Final Project: AirBnB

2025-06-14

```
library(ggplot2)
library(readr)
library(dplyr)
```

```
##
## 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(robustbase)
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

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

Model 1

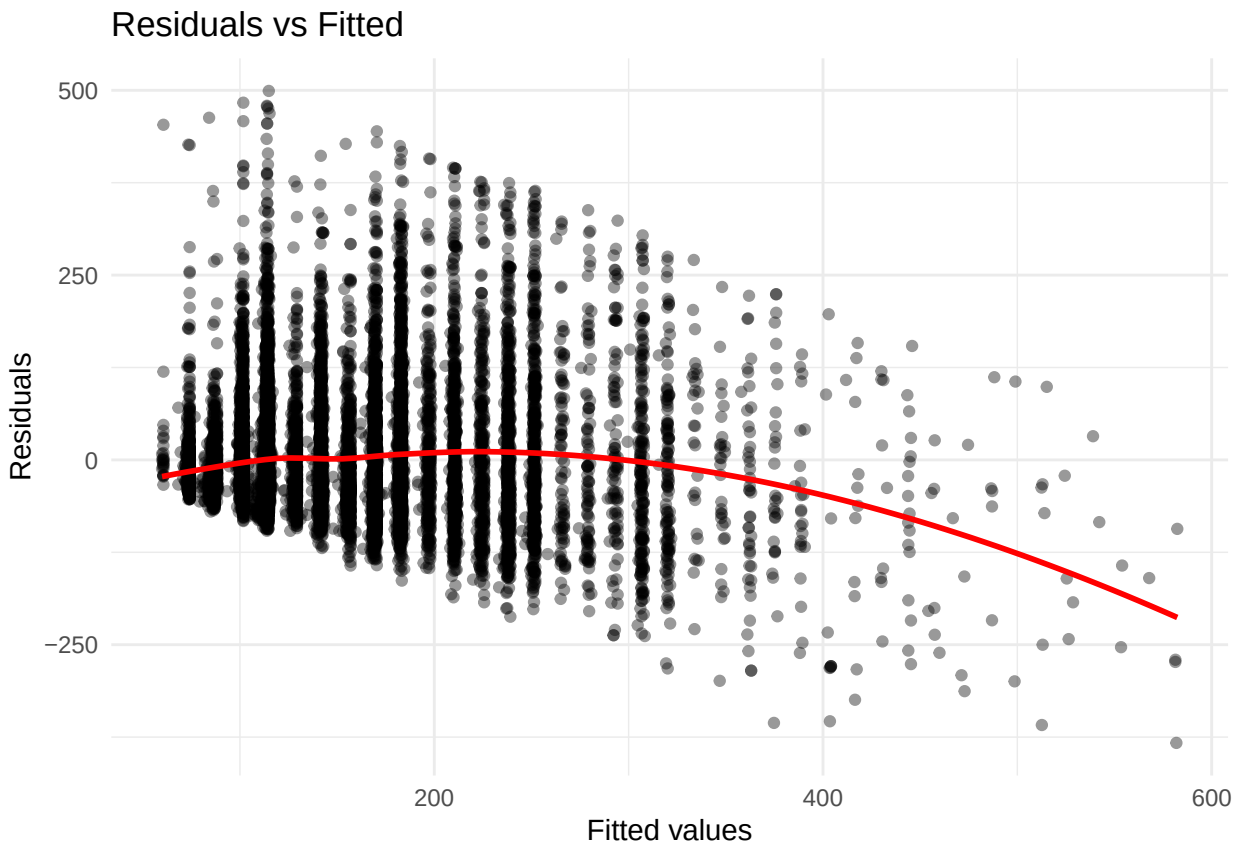
```
df <- read.csv('cleaned_airbnb.csv')
df$log_reviews <- log(df$number_of_reviews + 1)
model <- lm(price ~ accommodates +
              bedrooms +
              number_of_reviews +
              host_is_superhost +
              review_scores_rating,
            data = df)
summary(model)
```

```
##
## Call:
## lm(formula = price ~ accommodates + bedrooms + number_of_reviews +
##     host_is_superhost + review_scores_rating, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -382.84  -49.34  -19.59   31.05  499.16
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          33.029022   1.852539  17.829  < 2e-16 ***
```

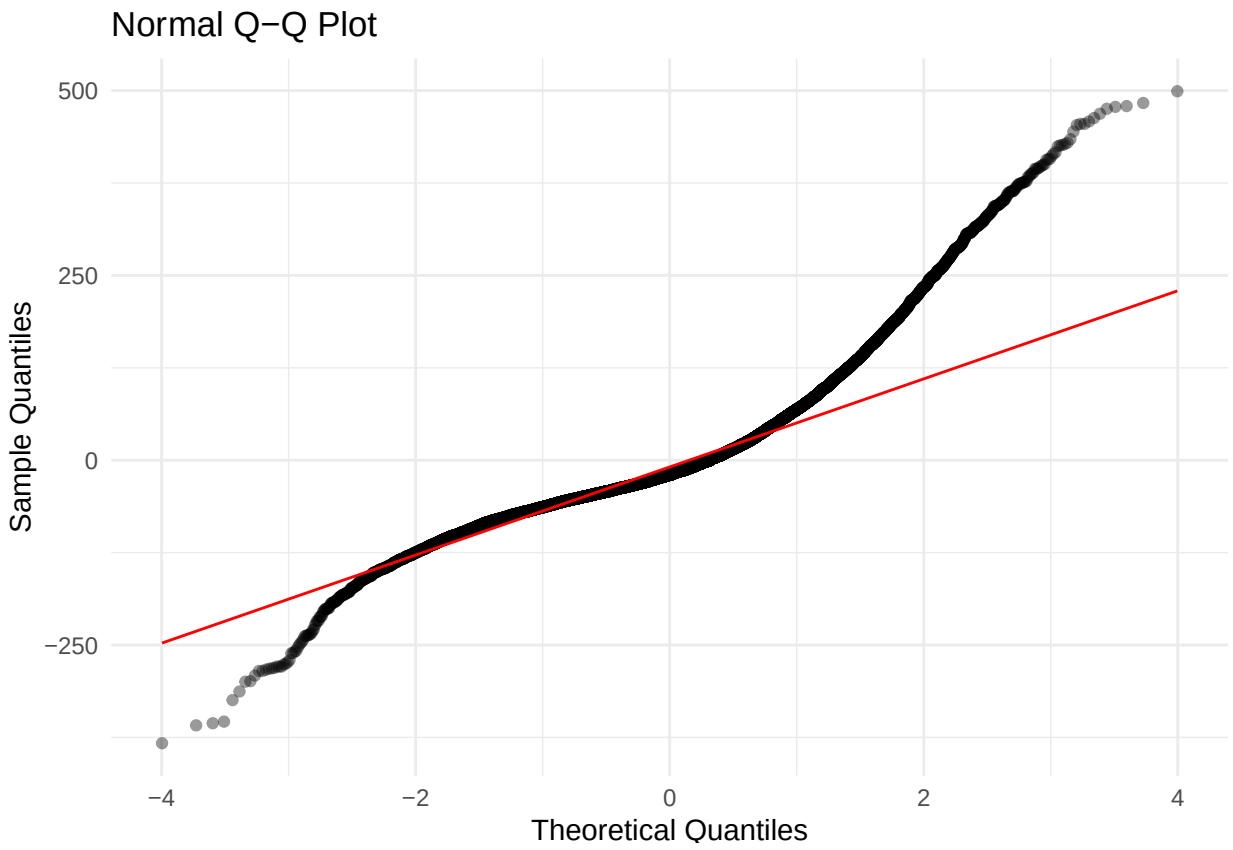
```
## accommodates         27.600425   0.539131  51.194  < 2e-16 ***
## bedrooms             13.506157   1.202855  11.228  < 2e-16 ***
## number_of_reviews     0.003958   0.011618   0.341    0.733
## host_is_superhost    -0.709177   1.462356  -0.485    0.628
## review_scores_rating  2.568063   0.361337   7.107 1.24e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 84.18 on 15545 degrees of freedom
## Multiple R-squared:  0.3609, Adjusted R-squared:  0.3607
## F-statistic:  1755 on 5 and 15545 DF,  p-value: < 2.2e-16
```

```
df$residuals <- residuals(model)
df$fitted <- fitted(model)
df$std_resid <- rstandard(model)
df$leverage <- hatvalues(model)
df$cooks_distance <- cooks.distance(model)

ggplot(df, aes(x = fitted, y = residuals)) +
  geom_point(alpha = 0.4) +
  geom_smooth(method = "loess", color = "red", se = FALSE) +
  labs(title = "Residuals vs Fitted", x = "Fitted values",
       y = "Residuals") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

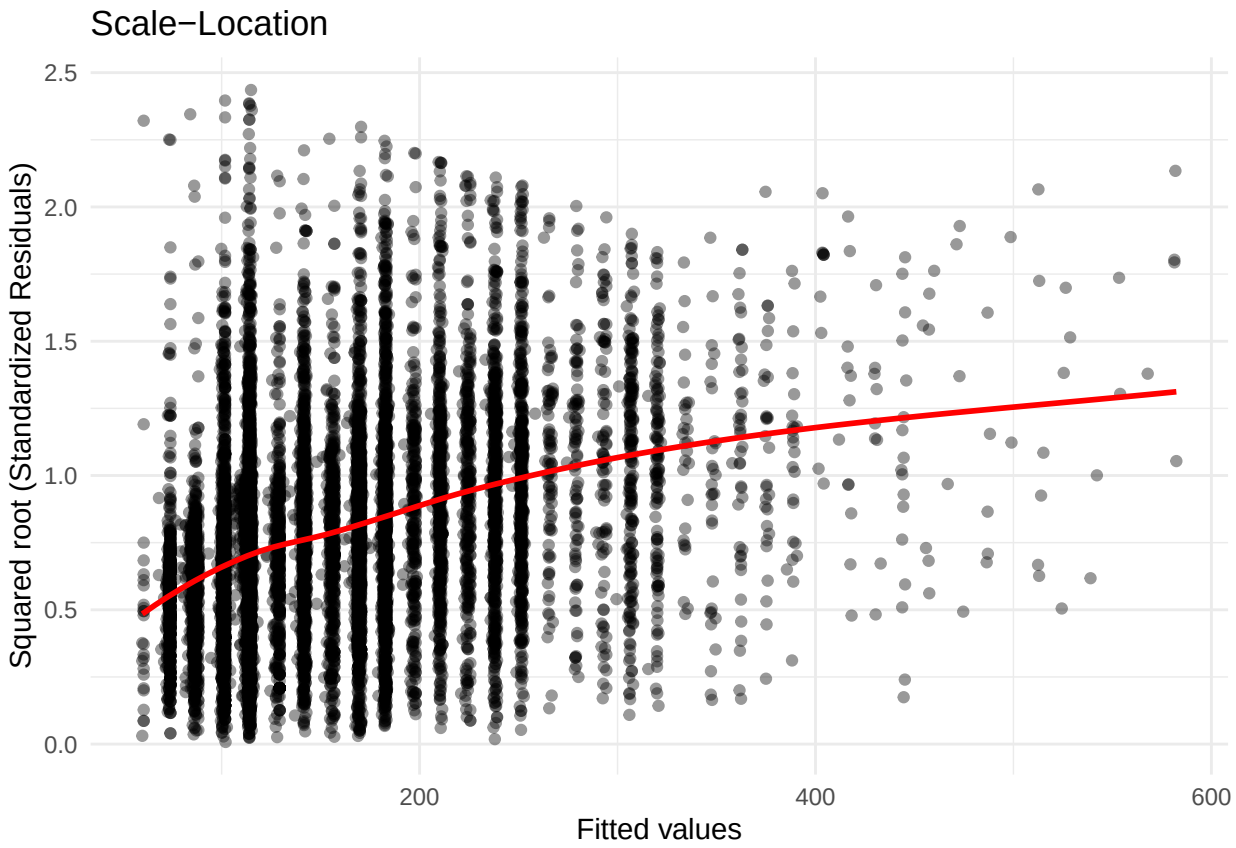


```
ggplot(df, aes(sample = residuals)) +
  stat_qq(alpha = 0.4) +
  stat_qq_line(color = "red") +
  labs(title = "Normal Q-Q Plot", x = "Theoretical Quantiles",
       y = "Sample Quantiles") +
  theme_minimal()
```



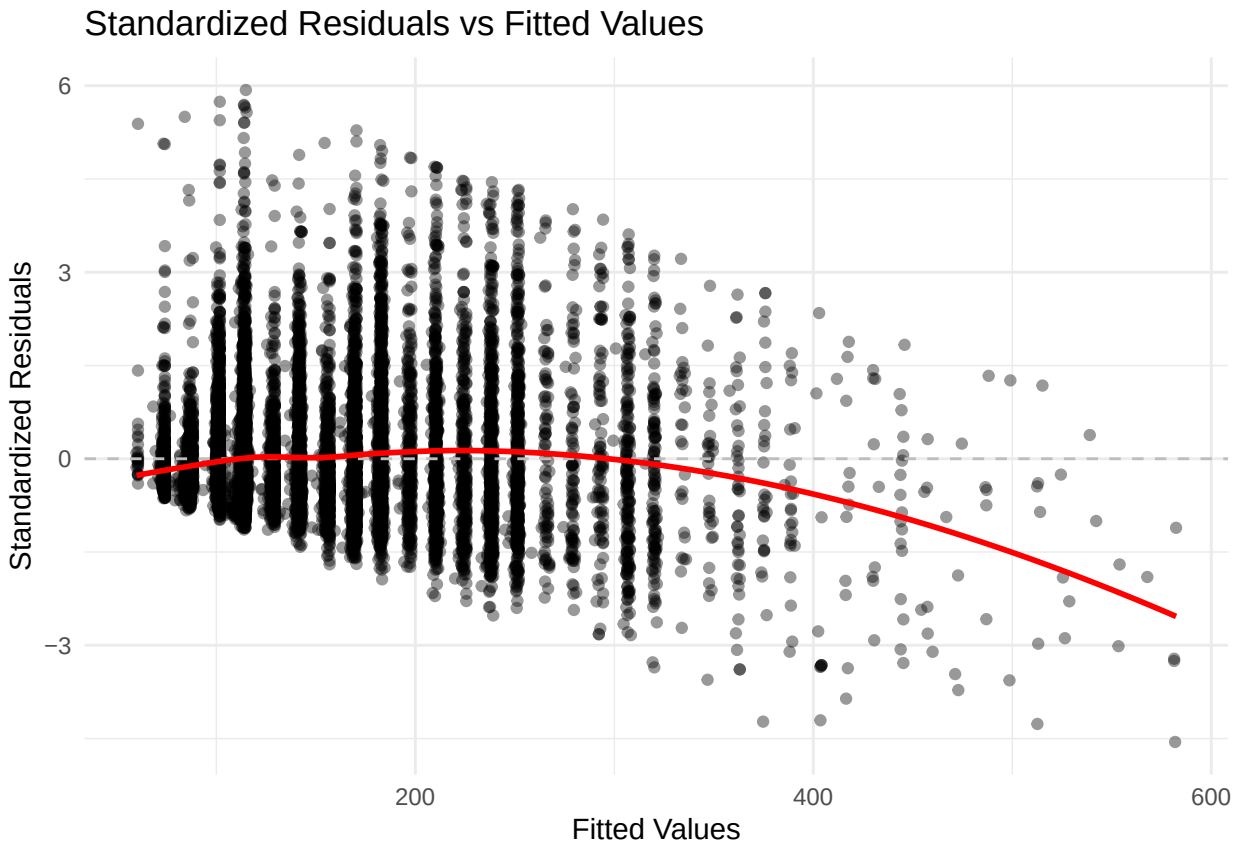

```
ggplot(df, aes(x = fitted, y = sqrt(abs(std_resid)))) +
  geom_point(alpha = 0.4) +
  geom_smooth(method = "loess", color = "red", se = FALSE) +
  labs(title = "Scale-Location", x = "Fitted values",
       y = "Squared root (Standardized Residuals)") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
df$fitted <- fitted(model)
df$std_resid <- rstandard(model)

ggplot(df, aes(x = fitted, y = std_resid)) +
geom_point(alpha = 0.4) +
geom_hline(yintercept = 0, linetype = "dashed", color = "gray") +
geom_smooth(method = "loess", color = "red", se = FALSE) +
labs(
title = "Standardized Residuals vs Fitted Values",
x = "Fitted Values",
y = "Standardized Residuals"
) +
theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
##`geom_smooth()` us
```

Fit transformed model

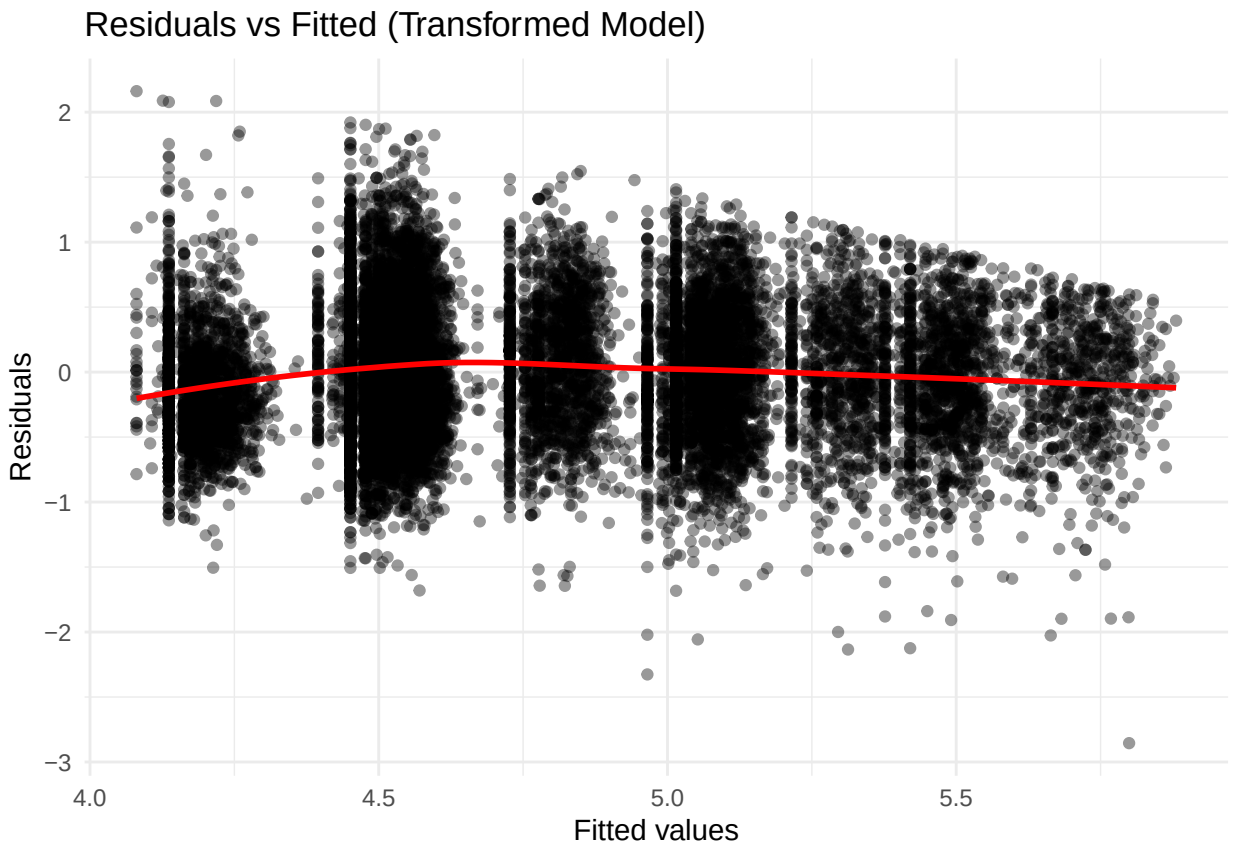
```
model_transformed <- lm(log(price) ~ poly(accommodates, 2) +
                          poly(bedrooms, 2) +
                          log_reviews +
                          review_scores_rating,
                        data = df)

summary(model_transformed)
```

```
##
## Call:
## lm(formula = log(price) ~ poly(accommodates, 2) + poly(bedrooms,
##     2) + log_reviews + review_scores_rating, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.85493 -0.35206 -0.02302  0.33472  2.16155
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 4.716979   0.008641 545.896  < 2e-16 ***
## poly(accommodates, 2)1     45.908141   0.793741  57.838  < 2e-16 ***
## poly(accommodates, 2)2    -17.141913   0.560855 -30.564  < 2e-16 ***
```

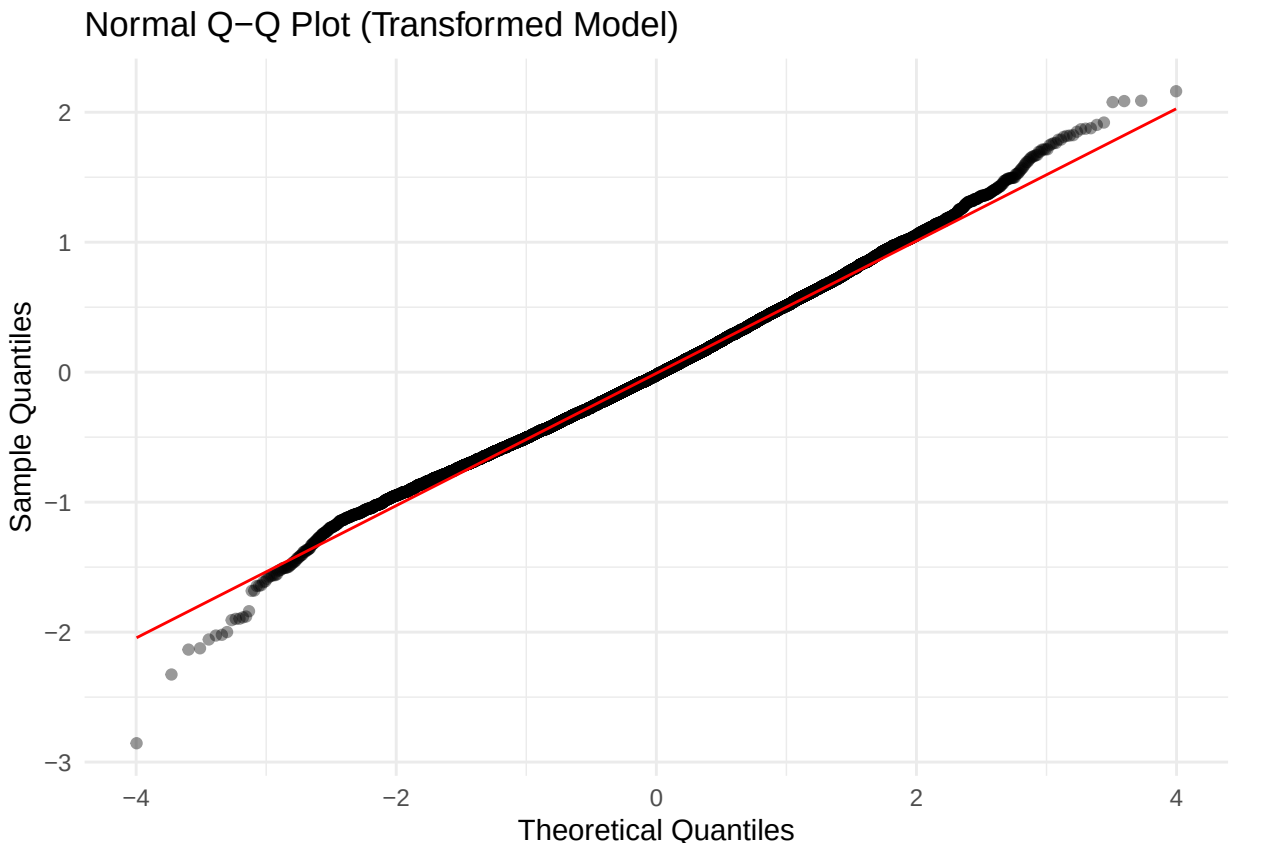
```
## poly(bedrooms, 2)1       4.905610   0.793948   6.179 6.62e-10 ***
## poly(bedrooms, 2)2      -0.618148   0.559645  -1.105    0.269
## log_reviews              0.027474   0.003442   7.983 1.53e-15 ***
## review_scores_rating     0.001491   0.002853   0.523    0.601
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5103 on 15544 degrees of freedom
## Multiple R-squared:  0.4127, Adjusted R-squared:  0.4125
## F-statistic:  1821 on 6 and 15544 DF,  p-value: < 2.2e-16
```

```r
# Add diagnostics to df
df$trans_resid <- residuals(model_transformed)
df$trans_fitted <- fitted(model_transformed)
df$trans_std_resid <- rstandard(model_transformed)
df$trans_leverage <- hatvalues(model_transformed)
df$trans_cooks_distance <- cooks.distance(model_transformed)



ggplot(df, aes(x = trans_fitted, y = trans_resid)) +
  geom_point(alpha = 0.4) +
  geom_smooth(method = "loess", color = "red", se = FALSE) +
  labs(title = "Residuals vs Fitted (Transformed Model)",
       x = "Fitted values", y = "Residuals") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

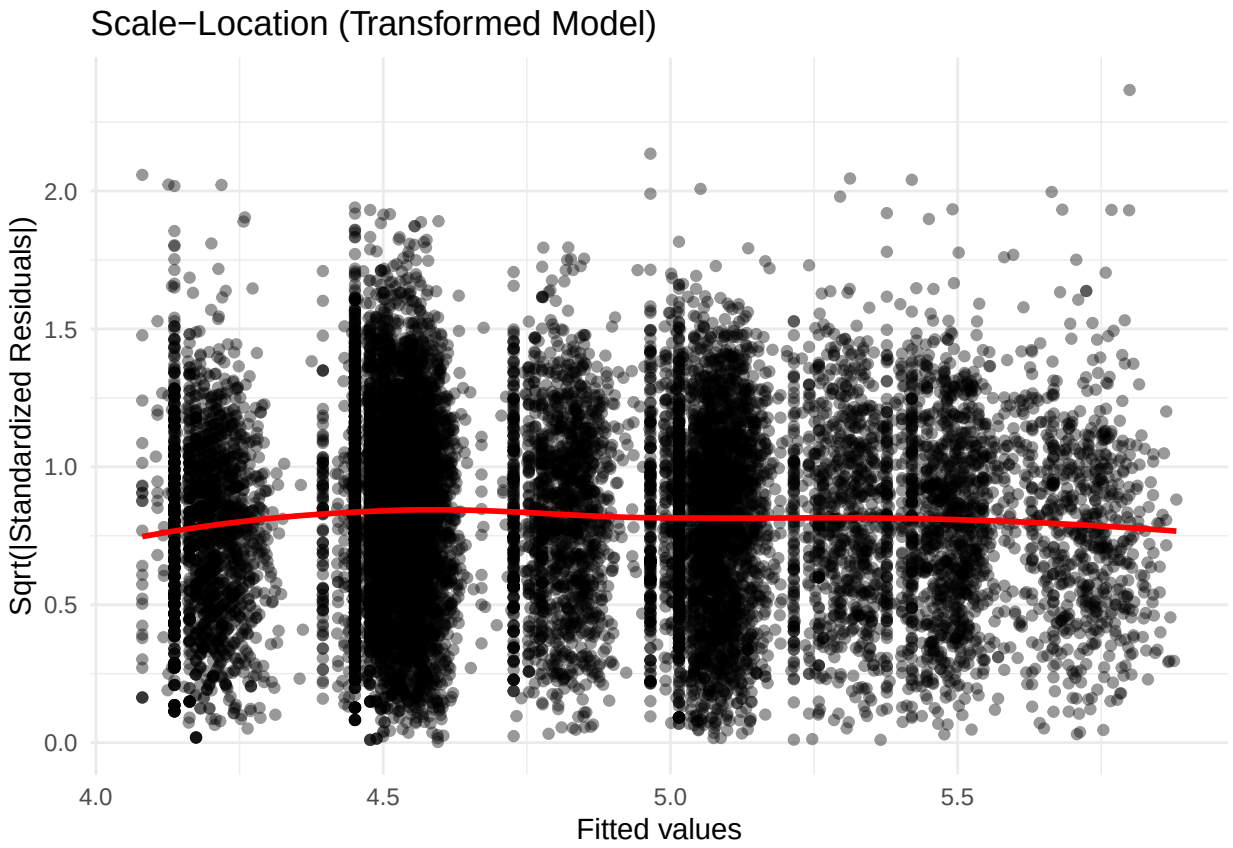


```
ggplot(df, aes(sample = trans_resid)) +
  stat_qq(alpha = 0.4) +
  stat_qq_line(color = "red") +
  labs(title = "Normal Q-Q Plot (Transformed Model)",
       x = "Theoretical Quantiles", y = "Sample Quantiles") +
  theme_minimal()
```



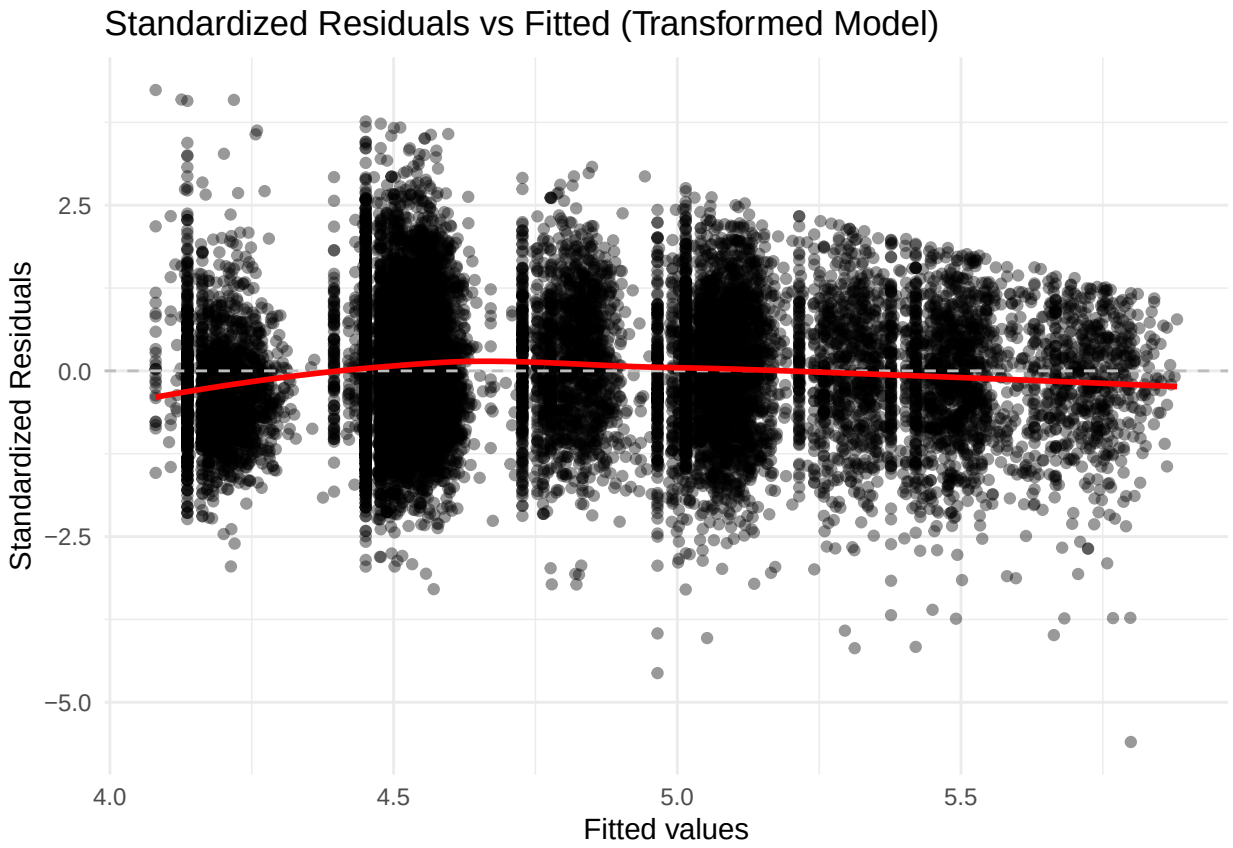

```
ggplot(df, aes(x = trans_fitted, y = sqrt(abs(trans_std_resid)))) +
  geom_point(alpha = 0.4) +
  geom_smooth(method = "loess", color = "red", se = FALSE) +
  labs(title = "Scale-Location (Transformed Model)",
       x = "Fitted values", y = "Sqrt(|Standardized Residuals|)") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
ggplot(df, aes(x = trans_fitted, y = trans_std_resid)) +
  geom_point(alpha = 0.4) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "gray") +
  geom_smooth(method = "loess", color = "red", se = FALSE) +
  labs(title = "Standardized Residuals vs Fitted (Transformed Model)",
       x = "Fitted values", y = "Standardized Residuals") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



#final model preliminary

```
model_final <- lm(
  log(price) ~
    poly(accommodates, 2) +
    poly(bedrooms, 2) +
    log_reviews +
    host_is_superhost +
    review_scores_rating +
    accommodates:bedrooms +
    accommodates:host_is_superhost +
    log_reviews:host_is_superhost +
    I(accommodates / (bedrooms + 1)),
  data = df
)


summary(model_final)
```

```
##
## Call:
## lm(formula = log(price) ~ poly(accommodates, 2) + poly(bedrooms,
##     2) + log_reviews + host_is_superhost + review_scores_rating +
##     accommodates:bedrooms + accommodates:host_is_superhost +
##     log_reviews:host_is_superhost + I(accommodates/(bedrooms +
##     1)), data = df)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -2.86738 -0.34985 -0.02297  0.33579  2.17176
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      4.488379   0.054714  82.034  < 2e-16 ***
## poly(accommodates, 2)1          25.575069   4.455974   5.740 9.67e-09 ***
## poly(accommodates, 2)2         -17.571488   0.776864 -22.618  < 2e-16 ***
## poly(bedrooms, 2)1               6.766458   1.777198   3.807 0.000141 ***
## poly(bedrooms, 2)2              -4.233694   0.979606  -4.322 1.56e-05 ***
## log_reviews                      0.009387   0.004530   2.072 0.038272 *
## host_is_superhost               -0.121901   0.021034  -5.795 6.94e-09 ***
## review_scores_rating             0.007060   0.002960   2.385 0.017076 *
## I(accommodates/(bedrooms + 1))   0.128351   0.029100   4.411 1.04e-05 ***
## accommodates:bedrooms            0.013678   0.003682   3.715 0.000204 ***
## host_is_superhost:accommodates   0.005521   0.004321   1.278 0.201281
## log_reviews:host_is_superhost    0.038045   0.005598   6.797 1.11e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5092 on 15539 degrees of freedom
## Multiple R-squared:  0.4154, Adjusted R-squared:  0.415
## F-statistic:  1004 on 11 and 15539 DF,  p-value: < 2.2e-16
```

```
# Extract diagnostics
df$final_resid <- residuals(model_final)
df$final_fitted <- fitted(model_final)
df$final_std_resid <- rstandard(model_final)


#calculate outliers
df$std_resid <- rstandard(model_final)
outlier_count <- sum(abs(df$std_resid) > 3)
cat("Number of potential outliers (|std_resid| > 3):", outlier_count, "\n")
```

```
## Number of potential outliers (|std_resid| > 3): 75
```

```
df$leverage <- hatvalues(model_final)
n <- nrow(df)
p <- length(coefficients(model_final))  # includes intercept
leverage_threshold <- 2 * p / n


#How many exceeds threshold?
high_lev_count <- sum(df$leverage > leverage_threshold)
cat("Number of high leverage points (h > 2p/n):", high_lev_count, "\n")
```

```
## Number of high leverage points (h > 2p/n): 641
```

```
cat("Threshold used:", leverage_threshold, "\n")
```

```
## Threshold used: 0.001543309
```

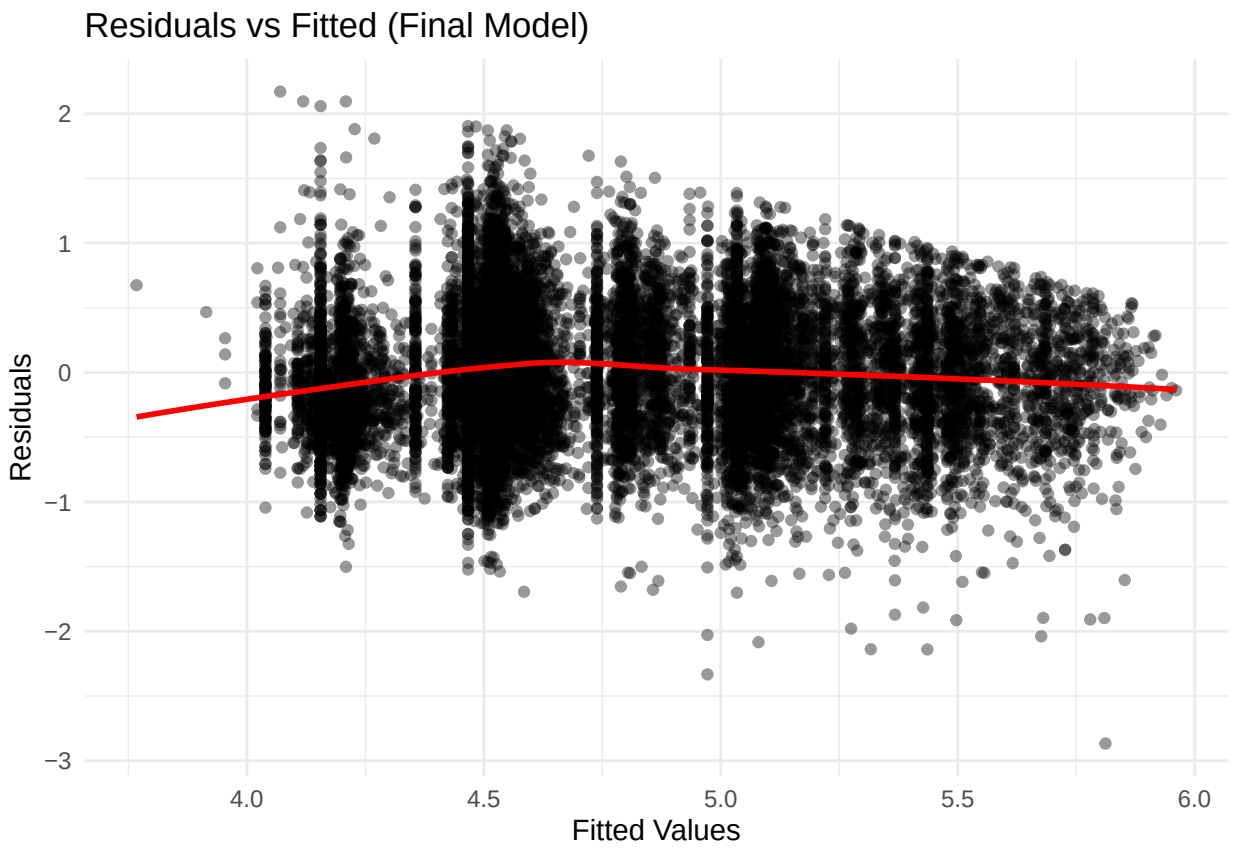
```
df$cooks_d <- cooks.distance(model_final)
n <- nrow(df)
cooks_threshold <- 4 / n
```

```r
# Count number of influential observations
influential_count <- sum(df$cooks_d > cooks_threshold)
cat("Number of influential points (Cook's D > 4/n):", influential_count, "\n")
```

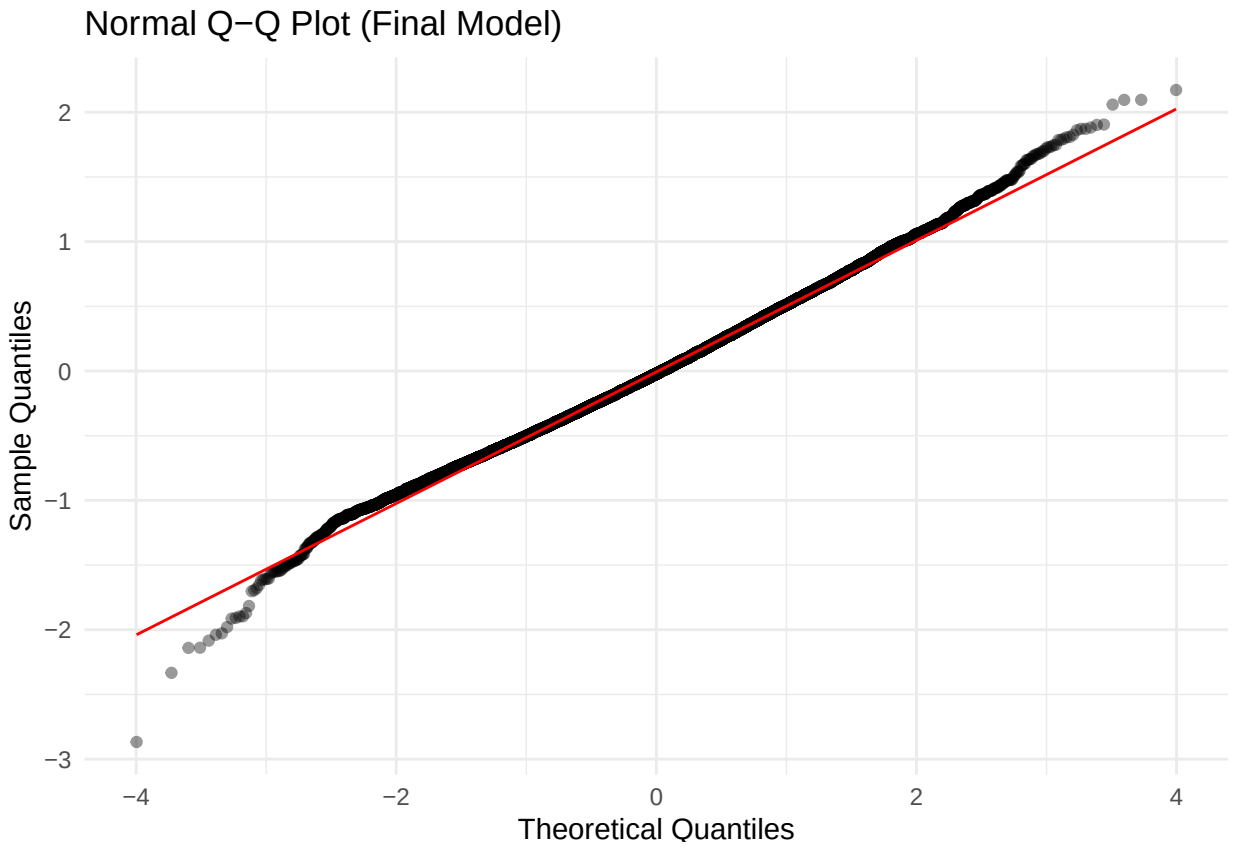
```
## Number of influential points (Cook's D > 4/n): 466
```

```r
# Plot 1: Residuals vs Fitted
ggplot(df, aes(x = final_fitted, y = final_resid)) +
  geom_point(alpha = 0.4) +
  geom_smooth(method = "loess", se = FALSE, color = "red") +
  labs(title = "Residuals vs Fitted (Final Model)",
       x = "Fitted Values", y = "Residuals") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

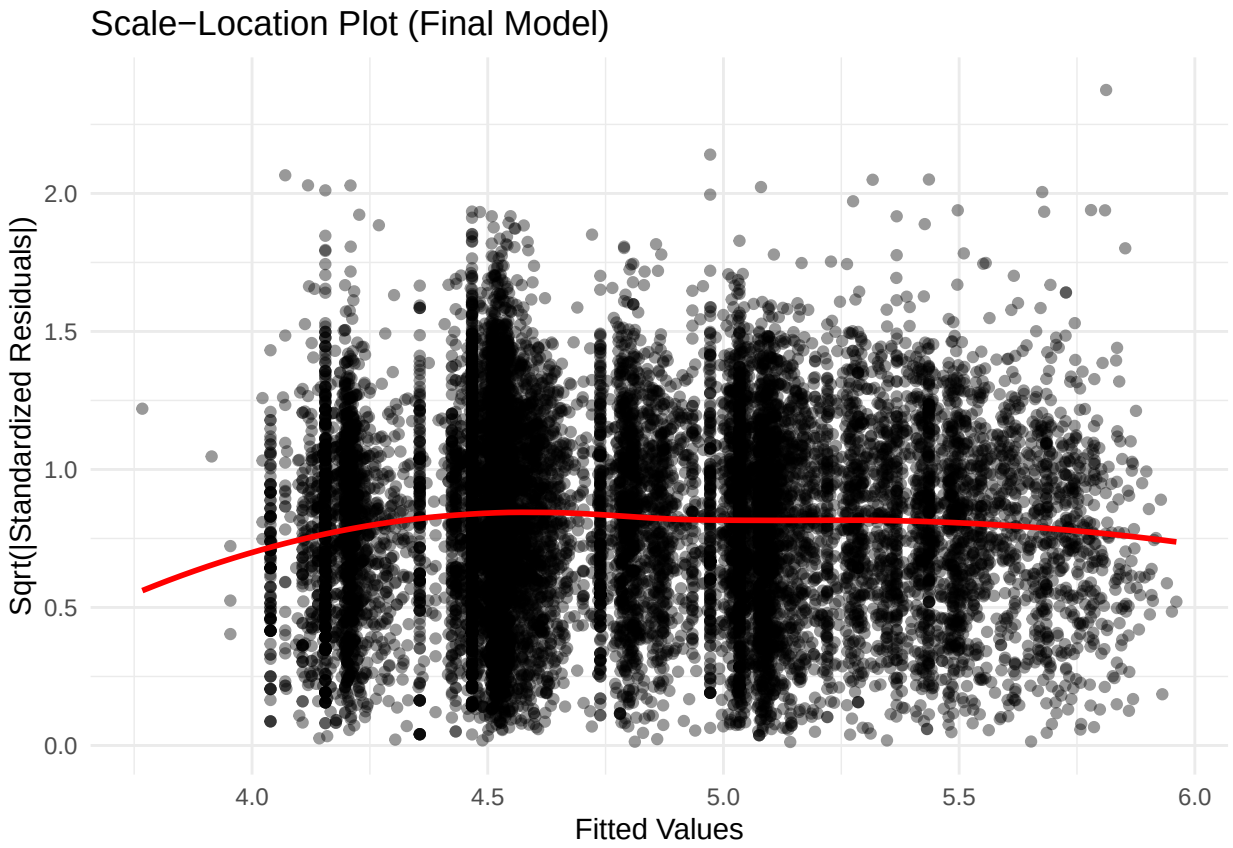



```r
# Plot 2: Normal Q-Q
ggplot(df, aes(sample = final_resid)) +
  stat_qq(alpha = 0.4) +
  stat_qq_line(color = "red") +
  labs(title = "Normal Q-Q Plot (Final Model)",
       x = "Theoretical Quantiles", y = "Sample Quantiles") +
  theme_minimal()
```



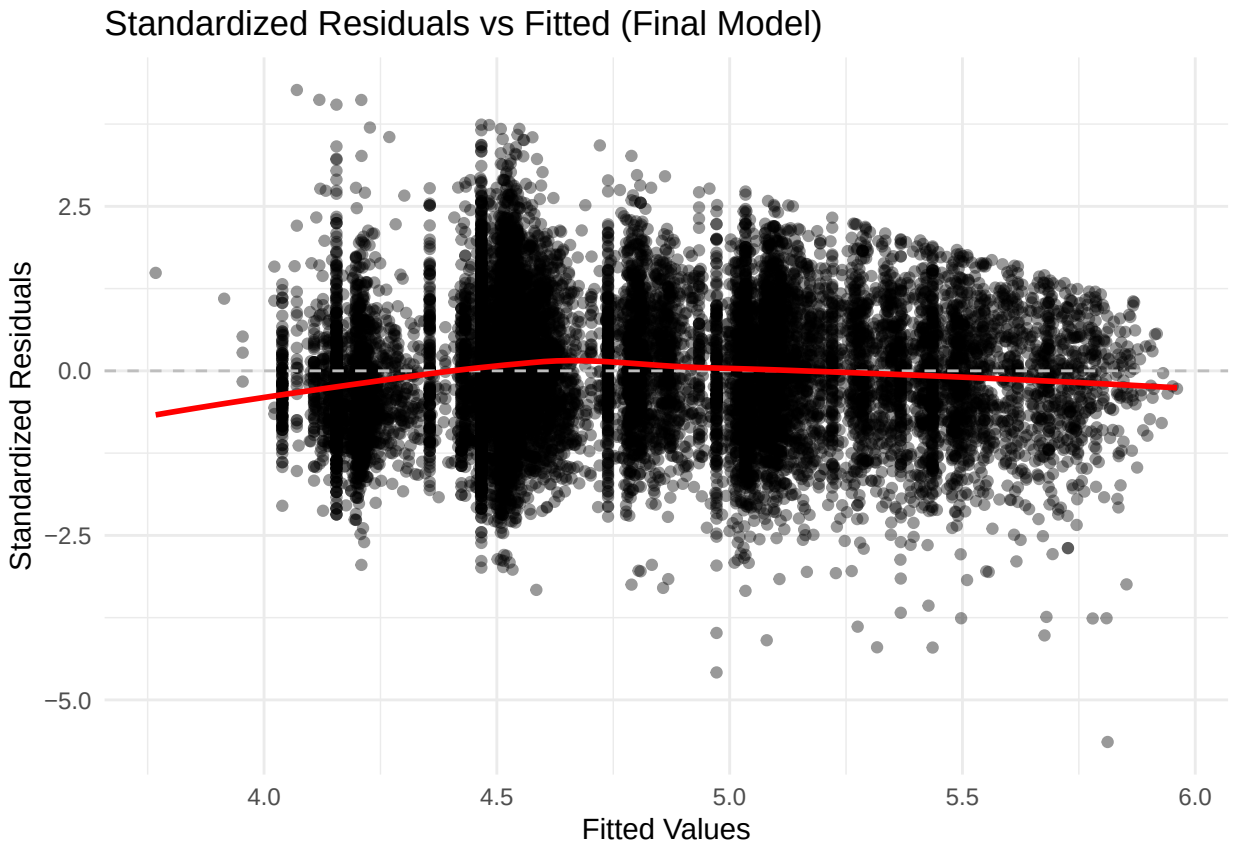

```
# Plot 3: Scale-Location Plot
ggplot(df, aes(x = final_fitted, y = sqrt(abs(final_std_resid)))) +
  geom_point(alpha = 0.4) +
  geom_smooth(method = "loess", se = FALSE, color = "red") +
  labs(title = "Scale-Location Plot (Final Model)",
       x = "Fitted Values", y = "Sqrt(|Standardized Residuals|)") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
# Plot 4: Standardized Residuals vs Fitted
ggplot(df, aes(x = final_fitted, y = final_std_resid)) +
  geom_point(alpha = 0.4) +
  geom_hline(yintercept = 0, color = "gray", linetype = "dashed") +
  geom_smooth(method = "loess", se = FALSE, color = "red") +
  labs(title = "Standardized Residuals vs Fitted (Final Model)",
       x = "Fitted Values", y = "Standardized Residuals") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



#robust regression model

```
model_final_robust <- lmrob(
  log(price) ~
    poly(accommodates, 2) +
    poly(bedrooms, 2) +
    log_reviews +
    host_is_superhost +
    review_scores_rating +
    accommodates:bedrooms +
    log_reviews:host_is_superhost +
    I(accommodates / (bedrooms + 1)),
  data = df
)


summary(model_final_robust)
```

```
##
## Call:
## lmrob(formula = log(price) ~ poly(accommodates, 2) + poly(bedrooms, 2) +
##     log_reviews + host_is_superhost + review_scores_rating + accommodates:bedrooms +
##     log_reviews:host_is_superhost + I(accommodates/(bedrooms + 1)), data = df)
##  \--> method = "MM"
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.80398 -0.33874 -0.01279  0.34061  3.26666
##
```

```
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      4.518490   0.080761  55.949  < 2e-16 ***
## poly(accommodates, 2)1          27.486305   6.463069   4.253 2.12e-05 ***
## poly(accommodates, 2)2         -26.892314   2.784071  -9.659  < 2e-16 ***
## poly(bedrooms, 2)1              -3.075528   3.629857  -0.847 0.396849
## poly(bedrooms, 2)2              -1.783933   1.635207  -1.091 0.275311
## log_reviews                      0.008015   0.004697   1.706 0.087946 .
## host_is_superhost               -0.105759   0.015842  -6.676 2.54e-11 ***
## review_scores_rating             0.008422   0.003012   2.796 0.005180 **
## I(accommodates/(bedrooms + 1))   0.038770   0.046765   0.829 0.407099
## accommodates:bedrooms            0.026042   0.006861   3.796 0.000148 ***
## log_reviews:host_is_superhost    0.039769   0.005487   7.248 4.43e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Robust residual standard error: 0.4995
## Multiple R-squared:  0.444,  Adjusted R-squared:  0.4437
## Convergence in 38 IRWLS iterations
##
## Robustness weights:
##  7 observations c(3180,3595,9292,12843,14318,14396,15384)
##   are outliers with |weight| <= 3.2e-06 ( < 6.4e-06);
##  1275 weights are ~= 1. The remaining 14269 ones are summarized as
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.001736 0.872800 0.949800 0.905300 0.985200 0.999000
## Algorithmic parameters:
##        tuning.chi                bb        tuning.psi        refine.tol
##         1.548e+00         5.000e-01         4.685e+00         1.000e-07
##           rel.tol         scale.tol         solve.tol          zero.tol
##         1.000e-07         1.000e-10         1.000e-07         1.000e-10
##       eps.outlier             eps.x warn.limit.reject warn.limit.meanrw
##         6.430e-06         2.037e-10         5.000e-01         5.000e-01
##      nResample         max.it         groups        n.group       best.r.s
##            500             50              5            400              2
##       k.fast.s          k.max    maxit.scale      trace.lev            mts
##              1            200            200              0           1000
##     compute.rd fast.s.large.n
##              0           2000
##                  psi           subsampling                   cov
##            "bisquare"         "nonsingular"         ".vcov.avar1"
## compute.outlier.stats
##                  "SM"
## seed : int(0)
```

```r
# Extract diagnostics
df$final_resid_r <- residuals(model_final_robust)
df$final_fitted_r <- fitted(model_final_robust)
df$final_std_resid_r <- residuals(model_final_robust) /
  summary(model_final_robust)$scale


df$robust_weight <- model_final_robust$rweights
df$is_downweighted <- df$robust_weight < 0.1
```

```r
# Count how many were heavily downweighted
sum(df$is_downweighted)  # This is the number of points lmrob treats as outliers
```

```
## [1] 24
```

```r
summary(model_final_robust)$r.squared
```

```
## [1] 0.4440129
```

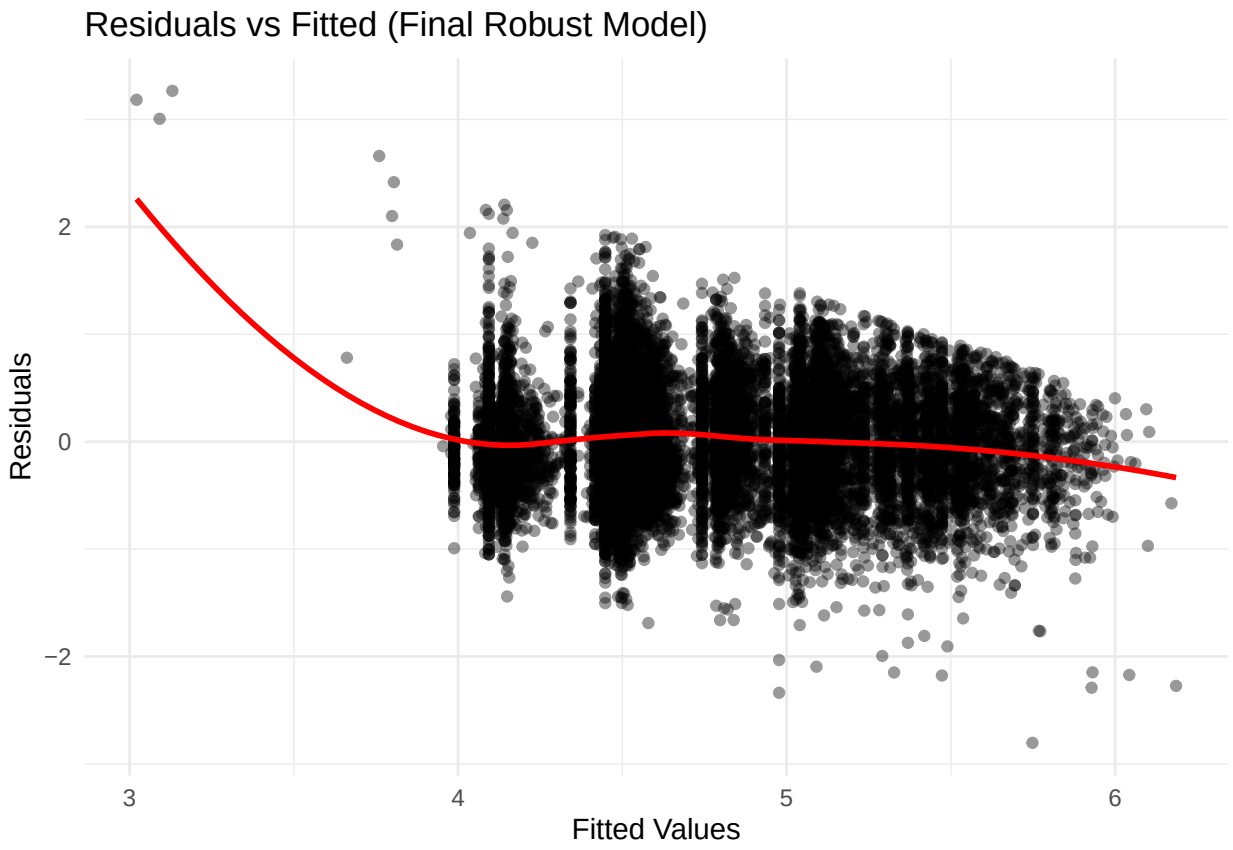
```r
# Number of observations used in the model (excluding NA rows)
# Custom function to compute AIC and BIC
# Custom function to compute AIC and BIC
get_custom_aic_bic <- function(model) {
  n <- length(residuals(model))
  p <- length(coef(model))  # include intercept
  rss <- sum(residuals(model)^2)

  aic <- n * log(rss / n) + 2 * p
  bic <- n * log(rss / n) + p * log(n)

  return(c(AIC = aic, BIC = bic))
}


# Compute metrics and store in a data frame
results <- data.frame(
  Model = c("model", "model_transformed", "model_final", '"model_final_robust"'),
  AIC = c(
    get_custom_aic_bic(model)[["AIC"]],
    get_custom_aic_bic(model_transformed)[["AIC"]],
    get_custom_aic_bic(model_final)[["AIC"]],
    get_custom_aic_bic(model_final_robust)[["AIC"]]
  ),
  BIC = c(
    get_custom_aic_bic(model)[["BIC"]],
    get_custom_aic_bic(model_transformed)[["BIC"]],
    get_custom_aic_bic(model_final)[["BIC"]],
    get_custom_aic_bic(model_final_robust)[["BIC"]]
  ),
  R_squared = c(
    summary(model)$r.squared,
    summary(model_transformed)$r.squared,
    summary(model_final)$r.squared,
    summary(model_final_robust)$r.squared
  ),
  Adj_R_squared = c(
    summary(model)$adj.r.squared,
    summary(model_transformed)$adj.r.squared,
    summary(model_final)$adj.r.squared,
    summary(model_final_robust)$adj.r.squared
  )
)
print(results)
```

```
##                   Model       AIC       BIC R_squared Adj_R_squared
```

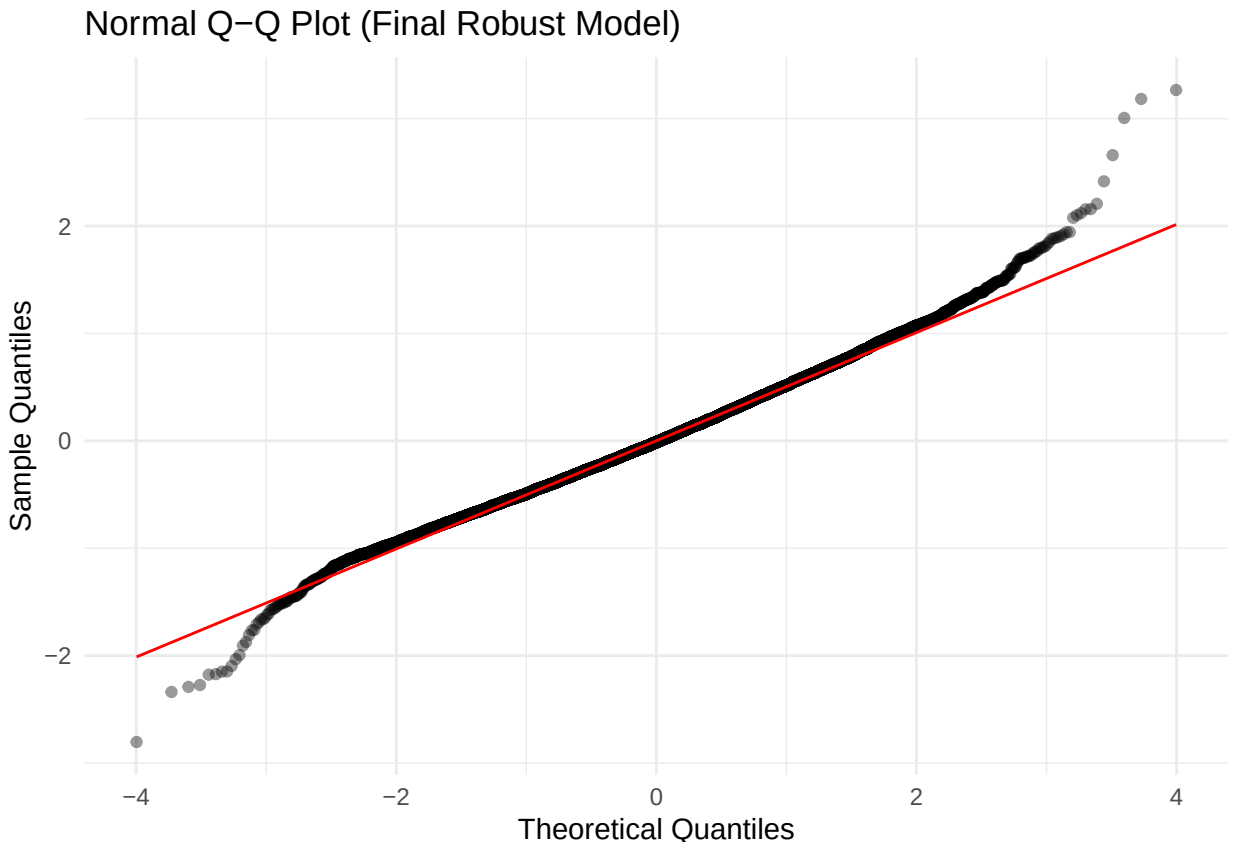
```
## 1                model 137880.29 137926.20 0.3608676     0.3606620
## 2    model_transformed -20919.40 -20865.84 0.4127301     0.4125034
## 3          model_final -20979.44 -20887.62 0.4153691     0.4149553
## 4 "model_final_robust" -20810.76 -20726.59 0.4440129     0.4436551
```

```r
# Plot 1: Residuals vs Fitted
ggplot(df, aes(x = final_fitted_r, y = final_resid_r)) +
  geom_point(alpha = 0.4) +
  geom_smooth(method = "loess", se = FALSE, color = "red") +
  labs(title = "Residuals vs Fitted (Final Robust Model)",
       x = "Fitted Values", y = "Residuals") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

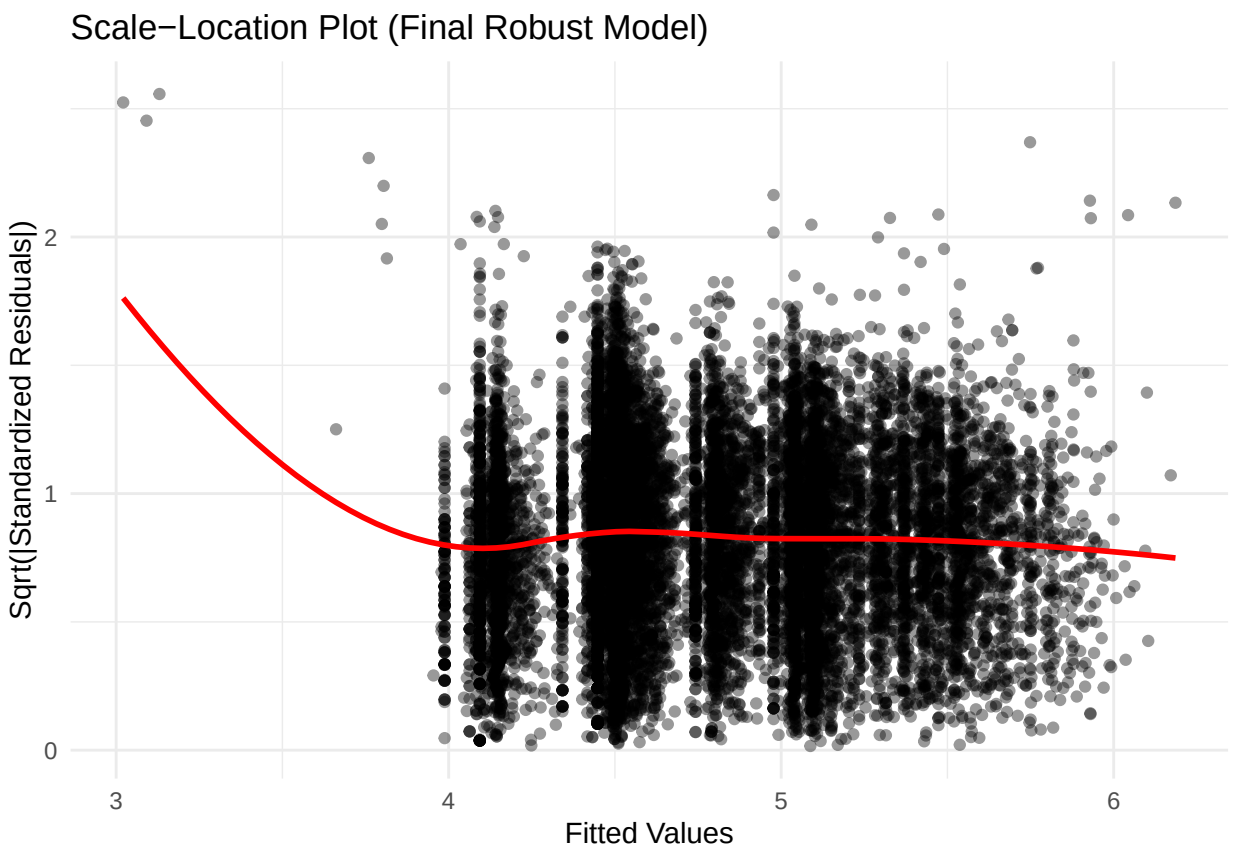



```r
# Plot 2: Normal Q-Q
ggplot(df, aes(sample = final_resid_r)) +
  stat_qq(alpha = 0.4) +
  stat_qq_line(color = "red") +
  labs(title = "Normal Q-Q Plot (Final Robust Model)",
       x = "Theoretical Quantiles", y = "Sample Quantiles") +
  theme_minimal()
```



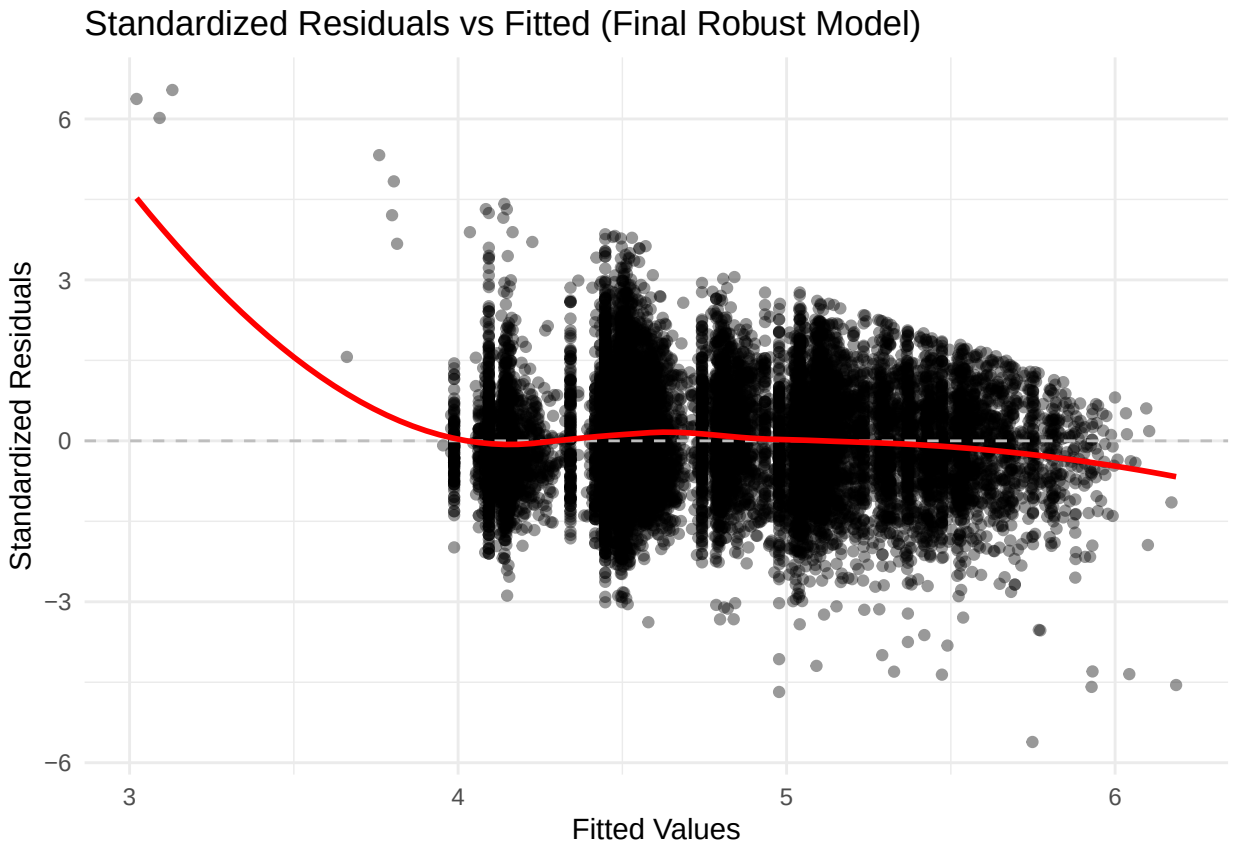

```
# Plot 3: Scale-Location Plot
ggplot(df, aes(x = final_fitted_r, y = sqrt(abs(final_std_resid_r)))) +
  geom_point(alpha = 0.4) +
  geom_smooth(method = "loess", se = FALSE, color = "red") +
  labs(title = "Scale-Location Plot (Final Robust Model)",
       x = "Fitted Values", y = "Sqrt(|Standardized Residuals|)") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
# Plot 4: Standardized Residuals vs Fitted
ggplot(df, aes(x = final_fitted_r, y = final_std_resid_r)) +
  geom_point(alpha = 0.4) +
  geom_hline(yintercept = 0, color = "gray", linetype = "dashed") +
  geom_smooth(method = "loess", se = FALSE, color = "red") +
  labs(title = "Standardized Residuals vs Fitted (Final Robust Model)",
       x = "Fitted Values", y = "Standardized Residuals") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```


Standardized Residuals vs Fitted (Final Robust Model)
Standardized Residuals
Fitted Values
6
3
0
−3
−6
3
4
5
6