AirBnB Project Analysis

2025-05-21

```
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(readr)
```

```
df <- read_csv("listings-2.csv")
```

```
## Rows: 21660 Columns: 79
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (25): listing_url, source, name, description, neighborhood_overview, pi...
## dbl  (42): id, scrape_id, host_id, host_listings_count, host_total_listings_...
## lgl   (7): host_is_superhost, host_has_profile_pic, host_identity_verified, ...
## date  (5): last_scraped, host_since, calendar_last_scraped, first_review, la...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
df <- df %>% filter(!is.na(price))
df$price <- as.numeric(gsub("[$,]", "", df$price))
```

```
df$bedrooms <- ifelse(
  is.na(df$bedrooms) & df$beds >= 1, 1,
  ifelse(is.na(df$bedrooms), 0, df$bedrooms)
)
```

```
df$review_scores_rating[is.na(df$review_scores_rating) & df$number_of_reviews == 0] <- 0
```

```
df$host_is_superhost[is.na(df$host_is_superhost)] <- "f"
```

```
df$host_is_superhost <- ifelse(df$host_is_superhost == "t", 1, 0)
```

```
df <- df %>% select(price, accommodates, bedrooms, number_of_reviews, host_is_superhost, review_scores_
```

```
price_cap <- quantile(df$price, 0.99)
df <- df %>% filter(price > 0, price < price_cap)
```

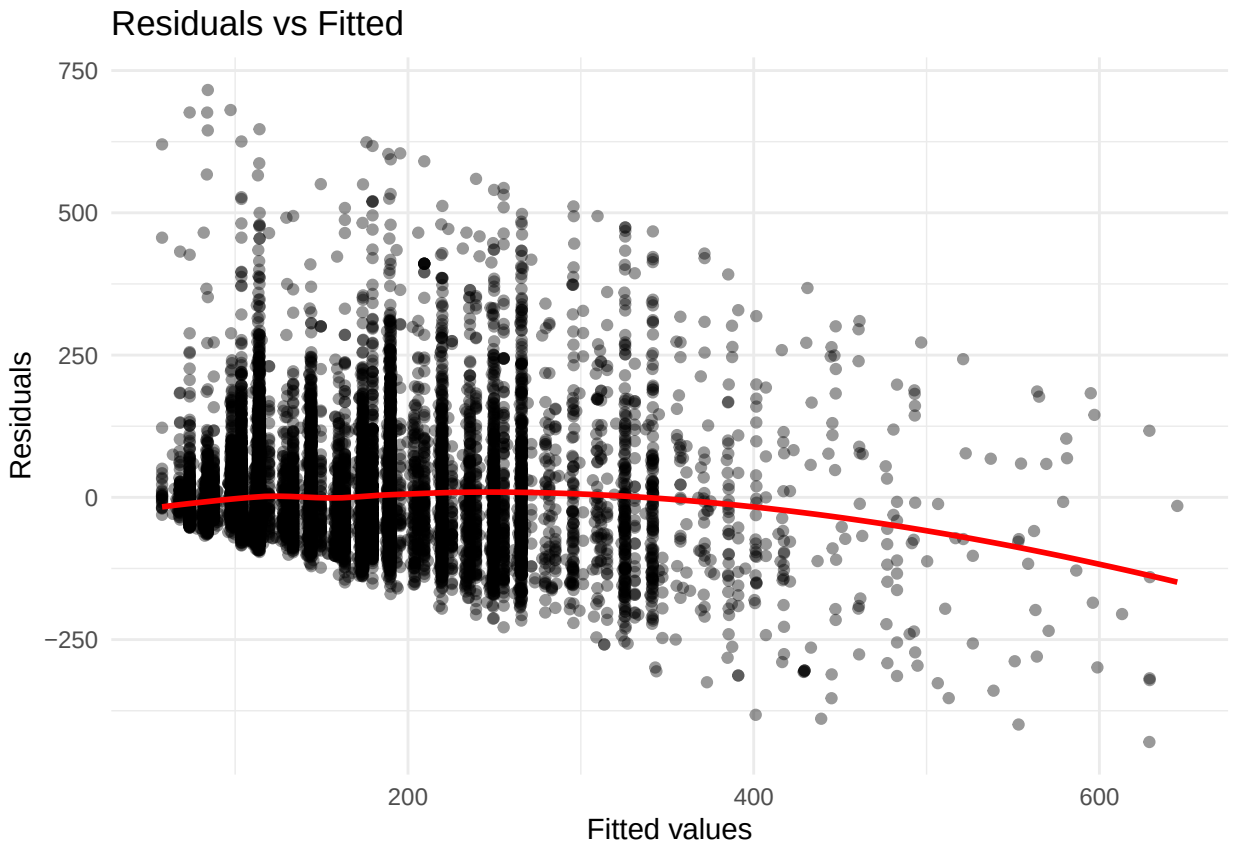
```
write_csv(df, "cleaned_listings.csv")
```

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

df <- read_csv("cleaned_listings.csv")
```

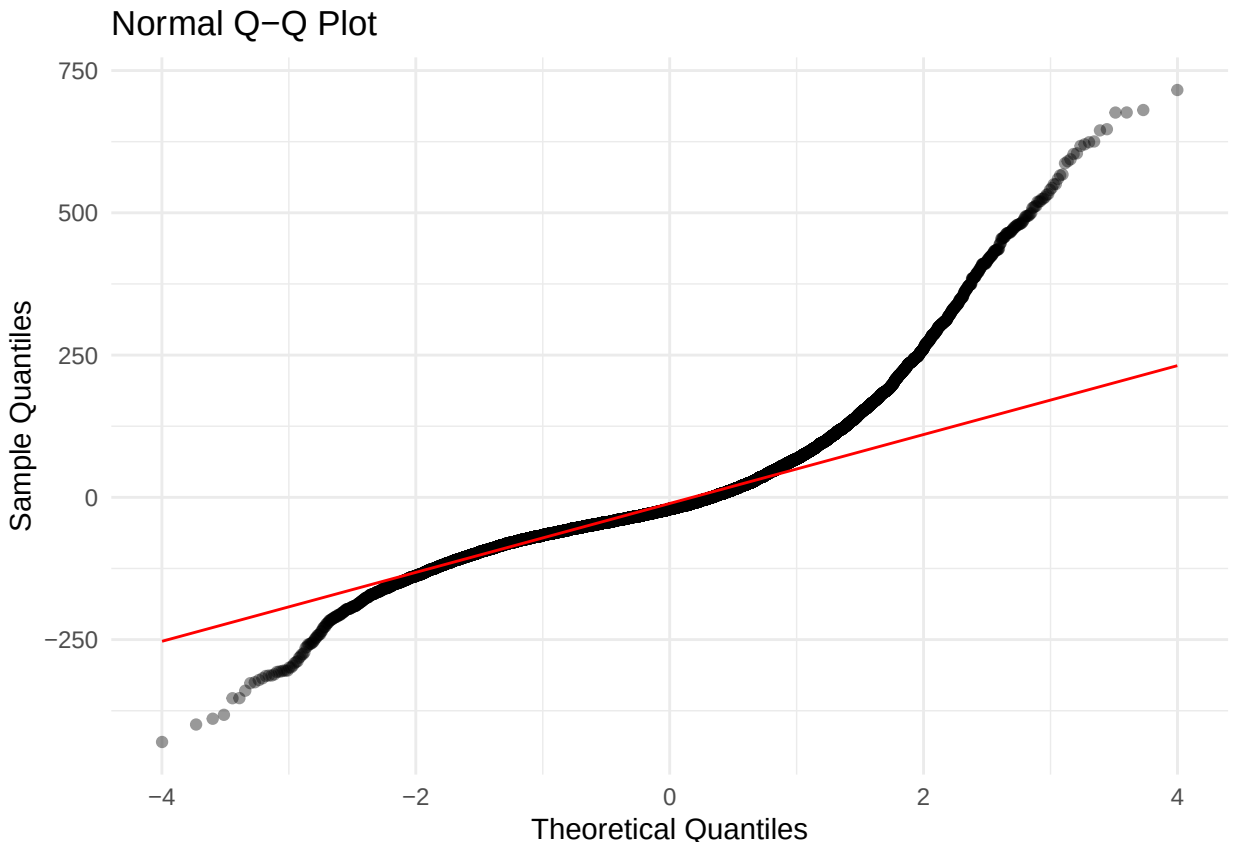
```
## Rows: 15709 Columns: 6
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (6): price, accommodates, bedrooms, number_of_reviews, host_is_superhost...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
model <- lm(price ~ accommodates + bedrooms + number_of_reviews +
              review_scores_rating + host_is_superhost, data = df)

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'
```

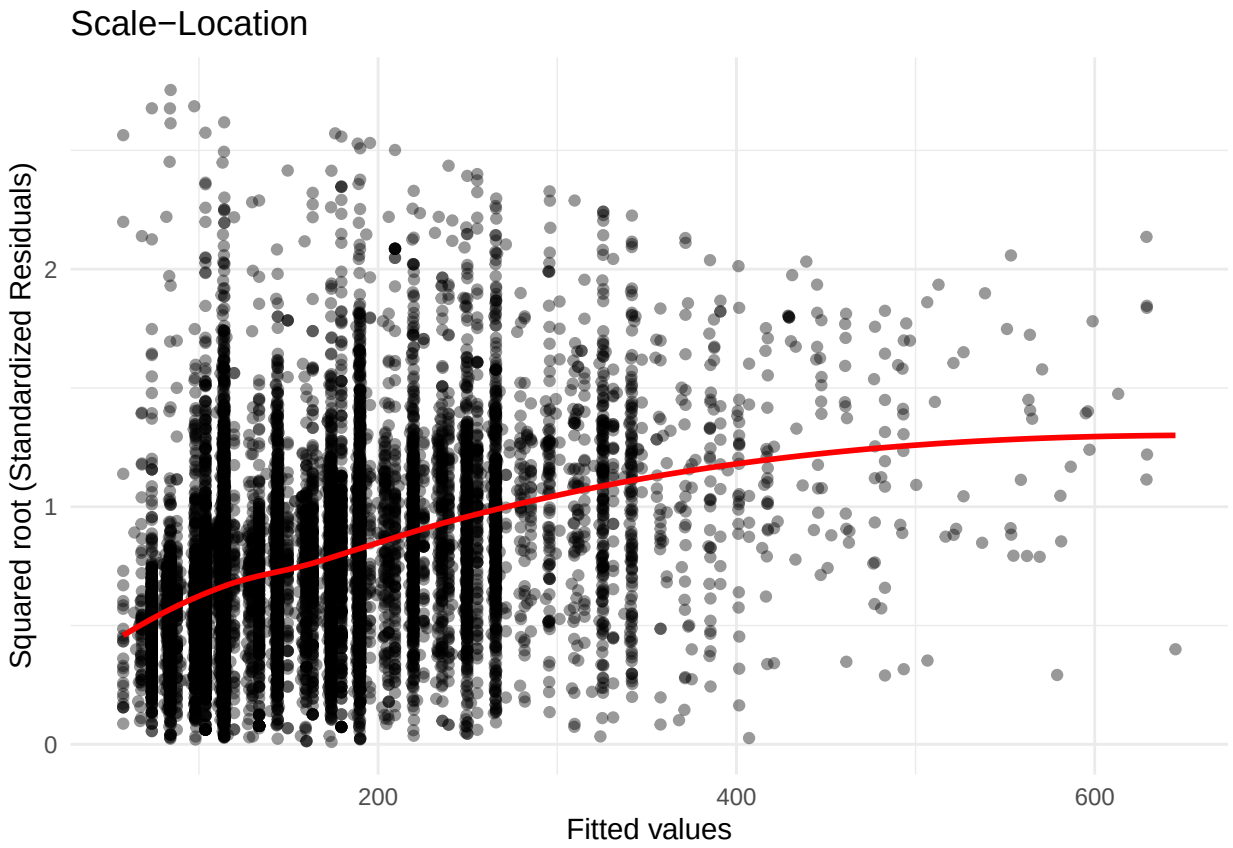



```r
ggplot(df, aes(sample = residuals)) +
  stat_qq(alpha = 0.4) +
  stat_qq_line(color = "red") +
  labs(title = "Normal Q-Q Plot", x = "Theoretical Quantiles", y = "Sample Quantiles") +
  theme_minimal()
```



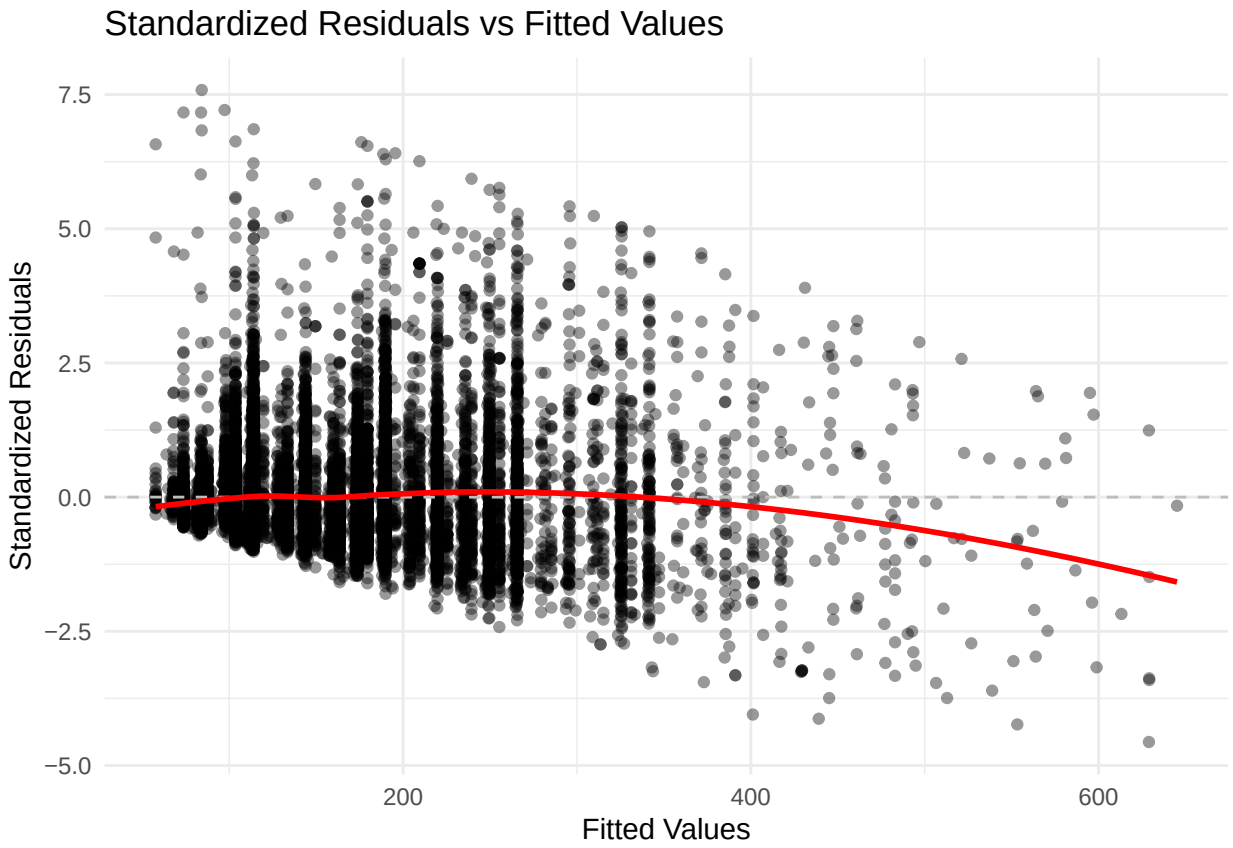

```
ggplot(df, aes(x = fitted, y = sqrt(abs(std_resid)))) +
  geom_point(alpha = 0.4) +
  geom_smooth(method = "loess", color = "red", se = FALSE) +
  labs(title = "Scale-Location", x = "Fitted values", y = "Squared root (Standardized Residuals)") +
  theme_minimal()
```

```
## `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'
```




```
# Load necessary libraries
library(readr)
library(dplyr)

# Read the cleaned dataset
cleaned_data <- read_csv("cleaned_listings.csv")
```

```
## Rows: 15709 Columns: 6
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (6): price, accommodates, bedrooms, number_of_reviews, host_is_superhost...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# Fit the linear model
model <- lm(price ~ accommodates + bedrooms + number_of_reviews +
              host_is_superhost + review_scores_rating,
            data = cleaned_data)

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

```
##
## Call:
## lm(formula = price ~ accommodates + bedrooms + number_of_reviews +
##     host_is_superhost + review_scores_rating, data = cleaned_data)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -429.92  -51.64  -20.63   30.05  715.81
##
## Coefficients: (1 not defined because of singularities)
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          27.782473   2.032905  13.666  < 2e-16 ***
## accommodates         29.930776   0.592350  50.529  < 2e-16 ***
## bedrooms             15.995078   1.330860  12.019  < 2e-16 ***
## number_of_reviews     0.002548   0.012825   0.199    0.843
## host_is_superhost           NA         NA      NA       NA
## review_scores_rating  2.089446   0.388847   5.373 7.84e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 94.38 on 15704 degrees of freedom
## Multiple R-squared:  0.3656, Adjusted R-squared:  0.3655
## F-statistic:  2263 on 4 and 15704 DF,  p-value: < 2.2e-16
```