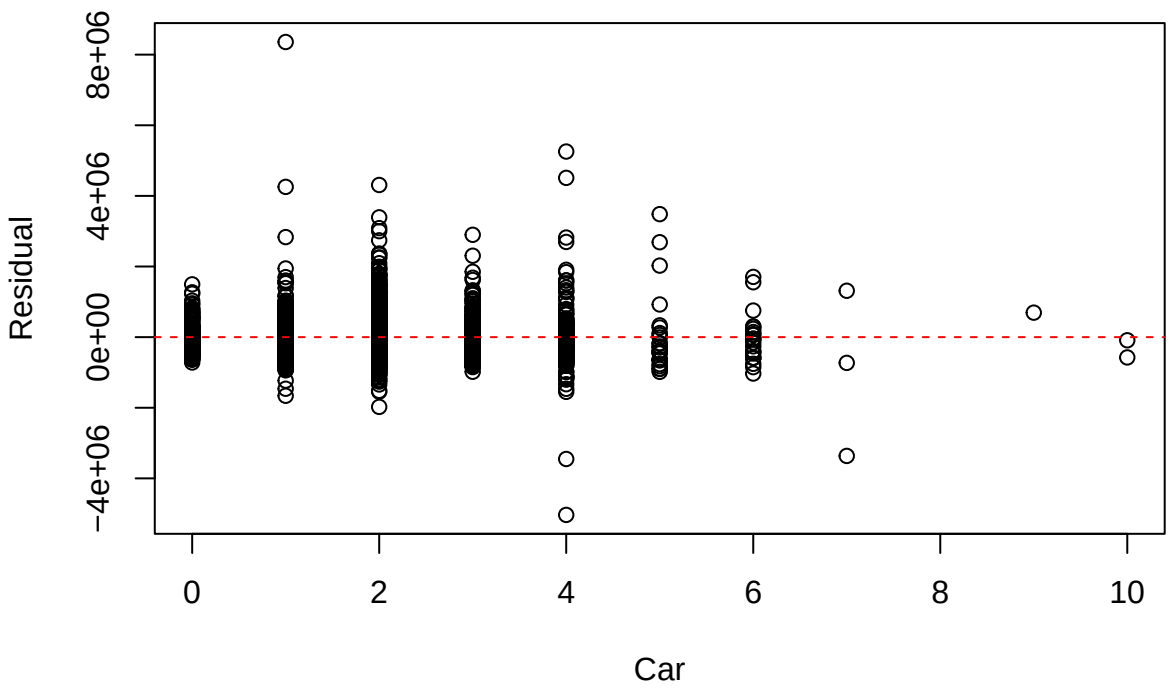

Residual vs Fitted
Residuals
8e+06
4e+06
0e+00
−4e+06
0e+00
2e+06
4e+06
6e+06
Fitted

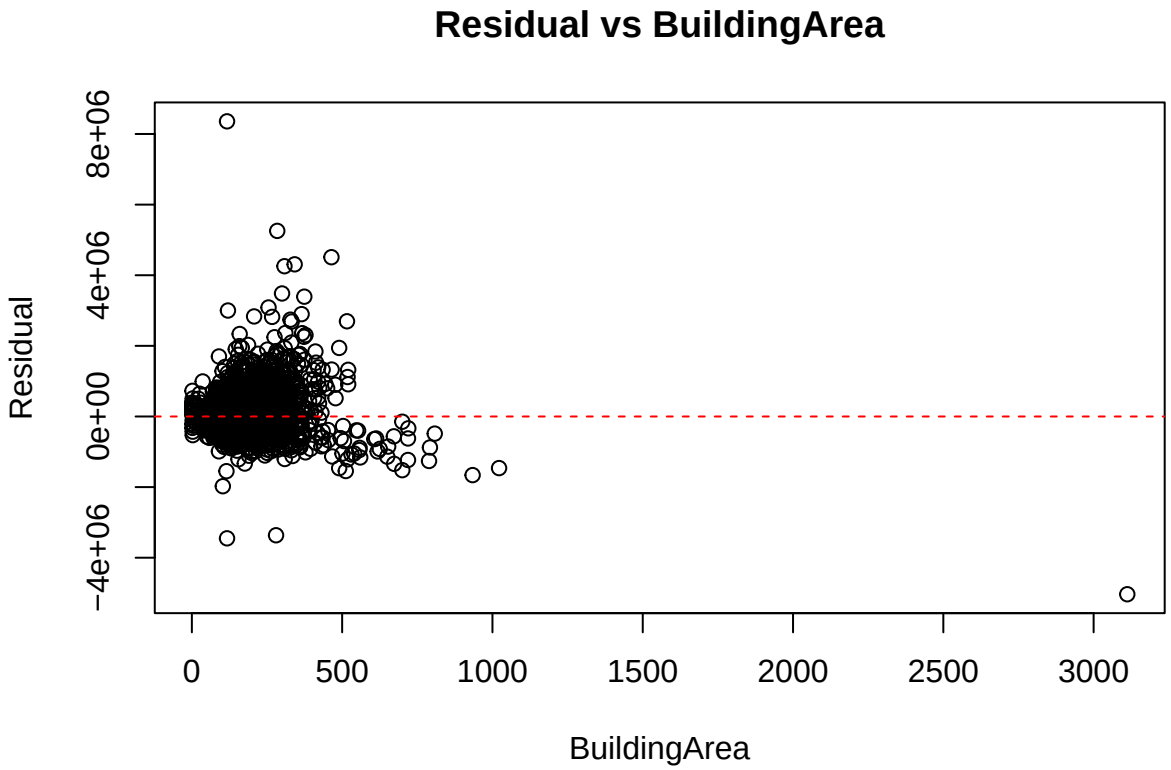

Residual vs Bedroom2
Residual
8e+06
4e+06
0e+00
−4e+06
0
2
4
6
8
Bedroom2

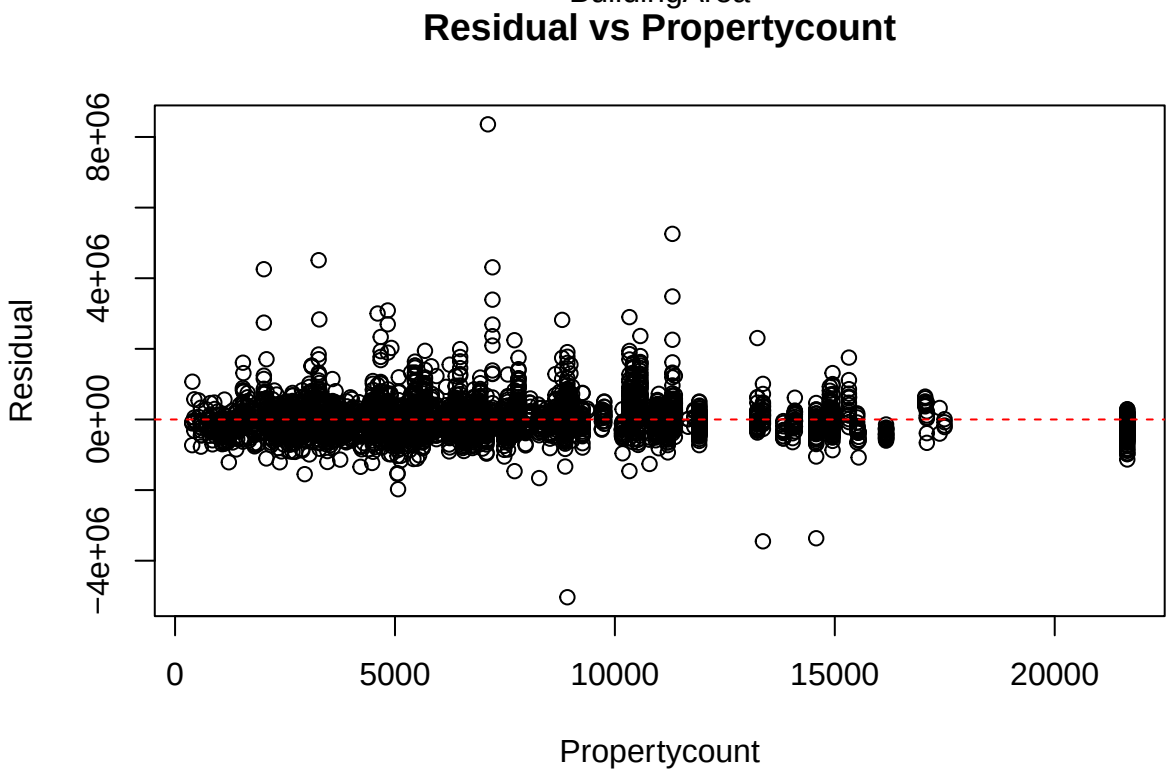

Residual vs Bathroom
Residual
8e+06
4e+06
0e+00
−4e+06
1
2
3
4
5
6
7
8
Bathroom

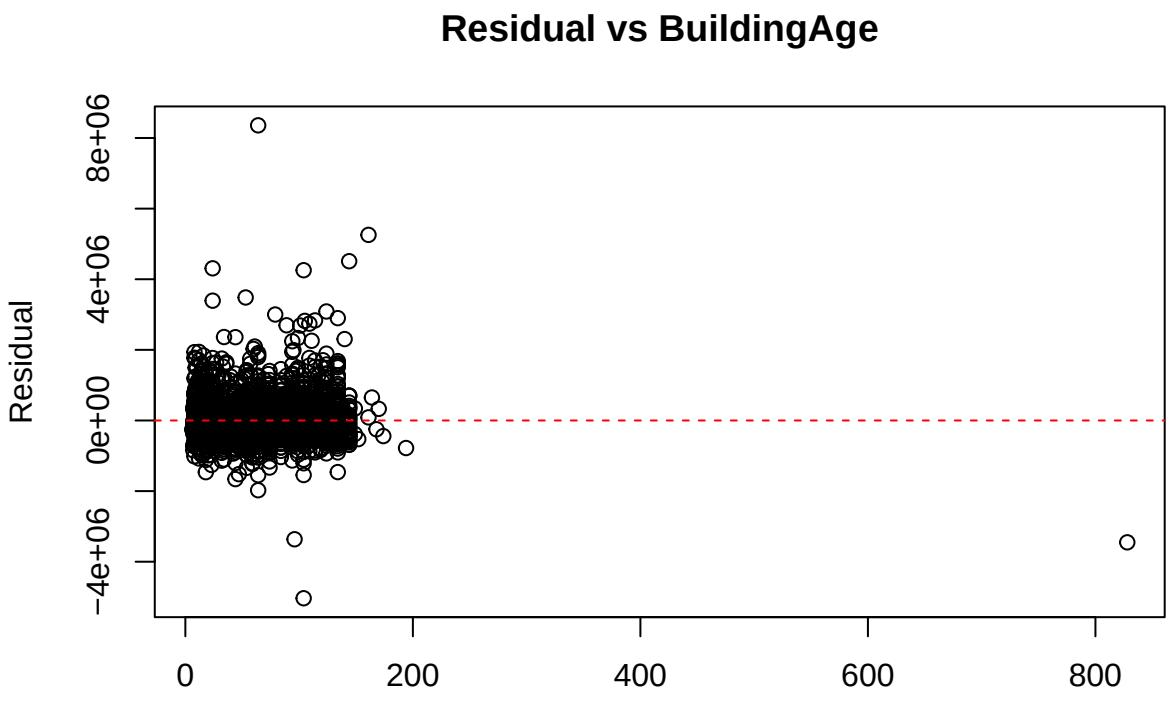

Residual vs Distance
Residual
8e+06
4e+06
0e+00
−4e+06
0
10
20
30
40
Distance

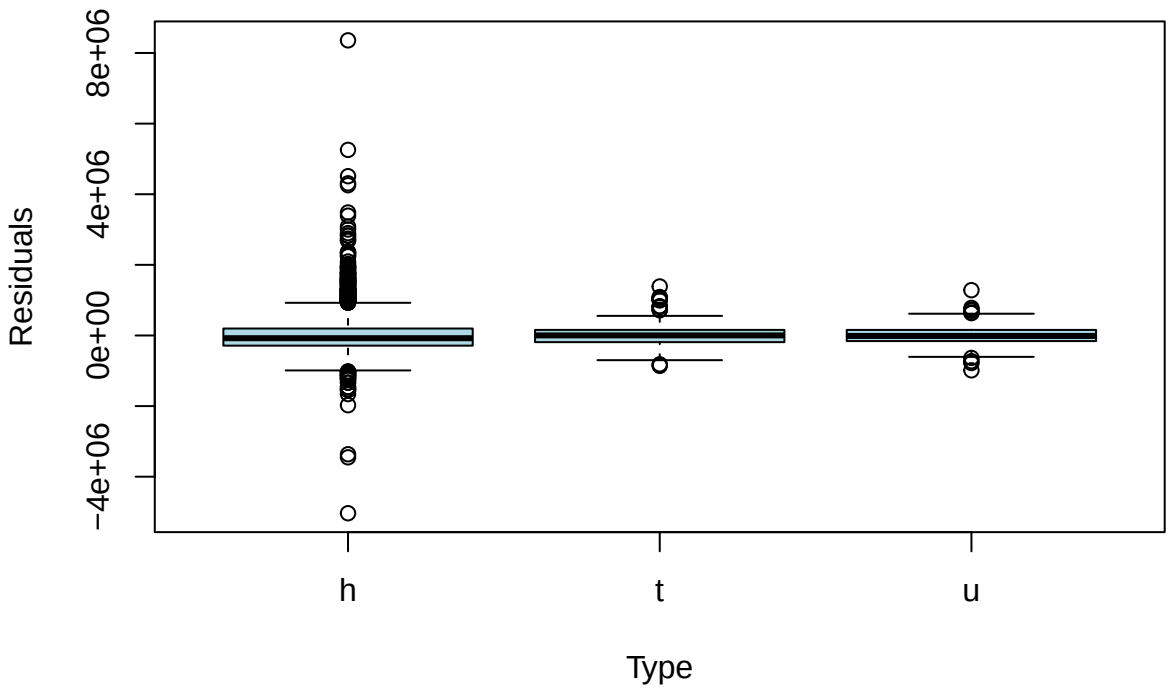

Residual vs Landsize
Residual
8e+06
4e+06
0e+00
−4e+06
0
10000
20000
30000
Landsize

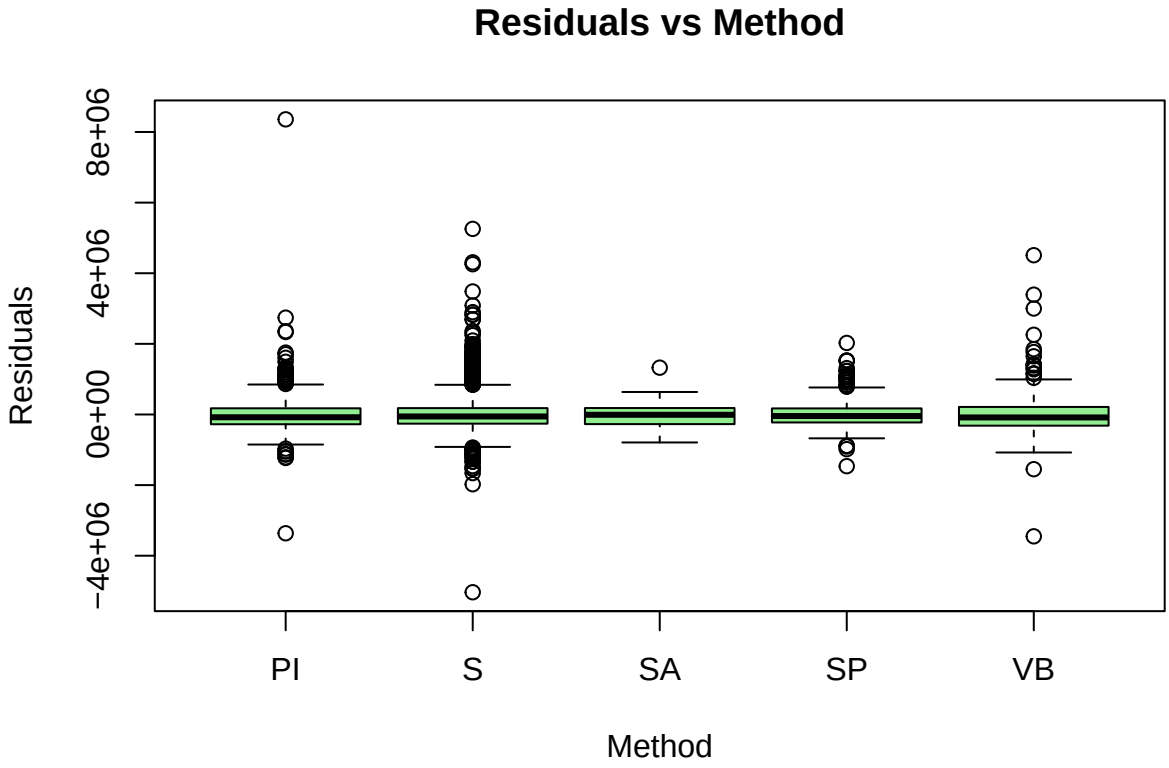

Residual vs Car
Residual
8e+06
4e+06
0e+00
−4e+06
0
2
4
6
8
10
Car

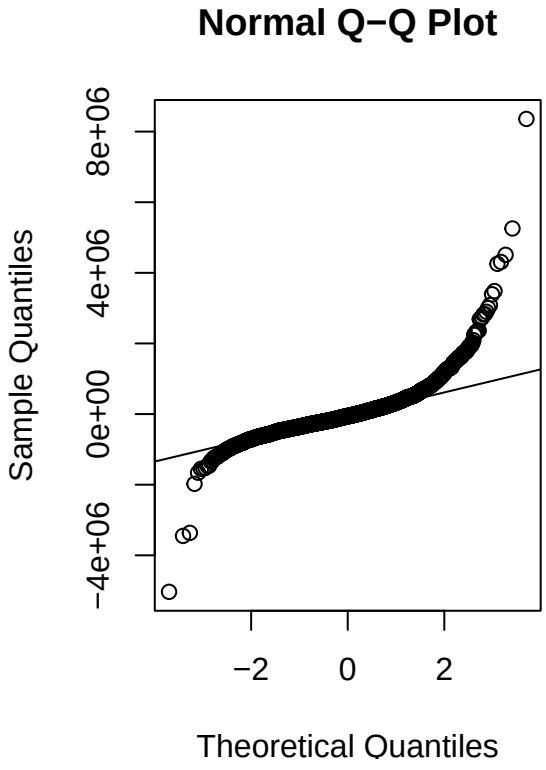

Residual vs BuildingArea
Residual
8e+06
4e+06
0e+00
−4e+06
0
500
1000
1500
2000
2500
3000
BuildingArea

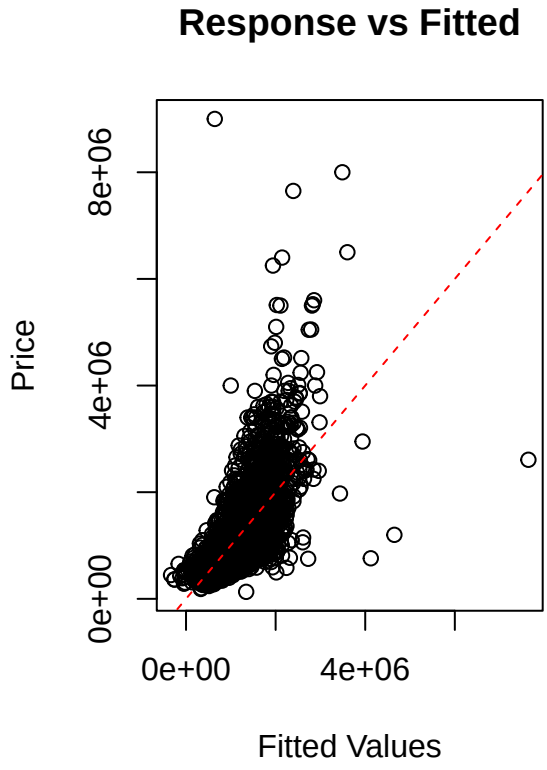

Residual vs Propertycount
Residual
8e+06
4e+06
0e+00
−4e+06
0
5000
10000
15000
20000
Propertycount

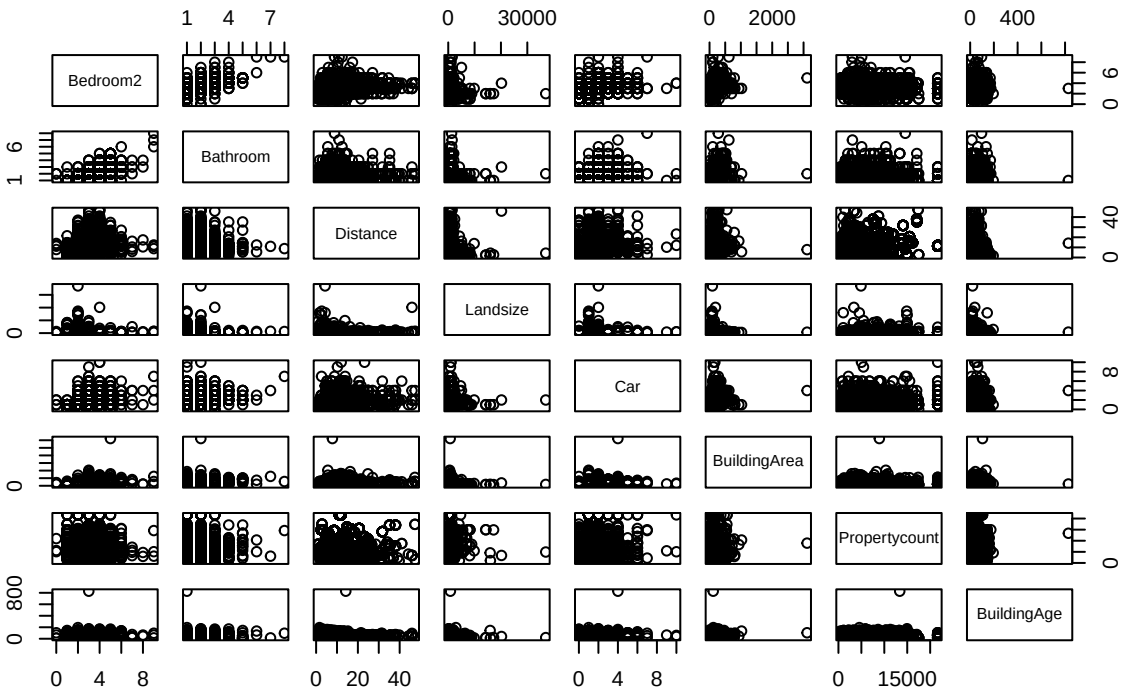

Residual vs BuildingAge
Residual
8e+06
4e+06
0e+00
−4e+06
0
200
400
600
800
BuildingAge


Residuals vs Type
Residuals
8e+06
4e+06
0e+00
−4e+06
h
t
u
Type


Residuals vs Method
Residuals
8e+06
4e+06
0e+00
−4e+06
PI
S
SA
SP
VB
Method


Normal Q−Q Plot
Sample Quantiles
8e+06
4e+06
0e+00
−4e+06
−2
0
2
Theoretical Quantiles


Response vs Fitted
Price
8e+06
4e+06
0e+00
0e+00
4e+06
Fitted Values




```
## bcPower Transformations to Multinormality
##               Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
## Bedroom2        0.7193        0.72       0.6759       0.7626
## Car             0.5835        0.58       0.5681       0.5989
## Distance        0.2518        0.25       0.2184       0.2852
## Landsize       -0.0342       -0.03      -0.0540      -0.0144
## BuildingArea    0.3110        0.31       0.2922       0.3297
## Price          -0.1203       -0.12      -0.1601      -0.0805
##
## Likelihood ratio test that transformation parameters are equal to 0
##  (all log transformations)
##                                       LRT df       pval
## LR test, lambda = (0 0 0 0 0 0) 16055.76  6 < 2.22e-16
##
## Likelihood ratio test that no transformations are needed
##                                       LRT df       pval
## LR test, lambda = (1 1 1 1 1 1) 22476.17  6 < 2.22e-16
```

$$\log(\text{Housing Price}) = \beta_0+\beta_1\cdot\text{Type}+\beta_2\cdot\text{Method}+\beta_3\cdot\text{Distance}^{1/4}+\beta_4\cdot\text{Bedroom}^{0.72}+\beta_5\cdot\text{Bathroom}+\beta_6\cdot\log(\text{Landsize})+\beta_7\cdot\text{Car}^{1/2}$$

4 Transform to Model 2

```
##
## Call:
## lm(formula = Price_trans ~ Type + Method + Distance_trans + Bedroom2_trans +
##     Bathroom + Landsize_trans + Car_trans + BuildingArea_trans +
##     Propertycount + BuildingAge, data = train_data)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.53676 -0.22102 -0.00802  0.20628  2.57858
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          1.205e+01  5.499e-02 219.136  < 2e-16 ***
## Typet               -2.056e-02  1.876e-02  -1.096 0.273080
## Typeu               -2.292e-01  1.768e-02 -12.965  < 2e-16 ***
## MethodS              5.652e-02  1.488e-02   3.799 0.000147 ***
## MethodSA             2.465e-02  6.098e-02   0.404 0.686102
## MethodSP            -3.126e-02  1.870e-02  -1.672 0.094674 .
## MethodVB             3.490e-02  2.186e-02   1.597 0.110387
## Distance_trans      -1.148e-01  3.495e-03 -32.833  < 2e-16 ***
## Bedroom2_trans       1.045e-01  1.485e-02   7.036 2.27e-12 ***
## Bathroom             1.526e-01  8.741e-03  17.454  < 2e-16 ***
## Landsize_trans       4.415e-02  7.381e-03   5.982 2.37e-09 ***
## Car_trans            6.232e-02  1.231e-02   5.062 4.30e-07 ***
## BuildingArea_trans   1.972e-01  7.822e-03  25.213  < 2e-16 ***
## Propertycount       -2.871e-06  1.072e-06  -2.679 0.007413 **
## BuildingAge          3.777e-03  1.498e-04  25.207  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3196 on 4624 degrees of freedom
## Multiple R-squared:  0.6139, Adjusted R-squared:  0.6127
## F-statistic: 525.1 on 14 and 4624 DF,  p-value: < 2.2e-16
```

Table 3: Model 2 Coefficients, t-values, and p-values

Variable	Coefficient Estimate	t-value	p-value
(Intercept)	12.0504	219.1357	0.0000
Typet	-0.0206	-1.0961	0.2731
Typeu	-0.2292	-12.9648	0.0000
MethodS	0.0565	3.7991	0.0001
MethodSA	0.0246	0.4042	0.6861
MethodSP	-0.0313	-1.6716	0.0947
MethodVB	0.0349	1.5968	0.1104
Distance_trans	-0.1148	-32.8330	0.0000
Bedroom2_trans	0.1045	7.0358	0.0000
Bathroom	0.1526	17.4544	0.0000
Landsize_trans	0.0442	5.9821	0.0000
Car_trans	0.0623	5.0623	0.0000
BuildingArea_trans	0.1972	25.2127	0.0000
Propertycount	0.0000	-2.6789	0.0074
BuildingAge	0.0038	25.2074	0.0000

$$\log(\text{Housing Price}) = \beta_0 + \beta_1 \cdot \text{TypeU} + \beta_2 \cdot \text{MethodS} + \beta_3 \cdot \text{Distance}^{1/4} + \beta_4 \cdot \text{Bedroom}^{0.72} + \beta_5 \cdot \text{Bathroom} + \beta_6 \cdot \log(\text{Landsize}) + \beta_7 \cdot \text{Car}$$

5 Transform to Model 3

```
##
## Call:
## lm(formula = Price_trans ~ Typeu + MethodS + Distance_trans +
##     Bedroom2_trans + Bathroom + Landsize_trans + Car_trans +
##     BuildingArea_trans + Propertycount + BuildingAge, data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.54700 -0.22011 -0.00848  0.20818  2.58713
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         1.201e+01  4.801e-02 250.199  < 2e-16 ***
## Typeu              -2.237e-01  1.711e-02 -13.078  < 2e-16 ***
## MethodS             6.160e-02  9.990e-03   6.166 7.60e-10 ***
## Distance_trans     -1.154e-01  3.465e-03 -33.307  < 2e-16 ***
## Bedroom2_trans      1.063e-01  1.480e-02   7.181 8.02e-13 ***
## Bathroom            1.530e-01  8.693e-03  17.603  < 2e-16 ***
## Landsize_trans      4.618e-02  7.140e-03   6.468 1.10e-10 ***
## Car_trans           6.274e-02  1.232e-02   5.094 3.64e-07 ***
## BuildingArea_trans  1.990e-01  7.779e-03  25.586  < 2e-16 ***
## Propertycount      -2.827e-06  1.071e-06  -2.640  0.00832 **
## BuildingAge         3.845e-03  1.393e-04  27.605  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3199 on 4628 degrees of freedom
## Multiple R-squared:  0.613,  Adjusted R-squared:  0.6121
## F-statistic:   733 on 10 and 4628 DF,  p-value: < 2.2e-16
```

```
## Analysis of Variance Table
##
## Model 1: Price_trans ~ Type + Method + Distance_trans + Bedroom2_trans +
##     Bathroom + Landsize_trans + Car_trans + BuildingArea_trans +
##     Propertycount + BuildingAge
## Model 2: Price_trans ~ Typeu + MethodS + Distance_trans + Bedroom2_trans +
##     Bathroom + Landsize_trans + Car_trans + BuildingArea_trans +
##     Propertycount + BuildingAge
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
## 1   4624 472.38
## 2   4628 473.50 -4   -1.1238 2.7501 0.02668 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

6 Check VIF

```
##              Typeu            MethodS     Distance_trans     Bedroom2_trans
##           1.367056           1.014356           1.299489           2.280635
##           Bathroom     Landsize_trans          Car_trans BuildingArea_trans
##           1.884371           1.260682           1.320197           2.030233
##      Propertycount        BuildingAge
##           1.007442           1.335149
```

$$\log(\text{Housing Price}) = \beta_0 + \beta_1 \cdot \text{Type} + \beta_2 \cdot \text{Method} + \beta_3 \cdot \text{Distance}^{1/4} + \beta_4 \cdot \text{Bedroom}^{0.72} + \beta_5 \cdot \text{Bathroom} + \beta_6 \cdot \log(\text{Landsize}) + \beta_7 \cdot \text{Car}^{1/2}$$

7 Create Model 4, validate compared to Model 3

```
# Drop the least significant predicor
model_4 <- lm(Price_trans ~ Typeu + MethodS + Distance_trans + Bedroom2_trans +
                      Bathroom + Landsize_trans + Car_trans +
                      BuildingArea_trans + BuildingAge, data = train_data)
summary(model_4)
```

```
##
## Call:
## lm(formula = Price_trans ~ Typeu + MethodS + Distance_trans +
##     Bedroom2_trans + Bathroom + Landsize_trans + Car_trans +
##     BuildingArea_trans + BuildingAge, data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.55703 -0.21910 -0.00849  0.20868  2.58735
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         11.9893720  0.0472008 254.008  < 2e-16 ***
## Typeu               -0.2258699  0.0170992 -13.209  < 2e-16 ***
## MethodS              0.0614311  0.0099960   6.146 8.64e-10 ***
## Distance_trans      -0.1154689  0.0034667 -33.308  < 2e-16 ***
## Bedroom2_trans       0.1069171  0.0148051   7.222 5.98e-13 ***
## Bathroom             0.1532293  0.0086983  17.616  < 2e-16 ***
## Landsize_trans       0.0465117  0.0071437   6.511 8.26e-11 ***
## Car_trans            0.0618802  0.0123193   5.023 5.28e-07 ***
## BuildingArea_trans   0.1992908  0.0077834  25.605  < 2e-16 ***
## BuildingAge          0.0038364  0.0001393  27.533  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3201 on 4629 degrees of freedom
## Multiple R-squared:  0.6124, Adjusted R-squared:  0.6116
## F-statistic: 812.6 on 9 and 4629 DF,  p-value: < 2.2e-16
```

```
# Compute Adjusted R², AIC, and BIC for Model 3
adj_r2 <- summary(model_3)$adj.r.squared
aic <- AIC(model_3)
bic <- BIC(model_3)

cat("Adjusted R² for Model 3:", adj_r2, "\n")
```

```
## Adjusted R² for Model 3: 0.6121426
```

```
cat("AIC for Model 3:", aic, "\n")
```

```
## AIC for Model 3: 2602.256
```

```r
cat("BIC for Model 3:", bic, "\n")
```

```
## BIC for Model 3: 2679.563
```

```r
# Compare with competing models
cat("Adjusted R² for Model 4:", summary(model_4)$adj.r.squared, "\n")
```

```
## Adjusted R² for Model 4: 0.6116425
```

```r
cat("AIC for Model 4:", AIC(model_4), "\n")
```

```
## AIC for Model 4: 2607.237
```

```r
cat("BIC for Model 4:", BIC(model_4), "\n")
```

```
## BIC for Model 4: 2678.101
```

```r
# Define the cross-validation method
cv_control <- trainControl(method = "cv", number = 10)

# Train Model 3 using cross-validation
cv_model_3 <- train(
  Price_trans ~ Typeu + MethodS + Distance_trans + Bedroom2_trans +
    Bathroom + Landsize_trans + Car_trans +
    BuildingArea_trans + Propertycount + BuildingAge,
  data = train_data,
  method = "lm",
  trControl = cv_control
)

# Display cross-validation results
print(cv_model_3)
```

```
## Linear Regression
##
## 4639 samples
##   10 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 4175, 4175, 4176, 4175, 4175, 4175, ...
## Resampling results:
##
##   RMSE       Rsquared   MAE
##   0.3208896  0.6104959  0.2510744
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```r
# Extract performance metrics (e.g., RMSE)
cat("Cross-validated RMSE for Model 3:", cv_model_3$results$RMSE, "\n")
```

```
## Cross-validated RMSE for Model 3: 0.3208896
```

```r
# Initialize storage for prediction errors
se <- NULL

# Perform LOOCV
for (i in 1:nrow(train_data)) {
```

```r
  # Create training and testing sets
  LOOtrain <- train_data[-i, ]
  LOOtest <- train_data[i, , drop = FALSE]  # Single observation for testing

  # Fit the model on training data (Model 3)
  model <- lm(Price_trans ~ Typeu + MethodS + Distance_trans + Bedroom2_trans +
                Bathroom + Landsize_trans + Car_trans +
                BuildingArea_trans + Propertycount + BuildingAge, data = LOOtrain)

  # Make prediction on the test observation
  fitted <- predict(model, newdata = LOOtest)

  # Compute squared prediction error
  se_i <- (LOOtest$Price_trans - fitted)^2

  # Store the error
  se <- c(se, se_i)
}

# Compute the mean squared error (MSE)
mse_loocv <- mean(se)
cat("LOOCV Mean Squared Error (MSE):", mse_loocv, "\n")
```

```
## LOOCV Mean Squared Error (MSE): 0.1031178
```

```r
# Initialize storage for prediction errors
se_model_4 <- NULL

# Perform LOOCV
for (i in 1:nrow(train_data)) {
  # Create training and testing sets
  LOOtrain <- train_data[-i, ]
  LOOtest <- train_data[i, , drop = FALSE]  # Single observation for testing

  # Fit the model on training data (Model 4)
  model <- lm(Price_trans ~ Typeu + MethodS + Distance_trans + Bedroom2_trans +
                Bathroom + Landsize_trans + Car_trans +
                BuildingArea_trans + BuildingAge, data = LOOtrain)

  # Make prediction on the test observation
  fitted <- predict(model, newdata = LOOtest)

  # Compute squared prediction error
  se_i <- (LOOtest$Price_trans - fitted)^2

  # Store the error
  se_model_4 <- c(se_model_4, se_i)
}

# Compute the mean squared error (MSE) for Model 4
mse_model_4 <- mean(se_model_4)
cat("LOOCV Mean Squared Error (MSE) for Model 4:", mse_model_4, "\n")
```

```
## LOOCV Mean Squared Error (MSE) for Model 4: 0.1032318
```

8 Check on Test Data with Model 3 and Model 4

```r
# Predict on the test data using Model 3
test_data$Typeu <- ifelse(test_data$Type == "u", 1, 0)  # Ensure Typeu exists in test data
test_data$MethodS <- ifelse(test_data$Method == "S", 1, 0)  # Ensure MethodS exists in test data

# Predictions for Model 3
predictions_model_3 <- predict(model_3, newdata = test_data)

# Compute performance metrics for Model 3
actual <- test_data$Price_trans
rmse_model_3 <- sqrt(mean((predictions_model_3 - actual)^2))
mae_model_3 <- mean(abs(predictions_model_3 - actual))
r2_model_3 <- cor(predictions_model_3, actual)^2

cat("Model 3 - Test RMSE:", rmse_model_3, "\n")
```

```
## Model 3 - Test RMSE: 0.3664022
```

```r
cat("Model 3 - Test MAE:", mae_model_3, "\n")
```

```
## Model 3 - Test MAE: 0.2880909
```

```r
cat("Model 3 - Test R²:", r2_model_3, "\n")
```

```
## Model 3 - Test R²: 0.5416221
```

```r
# Predictions for Model 4
predictions_model_4 <- predict(model_4, newdata = test_data)

# Compute performance metrics for Model 4
rmse_model_4 <- sqrt(mean((predictions_model_4 - actual)^2))
mae_model_4 <- mean(abs(predictions_model_4 - actual))
r2_model_4 <- cor(predictions_model_4, actual)^2

cat("Model 4 - Test RMSE:", rmse_model_4, "\n")
```

```
## Model 4 - Test RMSE: 0.3662991
```

```r
cat("Model 4 - Test MAE:", mae_model_4, "\n")
```

```
## Model 4 - Test MAE: 0.2876354
```

```r
cat("Model 4 - Test R²:", r2_model_4, "\n")
```

```
## Model 4 - Test R²: 0.5415343
```

```r
# Prepare validation metrics for Model 3 and Model 4
validation_metrics <- data.frame(
  Metric = c("Adjusted R²", "AIC", "BIC", "LOOCV MSE", "Test Data RMSE", "Test Data MAE", "Test Data R²
  Model_3 = c(
    summary(model_3)$adj.r.squared,
    AIC(model_3),
    BIC(model_3),
    mse_loocv,
    rmse_model_3,
    mae_model_3,
    r2_model_3
```

```
  ),
  Model_4 = c(
    summary(model_4)$adj.r.squared,
    AIC(model_4),
    BIC(model_4),
    mse_model_4,
    rmse_model_4,
    mae_model_4,
    r2_model_4
  )
)

# Format and display the table
library(kableExtra)
validation_metrics %>%
  kable(
    format = "latex",
    caption = "Comparison of Validation Metrics for Model 3 and Model 4",
    col.names = c("Metric", "Model 3", "Model 4"),
    digits = 4
  ) %>%
  kable_styling(latex_options = c("hold_position", "striped", "scale_down"))
```

Table 4: Comparison of Validation Metrics for Model 3 and Model 4

Metric	Model 3	Model 4
Adjusted R^2	0.6121	0.6116
AIC	2602.2564	2607.2367
BIC	2679.5635	2678.1015
LOOCV MSE	0.1031	0.1032
Test Data RMSE	0.3664	0.3663
Test Data MAE	0.2881	0.2876
Test Data R^2	0.5416	0.5415

9 Model 3 wins! Identify problematic observations!

```
# Compute leverage values
leverage <- hatvalues(model_3)

# Define a cutoff for high leverage points
n <- nrow(train_data)  # Number of observations
p <- length(coefficients(model_3))  # Number of predictors (including intercept)
cutoff_leverage <- 2 * p / n

# Identify high leverage points
high_leverage <- which(leverage > cutoff_leverage)
cat("High leverage points:", high_leverage, "\n")
```

```
## High leverage points: 57 70 74 95 104 152 156 173 241 260 262 317 382 388 430 439 463 516 542 568 60
```

```
# Compute standardized residuals
standardized_residuals <- rstandard(model_3)
```

```r
# Define a cutoff for outliers
cutoff_residuals <- 2  # Common cutoff for standardized residuals

# Identify outliers
outliers <- which(abs(standardized_residuals) > cutoff_residuals)
cat("Outlier points:", outliers, "\n")
```

```
## Outlier points: 23 30 53 70 103 104 152 200 226 239 247 389 409 411 421 424 437 439 444 445 446 568
```

```r
# Compute Cook's distance
cooks_d <- cooks.distance(model_3)

# Define a cutoff for Cook's distance
cutoff_cooks <- qf(0.5, p, n - p)  # Approximation for identifying influential points

# Identify influential observations
influential_cooks <- which(cooks_d > cutoff_cooks)
cat("Influential points (Cook's Distance):", influential_cooks, "\n")
```

```
## Influential points (Cook's Distance):
```

```r
# Compute DFFITS
dffits_values <- dffits(model_3)

# Define a cutoff for DFFITS
cutoff_dffits <- 2 * sqrt(p / n)

# Identify influential observations
influential_dffits <- which(abs(dffits_values) > cutoff_dffits)
cat("Influential points (DFFITS):", influential_dffits, "\n")
```

```
## Influential points (DFFITS): 1 23 30 49 53 70 103 104 152 173 247 317 357 382 389 421 428 437 439 44
```

```r
# Compute DFBetas
dfbetas_values <- dfbetas(model_3)

# Define a cutoff for DFBetas
cutoff_dfbetas <- 2 / sqrt(n)

# Identify influential observations for each coefficient
influential_dfbetas <- apply(dfbetas_values, 2, function(x) which(abs(x) > cutoff_dfbetas))

# Print influential points for each coefficient
cat("Influential points (DFBetas):\n")
```

```
## Influential points (DFBetas):
```

```r
for (coef in names(influential_dfbetas)) {
  cat(coef, ":", influential_dfbetas[[coef]], "\n")
}
```

```
## (Intercept) : 1 30 42 51 70 81 99 103 104 109 152 173 216 221 259 357 382 409 410 421 424 434 436 43
## Typeu : 70 74 92 104 146 152 175 181 202 205 219 221 230 266 276 290 294 299 312 317 347 351 357 358
## MethodS : 1 23 30 38 51 53 63 104 152 163 194 204 208 210 226 232 247 258 355 384 389 404 418 426 43
## Distance_trans : 1 53 104 542 607 610 626 629 631 633 635 669 781 894 914 921 1124 1178 1212 1290 13
## Bedroom2_trans : 23 30 53 70 104 109 152 163 173 200 226 241 283 317 355 382 406 428 439 445 452 460
```

```
## Bathroom : 1 31 49 103 125 128 152 156 173 226 234 241 247 251 262 304 311 317 355 357 364 382 389 4
## Landsize_trans : 1 51 70 74 92 95 104 200 202 208 219 221 226 251 257 258 378 389 416 428 433 439 44
## Car_trans : 30 38 47 49 70 79 104 124 141 146 216 219 259 265 269 378 389 418 439 538 542 568 601 60
## BuildingArea_trans : 23 63 70 95 99 103 104 126 148 152 173 192 210 224 228 241 247 248 259 260 283 
## Propertycount : 30 53 389 421 437 439 445 601 629 631 633 635 669 921 960 1025 1085 1087 1089 1178 1
## BuildingAge : 1 7 43 49 51 53 63 103 104 109 124 125 146 226 259 357 406 407 410 414 418 421 431 434
```

```
# Compute leverage values
leverage <- hatvalues(model_3)

# Define a cutoff for high leverage points
n <- nrow(train_data)  # Number of observations
p <- length(coefficients(model_3))  # Number of predictors (including intercept)
cutoff_leverage <- 2 * p / n

# Identify high leverage points
high_leverage <- which(leverage > cutoff_leverage)
num_high_leverage <- length(high_leverage)
prop_high_leverage <- num_high_leverage / n
cat("High leverage points:", num_high_leverage, "(", round(100 * prop_high_leverage, 2), "% of dataset)"
```

```
## High leverage points: 302 ( 6.51 % of dataset)
```

```
# Compute standardized residuals
standardized_residuals <- rstandard(model_3)

# Define a cutoff for outliers
cutoff_residuals <- 2  # Common cutoff for standardized residuals

# Identify outliers
outliers <- which(abs(standardized_residuals) > cutoff_residuals)
num_outliers <- length(outliers)
prop_outliers <- num_outliers / n
cat("Outlier points:", num_outliers, "(", round(100 * prop_outliers, 2), "% of dataset)\n")
```

```
## Outlier points: 185 ( 3.99 % of dataset)
```

```
# Compute Cook's distance
cooks_d <- cooks.distance(model_3)

# Define a cutoff for Cook's distance
cutoff_cooks <- qf(0.5, p, n - p)  # Approximation for identifying influential points

# Identify influential observations
influential_cooks <- which(cooks_d > cutoff_cooks)
num_influential_cooks <- length(influential_cooks)
prop_influential_cooks <- num_influential_cooks / n
cat("Influential points (Cook's Distance):", num_influential_cooks, "(", round(100 * prop_influential_c
```

```
## Influential points (Cook's Distance): 0 ( 0 % of dataset)
```

```
# Compute DFFITS
dffits_values <- dffits(model_3)

# Define a cutoff for DFFITS
cutoff_dffits <- 2 * sqrt(p / n)
```

```r
# Identify influential observations
influential_dffits <- which(abs(dffits_values) > cutoff_dffits)
num_influential_dffits <- length(influential_dffits)
prop_influential_dffits <- num_influential_dffits / n
cat("Influential points (DFFITS):", num_influential_dffits, "(", round(100 * prop_influential_dffits, 2
```

```
## Influential points (DFFITS): 249 ( 5.37 % of dataset)
```

```r
# Compute DFBetas
dfbetas_values <- dfbetas(model_3)

# Define a cutoff for DFBetas
cutoff_dfbetas <- 2 / sqrt(n)

# Identify influential observations for each coefficient
influential_dfbetas <- apply(dfbetas_values, 2, function(x) which(abs(x) > cutoff_dfbetas))

# Count unique influential points across all coefficients
unique_influential_dfbetas <- unique(unlist(influential_dfbetas))
num_influential_dfbetas <- length(unique_influential_dfbetas)
prop_influential_dfbetas <- num_influential_dfbetas / n

# Print results for DFBetas
cat("Influential points (DFBetas):", num_influential_dfbetas, "(", round(100 * prop_influential_dfbetas
```

```
## Influential points (DFBetas): 1144 ( 24.66 % of dataset)
```

```r
# Prepare the data for the table
influential_points <- data.frame(
  Metric = c("High Leverage Points",
             "Outlier Points",
             "Influential Points (Cook's Distance)"),
  Count = c(num_high_leverage,
            num_outliers,
            num_influential_cooks),
  Proportion = c(
    paste0(round(100 * prop_high_leverage, 2), "%"),
    paste0(round(100 * prop_outliers, 2), "%"),
    paste0(round(100 * prop_influential_cooks, 2), "%")
  )
)

# Create the table using kable
library(knitr)
kable(
  influential_points,
  col.names = c("Metric", "Count", "Proportion"),
  caption = "Summary of Influential Points Analysis for Model 3"
)
```

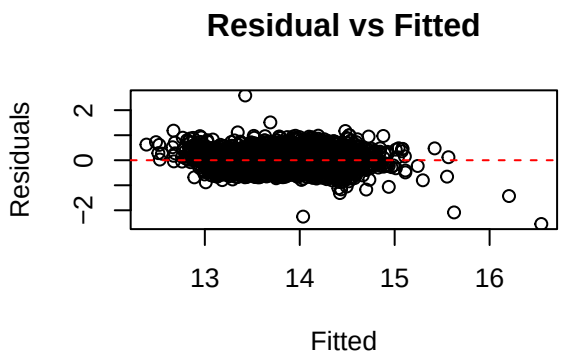
Table 5: Summary of Influential Points Analysis for Model 3

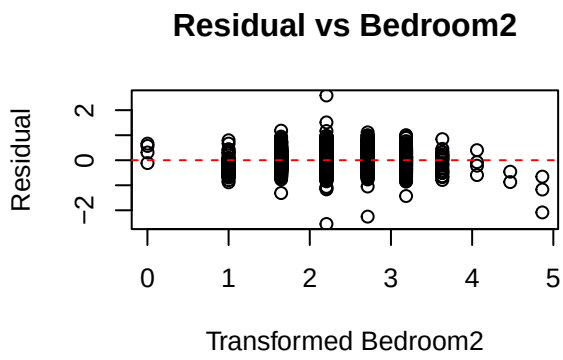
Metric	Count	Proportion
High Leverage Points	302	6.51%
Outlier Points	185	3.99%

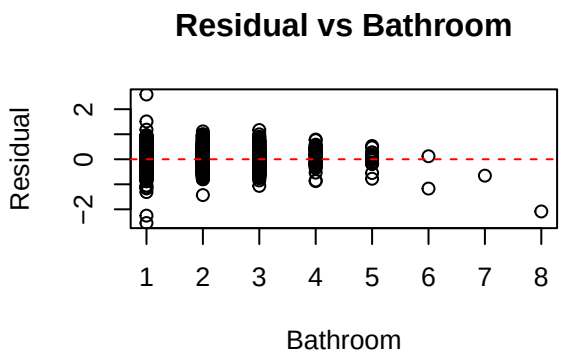
Metric	Count	Proportion
Influential Points (Cook's Distance)	0	0%

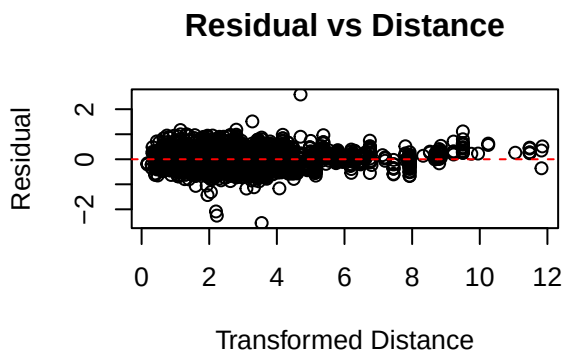
```r
# 666
#888
#999
```

```r
# Fit Model 3
model_3 <- lm(Price_trans ~ Typeu + MethodS + Distance_trans + Bedroom2_trans +
                Bathroom + Landsize_trans + Car_trans +
                BuildingArea_trans + Propertycount + BuildingAge, data = train_data)

# Extract residuals and fitted values from the model
y_value <- resid(model_3)
x_value <- fitted(model_3)

# Set up plotting layout
par(mfrow = c(2, 2))

# Plot Residuals vs Fitted Values
plot(x = x_value, y = y_value, main = "Residual vs Fitted", xlab = "Fitted", ylab = "Residuals")
abline(h = 0, col = "red", lty = 2)

# Residual plots for numeric predictors
plot(x = train_data$Bedroom2_trans, y = y_value, main = "Residual vs Bedroom2", xlab = "Transformed Bedr
abline(h = 0, col = "red", lty = 2)

plot(x = train_data$Bathroom, y = y_value, main = "Residual vs Bathroom", xlab = "Bathroom", ylab = "Res
abline(h = 0, col = "red", lty = 2)

plot(x = train_data$Distance_trans, y = y_value, main = "Residual vs Distance", xlab = "Transformed Dist
abline(h = 0, col = "red", lty = 2)
```

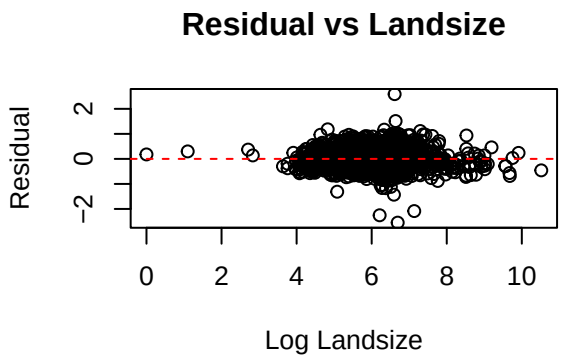


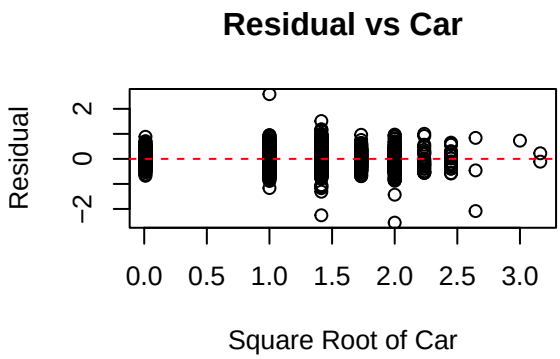


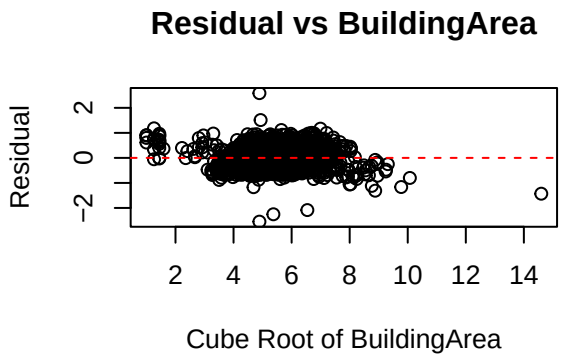


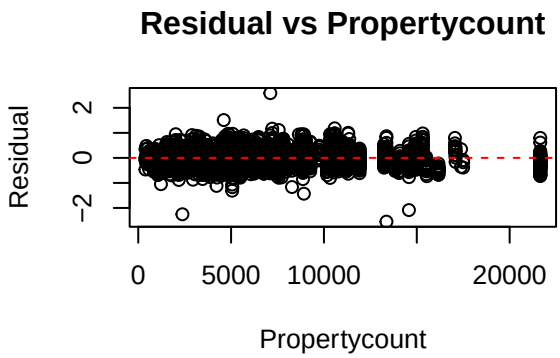



```r
plot(x = train_data$Landsize_trans, y = y_value, main = "Residual vs Landsize", xlab = "Log Landsize", y
abline(h = 0, col = "red", lty = 2)

plot(x = train_data$Car_trans, y = y_value, main = "Residual vs Car", xlab = "Square Root of Car", ylab
abline(h = 0, col = "red", lty = 2)

plot(x = train_data$BuildingArea_trans, y = y_value, main = "Residual vs BuildingArea", xlab = "Cube Roo
abline(h = 0, col = "red", lty = 2)

plot(x = train_data$Propertycount, y = y_value, main = "Residual vs Propertycount", xlab = "Propertycou
abline(h = 0, col = "red", lty = 2)
```

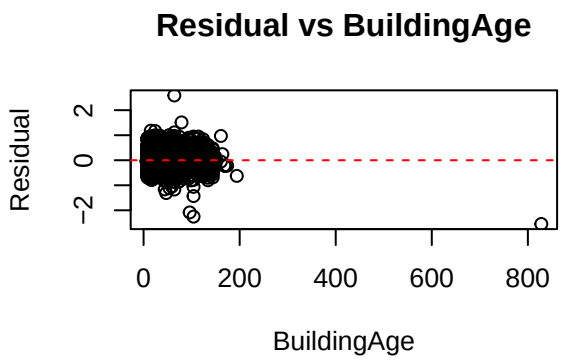


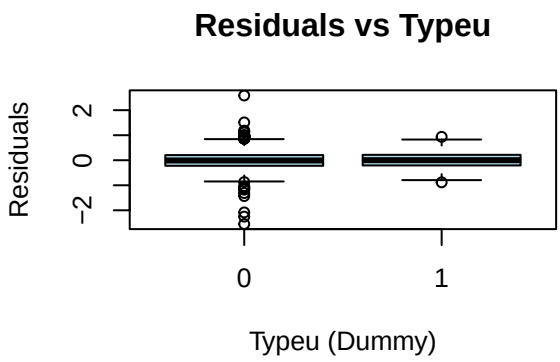


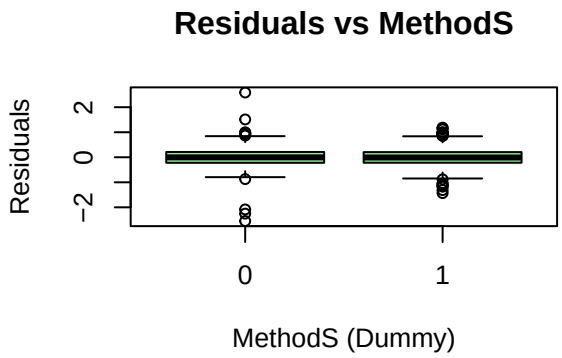




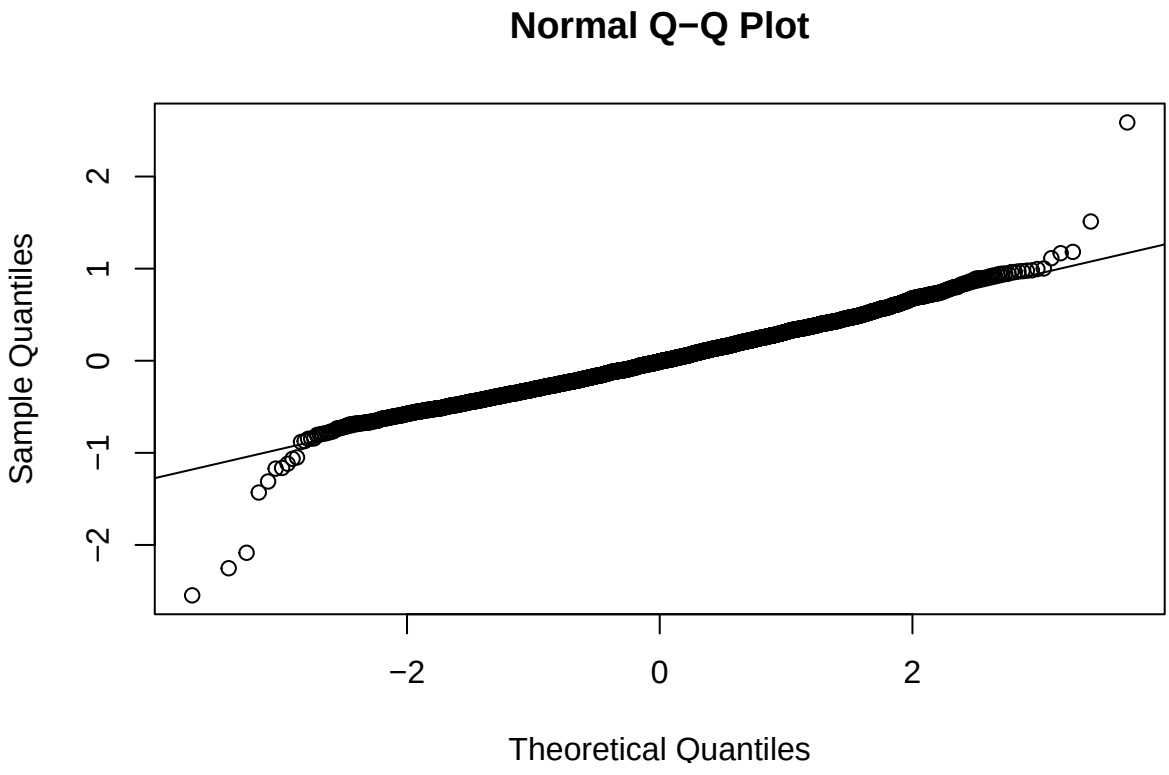

```r
plot(x = train_data$BuildingAge, y = y_value, main = "Residual vs BuildingAge", xlab = "BuildingAge", yl
abline(h = 0, col = "red", lty = 2)

# Residual plot for categorical predictors using boxplots
boxplot(y_value ~ train_data$Typeu, main = "Residuals vs Typeu", xlab = "Typeu (Dummy)", ylab = "Residua
boxplot(y_value ~ train_data$MethodS, main = "Residuals vs MethodS", xlab = "MethodS (Dummy)", ylab = "R

# Reset layout to default
par(mfrow = c(1,1))
```



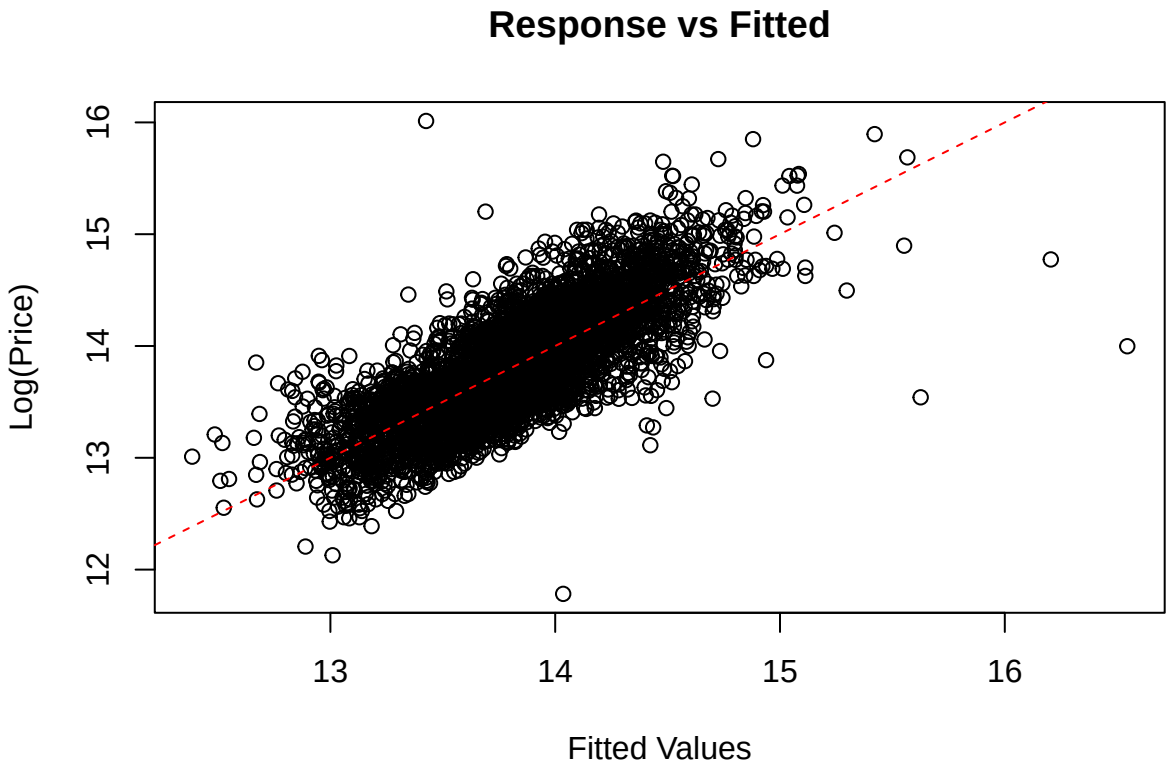




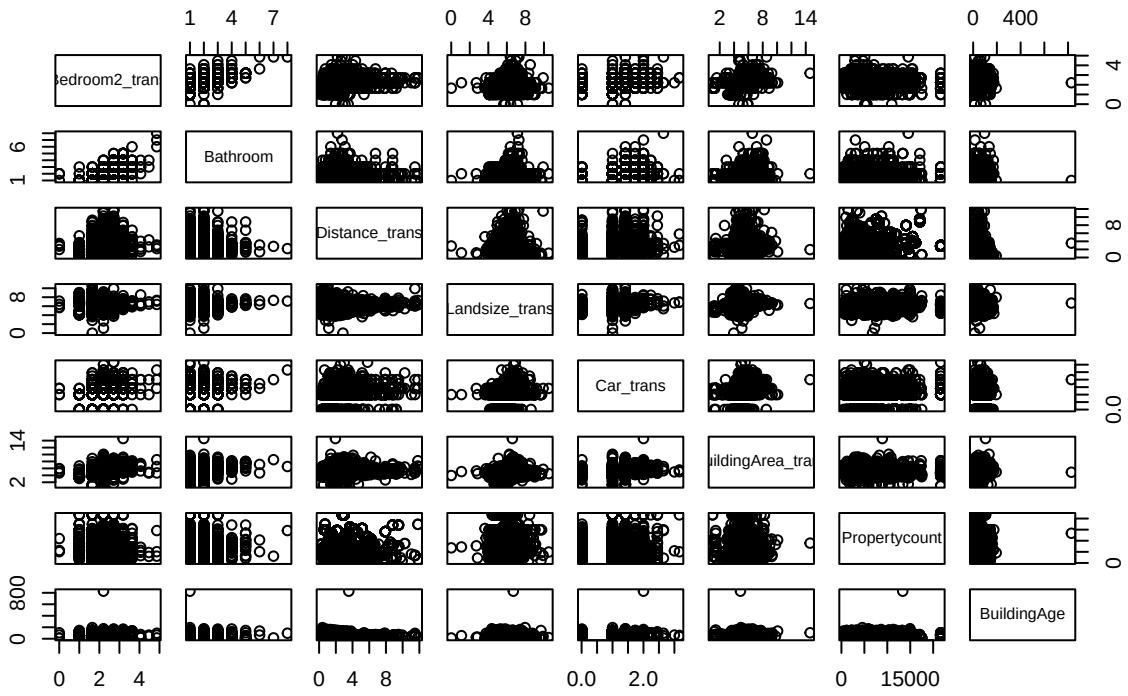

```r
# Q-Q Plot for normality of residuals
qqnorm(y_value, main = "Normal Q-Q Plot")
qqline(y_value)
```



```r
# Plotting the Response vs Fitted values to check additional conditions
plot(x = x_value, y = train_data$Price_trans, main = "Response vs Fitted",
     xlab = "Fitted Values", ylab = "Log(Price)")
abline(a = 0, b = 1, lty = 2, col = "red")
```




```r
# Reset plotting to single plots for pairwise scatter plot
pairs(train_data[, c("Bedroom2_trans", "Bathroom", "Distance_trans", "Landsize_trans",
                     "Car_trans", "BuildingArea_trans", "Propertycount", "BuildingAge")],
      main = "Pairwise Plots of Numeric Predictors (Transformed)",
      col = "black")
```




```r
# Prepare the data for the table
influential_points <- data.frame(
  Metric = c("High Leverage Points",
             "Outlier Points",
             "Influential Points (Cook's Distance)"),
  Count = c(num_high_leverage,
            num_outliers,
            num_influential_cooks),
  Proportion = c(
    paste0(round(100 * prop_high_leverage, 2), "%"),
    paste0(round(100 * prop_outliers, 2), "%"),
    paste0(round(100 * prop_influential_cooks, 2), "%")
  )
)

# Create the table using kable
library(knitr)
kable(
  influential_points,
  col.names = c("Metric", "Count", "Proportion"),
  caption = "Summary of Influential Points Analysis for Model 3"
)
```

Table 6: Summary of Influential Points Analysis for Model 3

Metric	Count	Proportion
High Leverage Points	302	6.51%
Outlier Points	185	3.99%
Influential Points (Cook's Distance)	0	0%

```r
# Load required library
library(knitr)  # For kable

# Get the summary of the model
model_3_summary <- summary(model_3)

# Extract coefficients from the summary
coefficients_table <- as.data.frame(model_3_summary$coefficients)

# Add row names (predictor names) as a column
coefficients_table <- cbind(Predictor = rownames(coefficients_table), coefficients_table)

# Generate the table using kable
kable(
  coefficients_table,
  col.names = c("Predictor", "Estimate", "Std. Error", "t-value", "p-value"),
  caption = "Summary of Model 3 Regression Coefficients"
)
```

Table 7: Summary of Model 3 Regression Coefficients

	Predictor	Estimate	Std. Error	t-value	p-value
(Intercept)	(Intercept)	12.0130275	0.0480140	250.198664	0.0000000
Typeu	Typeu	-0.2237274	0.0171074	-13.077789	0.0000000
MethodS	MethodS	0.0615967	0.0099897	6.166003	0.0000000
Distance_trans	Distance_trans	-0.1153949	0.0034646	-33.306939	0.0000000
Bedroom2_trans	Bedroom2_trans	0.1062626	0.0147976	7.181067	0.0000000
Bathroom	Bathroom	0.1530217	0.0086931	17.602744	0.0000000
Landsize_trans	Landsize_trans	0.0461828	0.0071401	6.468048	0.0000000
Car_trans	Car_trans	0.0627383	0.0123157	5.094181	0.0000004
BuildingArea_trans	BuildingArea_trans	0.1990320	0.0077790	25.585889	0.0000000
Propertycount	Propertycount	-0.0000028	0.0000011	-2.639884	0.0083213
BuildingAge	BuildingAge	0.0038452	0.0001393	27.605377	0.0000000