Báo cáo đồ án cuối kỳ - Project 3

2024-07-05

1. Giới thiệu chung

Bối cảnh và mục tiêu

Trong bối cảnh một thành phố đông đúc với nhu cầu thuê căn hộ ngày càng tăng cao, việc định giá căn hộ trở nên thách thức đối với những người cho thuê và môi giới bất động sản. Dữ liệu chứa các thông tin về giá thuê, diện tích, số phòng ngủ, số phòng tắm và nhiều yếu tố khác liên quan đến các căn hộ trong thành phố. Mục tiêu của phân tích này là xây dựng một mô hình dự đoán giá thuê căn hộ chính xác, giúp người cho thuê và môi giới định giá căn hộ một cách hợp lý, phù hợp với nhu cầu thị trường.

Đề xuất các phương pháp xử lý dữ liệu

Phân tích thống kê và trực quan hóa dữ liệu

- Phân tích Exploratory Data Analysis (EDA): Thực hiện EDA để khám phá các mẫu, xu hướng và mối quan hệ trong dữ liệu thông qua các biểu đồ như scatter plots, box plots, và pair plots.
- Phân phối chuẩn hóa và biểu đồ histogram: Sử dụng biểu đồ histogram và phân phối chuẩn hóa để hiểu rõ hơn về tính phân phối của các biến số và phát hiện các bất thường.
- Phân tích tương quan: Sử dụng ma trận tương quan và biểu đồ scatter plot để xác định các biến quan trọng và ảnh hưởng của chúng đến mô hình dự đoán giá thuê căn hộ.

Tiền xử lý dữ liệu

Xử lý giá trị thiếu (Missing Value Imputation):

- Điền giá trị thiếu: Sử dụng phương pháp điền giá trị trung bình, trung vị hoặc mode của biến tương ứng để điền vào các giá trị thiếu trong dữ liệu.
- Loại bỏ mẫu dữ liệu chứa giá trị thiếu: Nếu tỷ lệ thiếu không quá cao và không ảnh hưởng nhiều đến tính chính xác của mô hình, có thể xem xét loại bỏ các mẫu dữ liệu chứa giá trị thiếu.

Chuyển đổi định dạng dữ liệu (Data Transformation):

- Biến chuỗi thành biến số (String to Numeric): Chuyển đổi các biến chuỗi (như số lượng phòng ngủ, phòng tắm) thành các biến numeric để thuận tiện cho việc xử lý và phân tích.
- Biến thời gian thành dạng datetime (Time Conversion): Chuyển đổi biến thời gian thành dạng datetime để dễ dàng thực hiện các phân tích thời gian. Biến đổi log (Log-Transformation):
- Áp dụng log-transform để cải thiện phân phối của dữ liệu và giảm thiểu ảnh hưởng của các giá trị cực đoan, đặc biệt là đối với các biến có phân phối lệch. Tạo biến giả (Dummy Variables):

- Tạo các biến giả cho các biến phân loại như thành phố, tiện ích có/không (ví dụ: có bể bơi, có phòng gym), để biến chúng thành các biến số có thể sử dụng trong mô hình hồi quy.

Chuẩn hóa dữ liệu (Data Normalization): Sử dụng kỹ thuật chuẩn hóa (như Min-Max Scaling hoặc Z-score Standardization) để đảm bảo các biến số nằm trong cùng một phạm vi giá trị, giúp mô hình học tốt hơn và giảm thiểu bias.

Xử lý giá trị ngoại lai (Outlier Treatment): Xác định và xử lý các giá trị ngoại lai bằng cách sử dụng các phương pháp như IQR (Interquartile Range) hoặc phân tích biểu đồ boxplot để loại bỏ hoặc điều chỉnh các giá trị ngoại lai.

Xây dựng và đánh giá mô hình

- Chia dữ liệu (Train-Test Split): Chia dữ liệu thành tập huấn luyện và tập kiểm tra độc lập với tỷ lệ 70-30 hoặc 80-20 để đảm bảo tính đại diện và khả năng tổng quát hóa của mô hình.
- Cross-validation: Áp dụng k-fold cross-validation để giảm thiểu bias và variance của mô hình, đồng thời cải thiện tính khả thi và dự đoán của nó trên dữ liệu mới.
- Đánh giá hiệu suất mô hình: Sử dụng các chỉ số đánh giá như Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) và R-squared để đánh giá hiệu suất của mô hình.

Kiểm tra và tối ưu hóa mô hình

- Kiểm tra giả định của mô hình hồi quy: Đảm bảo tính tuyến tính, tính đồng nhất phương sai và phân phối chuẩn của phần dư để đảm bảo tính hợp lệ của kết quả.
- Loại bỏ giá trị ngoại lai (Outlier Removal): Phân tích biểu đồ thặng dư và Cook's distance để xác định và loại bỏ các điểm dữ liệu ngoại lai có ảnh hưởng lớn đến mô hình.
- Mở rộng mô hình hồi quy (hồi quy đa thức, hồi quy splines, GAM,...) giúp cải thiện độ chính xác của mô hình.

Kết quả và ứng dụng

- Mô hình dự đoán giá thuê căn hộ: Cung cấp dự đoán chính xác, hỗ trợ quyết định kinh doanh và tối ưu hóa lợi nhuận từ việc cho thuê căn hộ.
- Insights về thị trường: Phân tích phân phối và mối quan hệ của các biến chính như giá thuê, diện tích, số phòng ngủ và số phòng tắm, cung cấp những thông tin quan trọng về thị trường.

Các mục tiêu cần đạt được:

1. Xử lý dữ liệu hiệu quả: Mục tiêu của bước này là đảm bảo dữ liệu được chuẩn bị sạch sẽ và hoàn chỉnh.
2. Biến đổi dữ liệu để cải thiện phân phối: Phương pháp này giúp giảm thiểu ảnh hưởng của các giá trị cực đoan và đảm bảo các biến số có phân phối gần như chuẩn, điều kiện cần cho việc áp dụng mô hình hồi quy tuyến tính.
3. Xử lý các biến phân loại: Chúng giúp mô hình hồi quy tuyến tính hiểu được các biến phân loại và tích hợp chúng vào quá trình dự đoán giá thuê căn hộ.

4. Phân tích và loại bỏ giá trị ngoại lai: Mục tiêu là phân tích và loại bỏ các giá trị ngoại lai có thể ảnh hưởng đáng kể đến mô hình dự đoán. Phương pháp thường sử dụng bao gồm phân tích biểu đồ thặng dư để đánh giá tính tuyến tính từng phần, tính đồng nhất phương sai và phân tích biểu đồ Cook's distance để xác định và loại bỏ các điểm dữ liệu ngoại lai.

5. Phân tích tương quan và quan hệ giữa các biến: Mục tiêu của bước này là phân tích tương quan giữa các biến chính như giá thuê, diện tích, số phòng ngủ và số phòng tắm. Điều này giúp hiểu rõ hơn về mối quan hệ giữa các biến và xác định các biến quan trọng đối với mô hình dự đoán giá thuê căn hộ.

6. Xây dựng và đánh giá mô hình dự đoán: Mục tiêu là xây dựng một mô hình hồi quy tuyến tính chính xác để dự đoán giá thuê căn hộ. Sau đó, đánh giá hiệu suất của mô hình trên tập dữ liệu kiểm tra để đảm bảo tính khả thi và đáng tin cậy của dự đoán.

7. Tối ưu hóa mô hình và đưa ra dự đoán chính xác: Mục tiêu là tối ưu hóa mô hình để cung cấp dự đoán chính xác về giá thuê căn hộ. Điều này hỗ trợ quyết định kinh doanh trong việc định giá căn hộ và tối ưu hóa lợi nhuận từ hoạt động cho thuê.

8. Cung cấp những hiểu biết quan trọng về thị trường: Mục tiêu cuối cùng là phân tích phân phối và mối quan hệ của các biến để cung cấp những thông tin chi tiết và quan trọng về thị trường thuê căn hộ. Điều này giúp các đối tượng liên quan có cái nhìn rõ ràng hơn về nhu cầu và xu hướng thị trường, từ đó đưa ra các chiến lược kinh doanh hiệu quả.

Việc xử lý và phân tích dữ liệu một cách khoa học sẽ giúp xây dựng mô hình dự đoán giá thuê căn hộ chính xác và hiệu quả. Mô hình này sẽ hỗ trợ người cho thuê và môi giới định giá căn hộ hợp lý, đáp ứng nhu cầu thị trường, và tối ưu hóa lợi nhuận từ hoạt động kinh doanh bất động sản. Các phương pháp xử lý dữ liệu được đề xuất sẽ đảm bảo rằng dữ liệu đầu vào được chuẩn bị kỹ lưỡng và phù hợp để xây dựng các mô hình dự đoán chính xác và tin cậy.

2. Exploratory Data Analysis (EDA)

2.1 Tải thư viện và đọc dữ liệu

```
library(tidyverse)
library(ggplot2)
library(janitor)
library(dplyr)
library(leaps)
library(caret)
library(glmnet)
library(gridExtra)

data <- read.csv('D:/XLSLTK/Final Project/dataset/apartments_for_rent_classified_10K.csv',
                 encoding = 'latin1', sep = ';', na = c("", "NA", "N/A"))
dim(data)
```

2.2 Thống kê mô tả cho các biến số

```
glimpse(data)
```

```
## Rows: 10,000
## Columns: 22
## $ id            <dbl> 5668626895, 5664597177, 5668626833, 5659918074, 56686267~
## $ category      <chr> "housing/rent/apartment", "housing/rent/apartment", "hou~
## $ title         <chr> "Studio apartment 2nd St NE, Uhland Terrace NE, Washingt~
## $ body          <chr> "This unit is located at second St NE, Uhland Terrace NE~
## $ amenities     <chr> "null", "null", "null", "null", "null", "Dishwasher,Elev~
## $ bathrooms     <chr> "null", "null", "1", "1", "null", "1", "null", "null", "~
## $ bedrooms      <chr> "0", "1", "0", "0", "0", "0", "0", "0", "0", "0", "0", "~
## $ currency      <chr> "USD", "USD", "USD", "USD", "USD", "USD", "USD", "USD", ~
## $ fee           <chr> "No", "No", "No", "No", "No", "No", "No", "No", "No", "N~
## $ has_photo     <chr> "Thumbnail", "Thumbnail", "Thumbnail", "Thumbnail", "Thu~
## $ pets_allowed  <chr> "None", "None", "None", "None", "None", "null", "None", ~
## $ price         <int> 790, 425, 1390, 925, 880, 2475, 1800, 840, 1495, 890, 99~
## $ price_display <chr> "$790", "$425", "$1,390", "$925", "$880", "$2,475", "$1,~
## $ price_type    <chr> "Monthly", "Monthly", "Monthly", "Monthly", "Monthly", "~
## $ square_feet   <int> 101, 106, 107, 116, 125, 130, 132, 136, 138, 141, 146, 1~
## $ address       <chr> "null", "814 Schutte Rd", "null", "1717 12th Avenue", "n~
## $ cityname      <chr> "Washington", "Evansville", "Arlington", "Seattle", "Arl~
## $ state         <chr> "DC", "IN", "VA", "WA", "VA", "NY", "CA", "DC", "CA", "D~
## $ latitude      <chr> "38.9057", "37.9680", "38.8910", "47.6160", "38.8738", "~
## $ longitude     <chr> "-76.9861", "-87.6621", "-77.0816", "-122.3275", "-77.10~
## $ source        <chr> "RentLingo", "RentLingo", "RentLingo", "RentLingo", "Ren~
## $ time          <int> 1577359415, 1577017063, 1577359410, 1576667743, 15773594~
```

```
summary(data)
```

```
##        id              category            title               body          
##  Min.   :5.509e+09   Length:10000       Length:10000       Length:10000      
##  1st Qu.:5.509e+09   Class :character   Class :character   Class :character  
##  Median :5.669e+09   Mode  :character   Mode  :character   Mode  :character  
##  Mean   :5.623e+09                                                           
##  3rd Qu.:5.669e+09                                                           
##  Max.   :5.669e+09                                                           
##   amenities          bathrooms           bedrooms           currency        
##  Length:10000       Length:10000       Length:10000       Length:10000      
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##                                                                             
##                                                                             
##                                                                             
##      fee             has_photo          pets_allowed           price       
##  Length:10000       Length:10000       Length:10000       Min.   :  200  
##  Class :character   Class :character   Class :character   1st Qu.:  949  
##  Mode  :character   Mode  :character   Mode  :character   Median : 1270  
##                                                           Mean   : 1486  
##                                                           3rd Qu.: 1695  
##                                                           Max.   :52500  
##  price_display       price_type         square_feet        address         
##  Length:10000       Length:10000       Min.   :  101.0   Length:10000      
##  Class :character   Class :character   1st Qu.:  649.0   Class :character  
##  Mode  :character   Mode  :character   Median :  802.0   Mode  :character  
##                                        Mean   :  945.8                     
##                                        3rd Qu.: 1100.0
```

```
##                                       Max.   :40000.0
##    cityname            state              latitude          longitude
##  Length:10000       Length:10000       Length:10000       Length:10000
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##     source               time
##  Length:10000       Min.   :1.569e+09
##  Class :character   1st Qu.:1.569e+09
##  Mode  :character   Median :1.577e+09
##                     Mean   :1.575e+09
##                     3rd Qu.:1.577e+09
##                     Max.   :1.577e+09
```

Quán sát về các thông tin cơ bản về dữ liệu, ta có thế thấy:

- Một số biến có nhiều giá trị thiếu (ví dụ: amenities, bathrooms, address).
- Một số biến cần chuyển đổi sang kiểu số (ví dụ: bathrooms, bedrooms, latitude, longitude).
- Một số biến có giá trị cố định hoặc ít biến đổi (ví dụ: currency, price_type, source).
- Biến address có nhiều giá trị "null", có thể cần xem xét để xử lý hoặc loại bỏ.
- Các biến như title, body có thể chứa thông tin chi tiết bổ sung nhưng cần xử lý dạng văn bản nếu sử dụng. ## 2.3 Kiểm tra dữ liệu thiếu

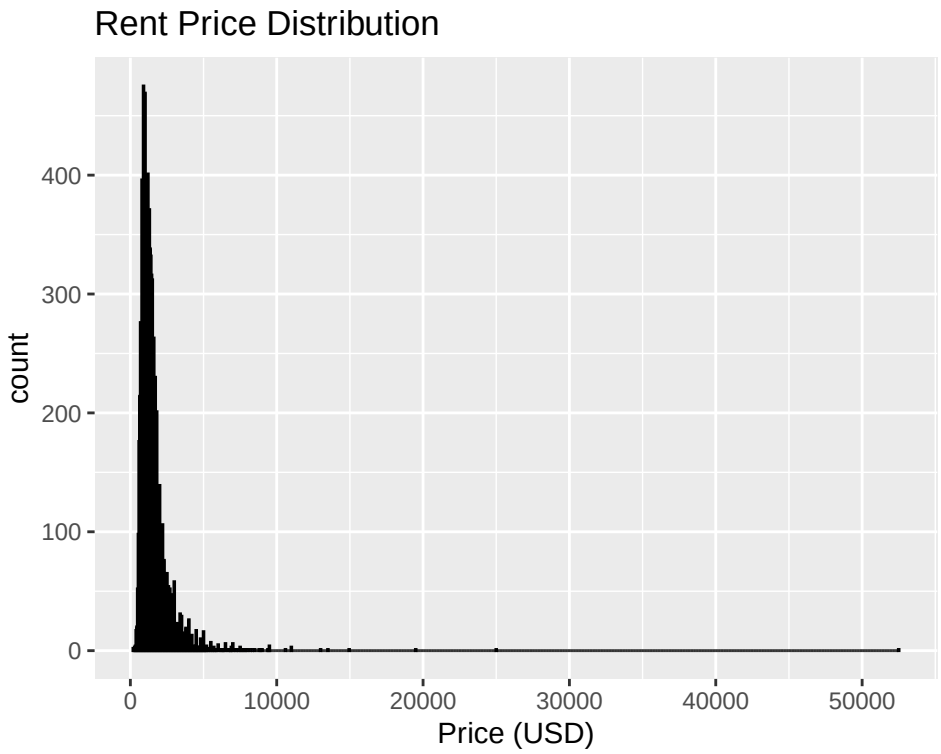
```r
colSums(is.na(data) | data == "null" | data == "NA" | data == "")
```

```
##            id      category         title          body     amenities
##             0             0             0             0          3549
##     bathrooms      bedrooms      currency           fee     has_photo
##            34             7             0             0             0
##  pets_allowed         price price_display    price_type   square_feet
##          1748             0             0             0             0
##       address      cityname         state      latitude     longitude
##          3327            77            77            10            10
##        source          time
##             0             0
```

2.4 Phân phối của một số biến

Phân phối giá thuê

```r
ggplot(data, aes(x = price)) +
  geom_histogram(binwidth = 50, fill = "blue", color = "black") +
  ggtitle("Rent Price Distribution") +
  xlab("Price (USD)")
```



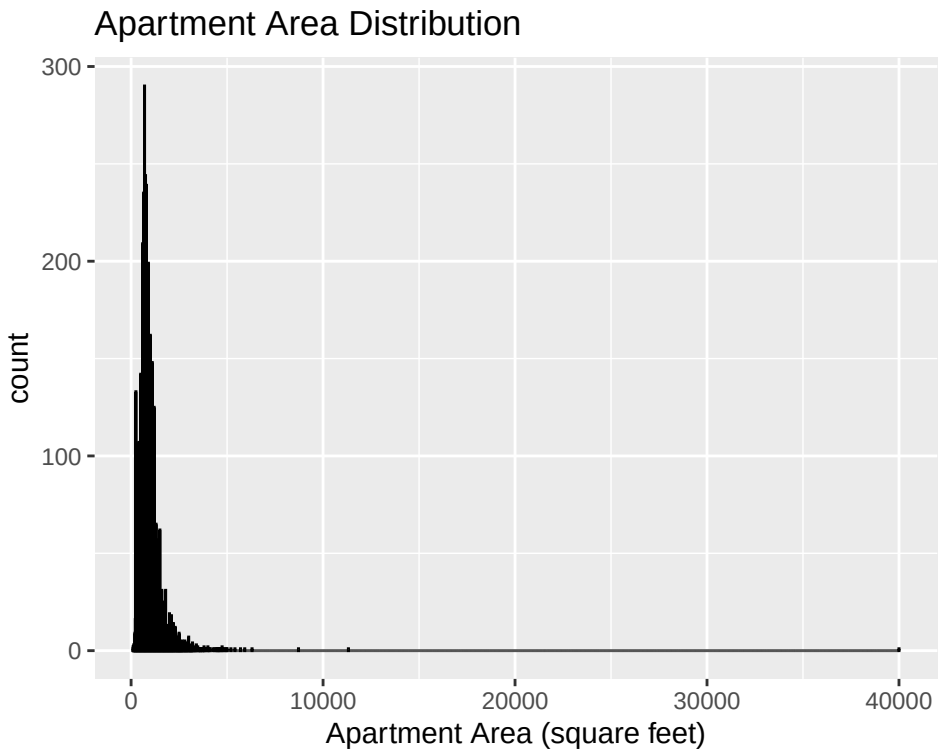
Dựa vào biểu đồ phân phối giá thuê, có thể nhận thấy một số điểm sau:

- Phân phối giá thuê có dạng phân phối không đối xứng (skewed distribution), với đường cong nghiêng về phía giá thuê thấp hơn.
- Giá thuê tập trung chủ yếu ở khoảng dưới 10,000 USD, với một số giá thuê cao hơn 30,000 USD nhưng số lượng rất ít.
- Có một "đỉnh" rõ rệt ở khoảng 0-100 USD, cho thấy có nhiều căn hộ có giá thuê rất thấp.
- Phân phối giá thuê có sự phân tán lớn, từ giá thuê rất thấp đến rất cao, cho thấy thị trường cho thuê căn hộ khá đa dạng.

Nhìn chung, biểu đồ phản ánh một thị trường cho thuê căn hộ có sự phân hóa và phân tán giá thuê khá rõ rệt. Điều này có thể liên quan đến các yếu tố như vị trí, loại hình, chất lượng căn hộ và nhu cầu của khách hàng.

Phân phối diện tích căn hộ

```
ggplot(data, aes(x = square_feet)) +
  geom_histogram(binwidth = 10, fill = "green", color = "black") +
  ggtitle("Apartment Area Distribution") +
  xlab("Apartment Area (square feet)")
```



Dựa vào biểu đồ phân phối diện tích căn hộ, có thể nhận thấy một số điểm sau:

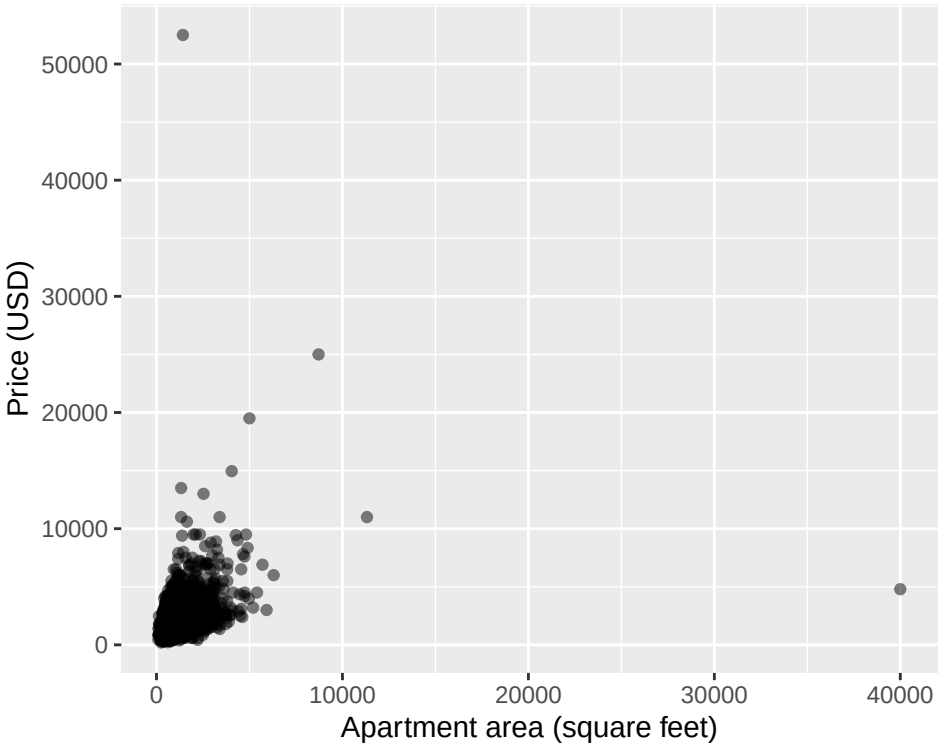
- Phân phối diện tích căn hộ có dạng phân phối không đối xứng (skewed distribution), với đường cong nghiêng về phía diện tích thấp hơn.
- Diện tích căn hộ tập trung chủ yếu ở khoảng dưới 2,000 square feet, với một số căn hộ có diện tích lên đến khoảng 35,000 square feet nhưng số lượng rất ít.
- Có một "đỉnh" rõ rệt ở khoảng 0-500 square feet, cho thấy có nhiều căn hộ có diện tích rất nhỏ.
- Phân phối diện tích căn hộ có sự phân tán lớn, từ diện tích rất nhỏ đến rất lớn, cho thấy thị trường cho thuê có sự đa dạng về kích thước căn hộ.

Nhìn chung, biểu đồ phản ánh một thị trường cho thuê có sự phân hóa và phân tán diện tích căn hộ khá rõ rệt. Điều này có thể liên quan đến các yếu tố như vị trí, mức độ xây dựng, nhu cầu của khách hàng và các chính sách quy hoạch của địa phương.

2.5 Mối quan hệ giữa các biến

Mối quan hệ giữa diện tích căn hộ và giá thuê

```
ggplot(data, aes(x = square_feet, y = price)) +
  geom_point(alpha = 0.5) +
  xlab("Apartment area (square feet)") +
  ylab("Price (USD)")
```



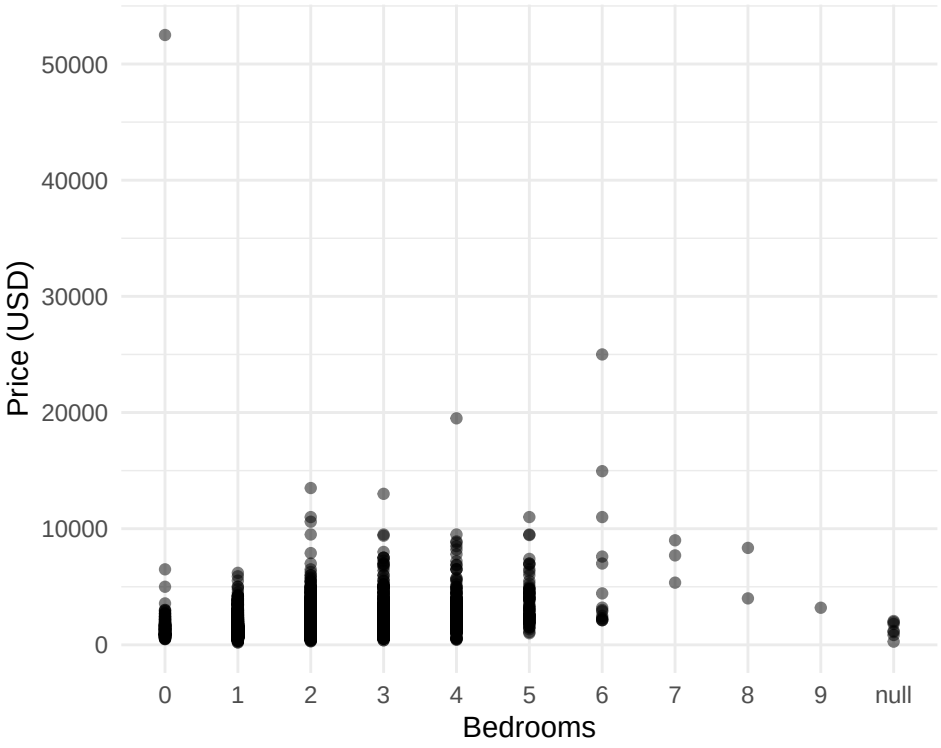
Nhìn vào mối quan hệ giữa diện tích căn hộ và giá thuê, có thể nhận thấy:

- Có mối quan hệ tương đối tích cực giữa diện tích căn hộ và giá thuê, nghĩa là khi diện tích căn hộ tăng, giá thuê cũng có xu hướng tăng theo.
- Tuy nhiên, mối quan hệ này không hoàn toàn tuyến tính, mà khá phân tán. Có những căn hộ có diện tích lớn nhưng giá thuê không cao, và ngược lại.
- Phần lớn các điểm dữ liệu tập trung ở khu vực diện tích nhỏ hơn 20,000 square feet và giá thuê dưới 20,000 USD, cho thấy đây là phân khúc chiếm đa số trong thị trường.
- Vẫn có một số căn hộ có diện tích lên đến 35,000 square feet và giá thuê lên đến 40,000 USD, cho thấy sự đa dạng về quy mô và phân khúc giá trong thị trường cho thuê này.

Nhìn chung, biểu đồ cho thấy diện tích căn hộ là một yếu tố ảnh hưởng đến giá thuê, nhưng không phải là yếu tố duy nhất. Các yếu tố khác như vị trí, chất lượng, tiện ích cũng đóng vai trò quan trọng trong xác định giá thuê căn hộ.

Mối quan hệ giữa số phòng ngủ và giá thuê

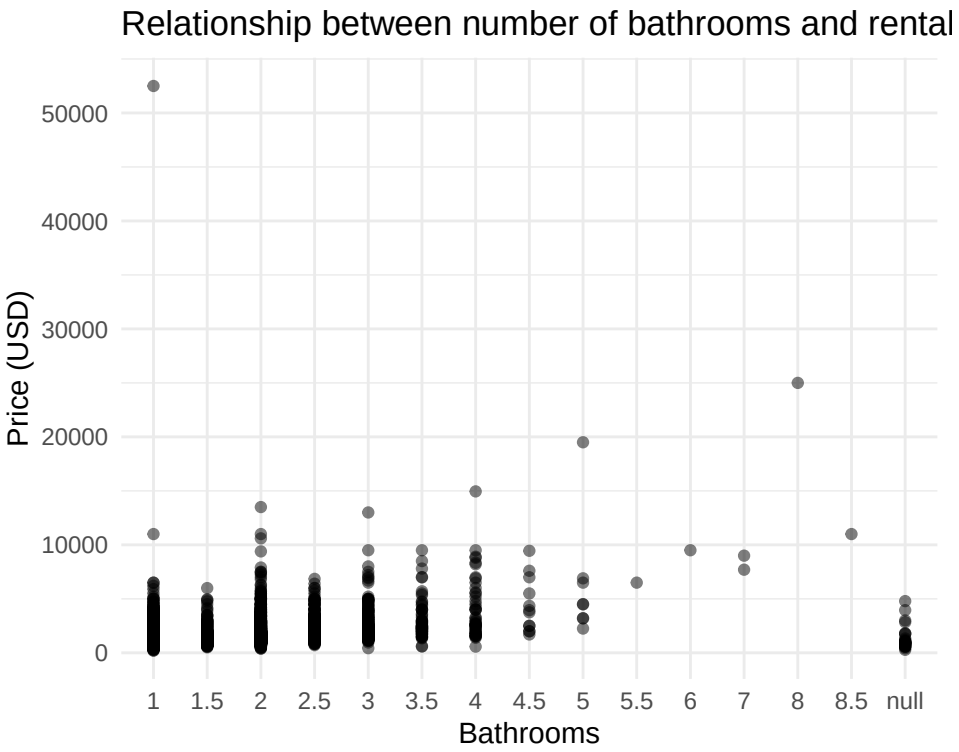
```
ggplot(data, aes(x = bedrooms, y = price)) +
  geom_point(alpha = 0.5) +
  xlab("Bedrooms") +
  ylab("Price (USD)") +
  theme_minimal()
```



- Có mối quan hệ tương đối tích cực giữa số phòng ngủ và giá thuê, nghĩa là khi số phòng ngủ tăng, giá thuê cũng có xu hướng tăng theo.
- Tuy nhiên, mối quan hệ này không hoàn toàn tuyến tính, mà khá phân tán. Có những căn hộ có số phòng ngủ nhiều nhưng giá thuê không cao, và ngược lại.
- Phần lớn các điểm dữ liệu tập trung ở khu vực số phòng ngủ từ 1 đến 5, với giá thuê chủ yếu dưới 20,000 USD, cho thấy đây là phân khúc chiếm đa số trong thị trường.
- Vẫn có một số ít căn hộ có số phòng ngủ lên đến 8 và 9, với giá thuê lên đến 30,000 USD, cho thấy sự đa dạng về quy mô và phân khúc giá trong thị trường cho thuê này.
- Một số điểm nằm ở "null" cho số phòng ngủ, có thể là những căn hộ studio hoặc không có thông tin về số phòng ngủ.

Mối quan hệ giữa số phòng tắm và giá thuê

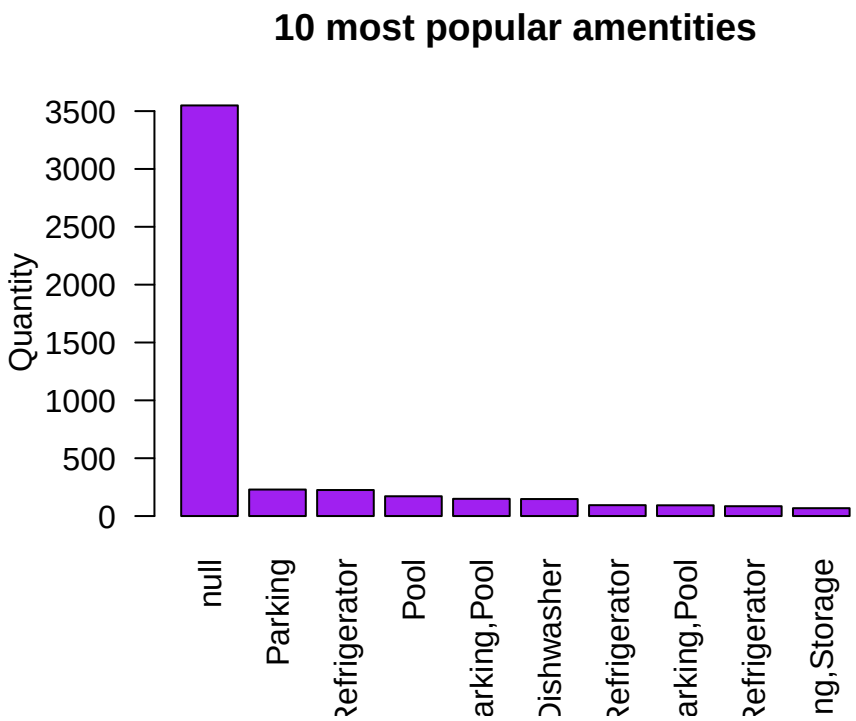
```
ggplot(data, aes(x = bathrooms, y = price)) +
  geom_point(alpha = 0.5) +
  ggtitle("Relationship between number of bathrooms and rental price") +
  xlab("Bathrooms") +
  ylab("Price (USD)") +
  theme_minimal()
```



- Tương tự như số phòng ngủ, có mối quan hệ tích cực giữa số phòng tắm và giá thuê. Khi số phòng tắm tăng, giá thuê có xu hướng tăng theo.
- Tuy nhiên, mối quan hệ này không hoàn toàn tuyến tính và khá phân tán. Có nhiều điểm dữ liệu với số phòng tắm tương đương nhưng giá thuê lại khác biệt đáng kể.
- Phần lớn các điểm dữ liệu tập trung ở khu vực số phòng tắm từ 1 đến 4, với giá thuê chủ yếu dưới 20,000 USD. Đây có vẻ là phân khúc chính trong thị trường.
- Vẫn có một số ít căn hộ có số phòng tắm cao hơn, lên đến 6, 7 và 8, với giá thuê lên đến 30,000 USD. Điều này cho thấy sự đa dạng về quy mô và phân khúc giá trong thị trường.
- Một số điểm ở "null" cho số phòng tắm, có thể là những căn hộ không có thông tin về số phòng tắm.

2.6 Hiển thị 10 tiện ích phổ biến nhất

```
amenities_counts <- table(data$amenities)

barplot(sort(amenities_counts, decreasing = TRUE)[1:10],
        las = 2,
        col = "purple",
        main = "10 most popular amentities",
        ylab = "Quantity")
```



Dựa vào biểu đồ, có thể thấy:

- Giá trị "null" lại chiếm số lượng lớn nhất. Điều này có thể chỉ ra rằng một số căn hộ cho thuê có thông tin về tiện ích chưa đầy đủ.
- Tiện ích phổ biến tiếp theo là "Parking", "Conditioning Pool", "Washer" và "Refrigerator". Đây là những tiện ích cơ bản và phổ biến thường được cung cấp trong các thông tin cho thuê căn hộ.
- Một số tiện ích khác như "Lerigator" và "arkingPol" có số lượng ít hơn, tức là chúng ít phổ biến hơn trong các căn hộ cho thuê được liệt kê.

3. Tiền xử lý dữ liệu

3.1 Loại bỏ các cột không cần thiết

Để tập trung vào việc phân tích giá thuê căn hộ theo diện tích, số phòng ngủ, số phòng tắm và địa điểm, chúng ta sẽ loại bỏ các cột không cần thiết. Do các thuộc tính chứa giá trị text không phải mục tiêu của chúng ta, **price_display** mang giá trị lặp lại so với **price**, một số thuộc tính không có sự phân biệt rõ ràng giữa các quan sát (như state, price_type…). Các tiện ích như điều hòa, sân bóng rổ, cáp, phòng gym, truy cập internet, bể bơi, tủ lạnh, v.v. Mặc dù các tiện ích này có thể ảnh hưởng đến giá thuê, nhưng để đơn giản hóa phân tích ban đầu, chúng ta sẽ loại bỏ chúng. Sau này, nếu cần, ta có thể xử lý văn bản để phân tích các tiện ích này, v.v…

```
columns_to_remove <- c("id", "category", "title", "body", "amenities", "currency", "fee",
                       "has_photo", "price_display", "price_type", "pets_allowed",
                       "address", "state", "latitude", "longitude", "source", "time")

data <- data[, !(names(data) %in% columns_to_remove)]
```

3.2 Xử lý dữ liệu thiếu

Thay thế các giá trị đặc biệt bằng NA trong cột bathrooms và bedrooms:

```
data$bathrooms[data$bathrooms %in% c("null", "NA", "")] <- NA
data$bedrooms[data$bedrooms %in% c("null", "NA", "")] <- NA

# Chuyển đổi sang dạng numeric
data$bathrooms <- as.numeric(data$bathrooms)
data$bedrooms <- as.numeric(data$bedrooms)
```

Vì số lượng các giá trị bị thiếu trong các cột này là rất ít so với số lượng mẫu mà ta có được. Vậy nên thay vì thay thế các giá trị bị thiếu bằng một giá trị khác thì ta sẽ loại bỏ chúng.

```
data <- data[complete.cases(data$bathrooms, data$bedrooms), ]
data <- data[!grepl("^\\s*$|^null$|^NA$", data$cityname, ignore.case = TRUE), ]
```

3.3 Xử lý cột cityname

```
data$cityname <- trimws(data$cityname)

location_stats <- data |>
  group_by(cityname) |>
  summarize(count = n()) |>
  arrange(desc(count))

location_stats
```

```
## # A tibble: 1,572 x 2
##    cityname      count
##    <chr>         <int>
##  1 Austin          522
##  2 Dallas          215
##  3 Houston         186
##  4 San Antonio     182
##  5 Los Angeles     165
##  6 Chicago         147
##  7 Madison         121
##  8 Portland        113
##  9 Denver          105
## 10 San Francisco   104
## # i 1,562 more rows
```

Thực hiện gộp các thành phố có ít hơn 10 căn hộ vào nhóm “other” để:

- Giảm số lượng biến phân loại, làm cho mô hình đơn giản và dễ quản lý hơn
- Giảm thiểu biến động từ các thành phố có ít dữ liệu, giúp mô hình ổn định hơn
- Tránh overfitting khi có quá nhiều biến với ít dữ liệu.

```r
# Đếm số lượng thành phố có số lượng căn hộ cho thuê ít hơn hoặc bằng 10
num_locations <- sum(location_stats$count <= 10)

# Lọc ra các thành phố có số lượng căn hộ cho thuê ít hơn hoặc bằng 10
location_stats_less_than_10 <- location_stats[location_stats$count <= 10, ]
location_stats_less_than_10
```

```
## # A tibble: 1,395 x 2
##    cityname   count
##    <chr>      <int>
##  1 Auburn        10
##  2 Bellingham    10
##  3 Cambridge     10
##  4 Chesapeake    10
##  5 Flanders      10
##  6 Hampton       10
##  7 Irmo          10
##  8 Kissimmee     10
##  9 Kyle          10
## 10 Manhattan     10
## # i 1,385 more rows
```

```r
# Tạo một vector logic để xác định các thành phố nằm trong điều kiện trên
is_less_than_10 <- data$cityname %in% location_stats_less_than_10$cityname

# Thay thế các thành phố thỏa điều kiện bằng nhãn "other"
data$cityname[is_less_than_10] <- "other"
num_unique_cities <- length(unique(data$cityname))
num_unique_cities
```

```
## [1] 178
```

3.4 Lọc dữ liệu

Lọc các hàng thỏa điều kiện diện tích chia cho số phòng ngủ nhỏ hơn 250, vì các căn hộ có diện tích như vậy là quá nhỏ đối với số phòng ngủ được cung cấp, có thể do các lý do như thông tin sai lệch về diện tích hoặc số phòng ngủ.

```r
filtered_data <- data[data$square_feet / data$bedrooms < 250, ]
df1 <- data[!(data$square_feet / data$bedrooms < 250), ]
df2 <- df1
```

Tiếp theo, ta sẽ tính toán giá trị trung bình của mỗi mét vuông (**price_per_square_feet**) từ hai cột dữ liệu là **price** (giá thuê) và **square_feet** (diện tích), bởi vì khi tham khảo giá thuê căn hộ, nhiều người sẽ quan tâm đến giá tiền trên 1 mét vuông là bao nhiêu nên đây có thể là một thuộc tính hữu ích.

```r
df2$price_per_sqft <- df2$price * 100000 / df2$square_feet
head(df2)
```

```
##    bathrooms bedrooms price square_feet      cityname price_per_sqft
```

```
## 3          1        0  1390         107     Arlington      1299065.4
## 4          1        0   925         116       Seattle       797413.8
## 6          1        0  2475         130         other      1903846.2
## 9          1        0  1495         138 San Francisco      1083333.3
## 15         1        0  1695         190 San Francisco       892105.3
## 39         1        0  1195         223       Seattle       535874.4
```

3.5 Loại bỏ các điểm ngoại lệ

Tạo một hàm để loại bỏ ngoại lệ dựa trên mean và standard deviation của **price_per_sqft** theo từng thành phố. Phạm vi mean ± 3*standard deviation là một ngưỡng phổ biến để xác định các điểm dữ liệu bất thường nên ta sẽ áp dụng chúng.

```
remove_pps_outliers <- function(df) {
  df_out <- data.frame() # Tạo DataFrame rỗng để lưu kết quả

  # Lặp qua từng nhóm thành phố
  city_stats <- df |>
    group_by(cityname) |>
    summarize(mean_pps = mean(price_per_sqft),
              std_pps = sd(price_per_sqft))

  # Lặp qua từng nhóm thành phố
  for (key in unique(df$cityname)) {
    subdf <- df[df$cityname == key, ] # Lấy các dòng dữ liệu của từng thành phố

    m <- city_stats$mean_pps[city_stats$cityname == key] # Lấy mean
    st <- city_stats$std_pps[city_stats$cityname == key] # Lấy standard deviation

    # Lọc các dòng dữ liệu ngoài phạm vi mean ± std
    reduced_df <- subdf[(subdf$price_per_sqft > (m - 3*st)) &
                        (subdf$price_per_sqft <= (m + 3*st)), ]

    # Ghép reduced_df vào df_out
    df_out <- rbind(df_out, reduced_df)
  }

  return(df_out)
}

df3 <- remove_pps_outliers(df2)
```

Ta chỉ lấy các dòng trong df3 có số phòng ngủ lớn hơn 0:

```
df3 <- df3[df3$bedrooms > 0, ]
dim(df3)
```

```
## [1] 9225    6
```

Tiếp tục loại bỏ các điểm ngoại lệ cho **square_feet** và **price**:

```r
sqft_mean <- mean(df3$square_feet)
sqft_std <- sd(df3$square_feet)
price_mean <- mean(df3$price)
price_std <- sd(df3$price)

df3 <- df3[abs(df3$square_feet - sqft_mean) < 3 * sqft_std &
             abs(df3$price - price_mean) < 3 * price_std, ]

#Loại bỏ các dòng có các giá trị không hợp lệ
df3 <- df3[df3$bathrooms >= 0 &
             df3$bedrooms >= 0 &
             df3$square_feet > 0 &
             df3$price > 0, ]

# Log transform cho cột price để normalize phân phối của giá.
df3$price <- log(df3$price)

#Loại bỏ outliers cho cột price
df3 <- df3[df3$price < mean(df3$price) + 3 * sd(df3$price), ]
head(df3)
```

```
##      bathrooms bedrooms    price square_feet  cityname price_per_sqft
## 104          1        1 6.774224         250 Arlington       350000.0
## 820          1        1 6.505784         480 Arlington       139375.0
## 1118         1        1 6.395262         518 Arlington       115637.1
## 1245         1        2 7.649693         533 Arlington       393996.2
## 1256         1        2 7.673223         535 Arlington       401869.2
## 1569         1        2 7.577122         568 Arlington       343838.0
```

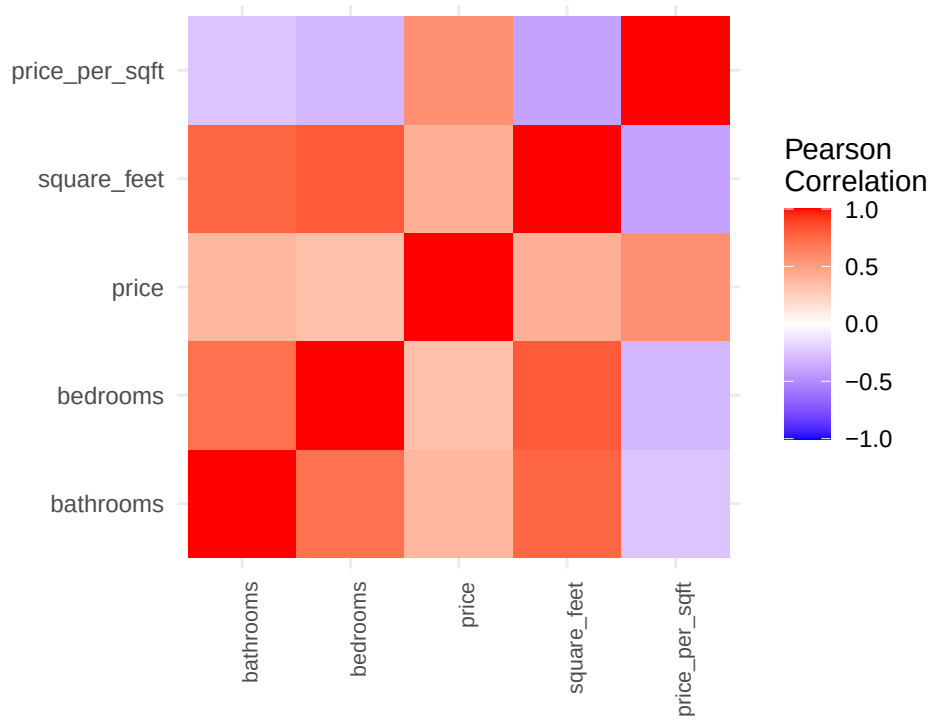
3.6 Tạo biến giả (dummy variables) cho cột cityname

```r
dummy_vars <- model.matrix(~ cityname + 0, data = df3)

# Chuyển đổi kết quả thành data frame
dummy_df <- as.data.frame(dummy_vars)

# Loại bỏ cột intercept để tránh dummy variable trap
dummy_df <- dummy_df[, -1]

# Nối các biến giả vào dữ liệu gốc (df3)
df4 <- cbind(df3, dummy_df)

## Loại bỏ cột cityname sau khi đã tạo các biến giả
final_df <- subset(df4, select = -cityname)
final_df <- clean_names(final_df)

dim(final_df)
```

```
## [1] 8964  182
```

3.7 Tính toán ma trận tương quan giữa các biến số

```
cor_matrix <- cor(df3[, sapply(df3, is.numeric)], method = "pearson")
melted_cor_matrix <- reshape2::melt(cor_matrix)

ggplot(data = melted_cor_matrix, aes(x = Var1, y = Var2, fill = value)) +
  geom_tile() +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                       midpoint = 0, limit = c(-1, 1), space = "Lab",
                       name = "Pearson\nCorrelation") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, vjust = 1, size = 8, hjust = 1),
        axis.title.x = element_blank(),
        axis.title.y = element_blank()) +
  coord_fixed()
```



Từ biểu đồ heatmap, chúng ta có thể nhận xét mối tương quan giữa các biến so với biến price (giá thuê) như sau:

1. **bathrooms** có mối tương quan dương mạnh với giá thuê. Điều này cho thấy khi số lượng phòng tắm tăng, giá thuê căn hộ cũng tăng. Đây là điều hợp lý vì nhiều phòng tắm thường đi kèm với các căn hộ lớn hơn và tiện nghi hơn, dẫn đến giá thuê cao hơn.
2. **bedrooms** có mối tương quan dương trung bình với giá thuê. Khi số lượng phòng ngủ tăng, giá thuê cũng có xu hướng tăng. Số phòng ngủ là một yếu tố quan trọng trong việc định giá căn hộ, vì nhiều phòng ngủ hơn thường đồng nghĩa với diện tích lớn hơn và nhiều không gian sinh hoạt hơn.
3. **square_feet** có tương quan dương trung bình với giá thuê. Diện tích căn hộ càng lớn thì giá thuê càng cao. Diện tích là yếu tố chính trong việc định giá bất động sản, và một căn hộ lớn hơn thường có giá thuê cao hơn.

4. **price_per_sqft** có mối tương quan âm yếu với giá thuê. Điều này có thể gợi ý rằng các căn hộ có giá thuê cao hơn có giá trên mỗi mét vuông thấp hơn một chút. Một phần lý do có thể là nhu cầu đối với căn hộ nhỏ thường cao hơn do chúng phù hợp với nhiều đối tượng thuê nhà, bao gồm sinh viên, người độc thân và các gia đình nhỏ. Điều này có thể đẩy giá trên mỗi mét vuông của các căn hộ nhỏ lên cao hơn so với các căn hộ lớn hơn, vốn có nhu cầu thấp hơn.

4. Xây dựng mô hình hồi quy tuyến tính

Trước khi tiến hành xây dựng mô hình hồi quy đơn giản, ta sẽ chia tập dữ liệu thành hai tập là huấn luyện (**train_data**) và kiểm thử (**test_data**) với tỉ lệ là 7/3:

```
set.seed(42)
trainIndex <- createDataPartition(final_df$price, p = 0.7, list = FALSE)
train_data <- final_df[trainIndex, ]
test_data <- final_df[-trainIndex, ]
```

4.1 Xây dựng mô hình hồi quy đơn giản

```
lm_model <- lm(price ~ ., data = train_data)

# Thống kê tổng hợp
summary(lm_model)
```

```
##
## Call:
## lm(formula = price ~ ., data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.10347 -0.05863  0.02478  0.09842  0.95177
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 5.598e+00  7.411e-02  75.539  < 2e-16 ***
## bathrooms                   3.178e-02  6.356e-03   5.000 5.88e-07 ***
## bedrooms                   -2.863e-02  4.455e-03  -6.427 1.40e-10 ***
## square_feet                 8.587e-04  1.108e-05  77.498  < 2e-16 ***
## price_per_sqft              4.745e-06  4.045e-08 117.296  < 2e-16 ***
## cityname_alexandria         3.315e-02  9.021e-02   0.367  0.71330
## cityname_alpharetta         8.217e-02  8.436e-02   0.974  0.33007
## cityname_ames              -5.395e-02  7.706e-02  -0.700  0.48391
## cityname_anaheim            4.630e-03  8.585e-02   0.054  0.95699
## cityname_anchorage         -1.083e-02  8.229e-02  -0.132  0.89526
## cityname_ann_arbor          4.298e-02  8.665e-02   0.496  0.61988
## cityname_apex              -4.427e-02  8.663e-02  -0.511  0.60933
## cityname_arlington         -4.927e-03  7.662e-02  -0.064  0.94873
## cityname_atlanta            1.337e-02  7.918e-02   0.169  0.86588
## cityname_aurora            -1.419e-01  9.643e-02  -1.472  0.14118
## cityname_austin             1.163e-02  7.419e-02   0.157  0.87539
## cityname_azle               6.714e-02  9.180e-02   0.731  0.46461
```

```
## cityname_baltimore        -1.170e-02  8.376e-02  -0.140  0.88889
## cityname_baton_rouge      -1.266e-01  8.231e-02  -1.538  0.12415
## cityname_bedford           7.609e-02  8.191e-02   0.929  0.35295
## cityname_bellevue          2.831e-02  8.133e-02   0.348  0.72776
## cityname_bismarck         -3.174e-02  9.400e-02  -0.338  0.73562
## cityname_bloomington      -4.103e-02  7.898e-02  -0.520  0.60342
## cityname_boise             1.163e-02  8.880e-02   0.131  0.89579
## cityname_boston           -1.477e-01  8.720e-02  -1.693  0.09042 .
## cityname_bothell           1.286e-01  8.667e-02   1.484  0.13792
## cityname_burlington       -4.642e-02  9.384e-02  -0.495  0.62080
## cityname_cary             -3.724e-02  8.503e-02  -0.438  0.66144
## cityname_cedar_falls      -6.445e-02  9.179e-02  -0.702  0.48265
## cityname_cedar_park        1.868e-02  9.381e-02   0.199  0.84218
## cityname_champaign        -3.488e-02  7.933e-02  -0.440  0.66022
## cityname_chapel_hill       4.996e-02  9.179e-02   0.544  0.58629
## cityname_charlotte        -1.119e-01  7.975e-02  -1.403  0.16066
## cityname_cherry_hill       1.315e-01  9.967e-02   1.320  0.18697
## cityname_chicago           2.068e-02  7.570e-02   0.273  0.78470
## cityname_cincinnati        1.093e-02  7.676e-02   0.142  0.88675
## cityname_clayton          -6.454e-02  8.510e-02  -0.758  0.44823
## cityname_cleveland        -3.302e-02  8.661e-02  -0.381  0.70303
## cityname_colorado_springs  4.257e-02  7.699e-02   0.553  0.58035
## cityname_columbia         -5.115e-02  7.884e-02  -0.649  0.51655
## cityname_columbus          1.425e-02  8.015e-02   0.178  0.85886
## cityname_concord          -1.593e-02  8.579e-02  -0.186  0.85270
## cityname_coraopolis        7.313e-02  8.875e-02   0.824  0.40998
## cityname_cumming           1.907e-02  9.643e-02   0.198  0.84327
## cityname_dallas            2.847e-02  7.487e-02   0.380  0.70377
## cityname_davenport        -3.000e-02  9.382e-02  -0.320  0.74915
## cityname_dayton           -2.197e-01  8.374e-02  -2.624  0.00872 **
## cityname_decatur          -2.452e-02  8.574e-02  -0.286  0.77491
## cityname_denver            7.674e-02  7.615e-02   1.008  0.31358
## cityname_des_moines       -8.464e-02  9.381e-02  -0.902  0.36700
## cityname_detroit          -8.269e-02  9.406e-02  -0.879  0.37941
## cityname_durham            9.773e-04  8.015e-02   0.012  0.99027
## cityname_edmond           -9.152e-02  8.039e-02  -1.138  0.25500
## cityname_ellicott_city     1.386e-01  8.507e-02   1.630  0.10321
## cityname_euless            4.245e-03  7.883e-02   0.054  0.95706
## cityname_everett           4.974e-02  9.970e-02   0.499  0.61792
## cityname_fargo            -7.113e-02  9.383e-02  -0.758  0.44844
## cityname_fayetteville     -1.435e-01  9.188e-02  -1.562  0.11834
## cityname_federal_way       1.161e-01  9.385e-02   1.237  0.21619
## cityname_forest_lake       1.108e-01  9.183e-02   1.207  0.22745
## cityname_fort_collins      1.294e-02  8.664e-02   0.149  0.88124
## cityname_fort_worth       -1.336e-02  9.017e-02  -0.148  0.88225
## cityname_gaithersburg      1.480e-01  8.883e-02   1.666  0.09582 .
## cityname_georgetown       -3.664e-02  8.761e-02  -0.418  0.67581
## cityname_glen_burnie       5.747e-03  8.190e-02   0.070  0.94406
## cityname_glendale         -2.596e-02  8.666e-02  -0.300  0.76451
## cityname_grand_forks      -2.048e-01  8.435e-02  -2.428  0.01522 *
## cityname_grapevine         1.116e-01  9.182e-02   1.215  0.22437
## cityname_greensboro       -5.771e-02  8.124e-02  -0.710  0.47755
## cityname_greenville       -2.948e-02  9.390e-02  -0.314  0.75353
## cityname_hackensack        1.041e-01  9.191e-02   1.133  0.25739
```

```
## cityname_hanover           1.452e-01  8.508e-02   1.707  0.08792 .
## cityname_hartford          4.477e-02  9.635e-02   0.465  0.64218
## cityname_houston          -4.081e-02  7.518e-02  -0.543  0.58726
## cityname_humble           -1.696e-01  9.390e-02  -1.806  0.07102 .
## cityname_hyattsville       1.332e-01  8.879e-02   1.501  0.13352
## cityname_indianapolis     -1.414e-01  8.230e-02  -1.718  0.08588 .
## cityname_issaquah          1.696e-01  8.887e-02   1.908  0.05644 .
## cityname_jacksonville     -1.340e-01  8.661e-02  -1.547  0.12186
## cityname_jersey_city      -1.523e-02  8.140e-02  -0.187  0.85163
## cityname_kansas_city      -3.498e-02  7.637e-02  -0.458  0.64696
## cityname_kent              9.625e-02  7.935e-02   1.213  0.22522
## cityname_kirkland          9.946e-02  8.332e-02   1.194  0.23260
## cityname_lafayette        -1.087e-01  8.431e-02  -1.289  0.19736
## cityname_lakewood         -4.178e-02  8.879e-02  -0.471  0.63797
## cityname_las_vegas         1.425e-03  7.708e-02   0.018  0.98525
## cityname_laurel            1.329e-01  8.500e-02   1.564  0.11795
## cityname_lawrence         -6.987e-03  8.660e-02  -0.081  0.93570
## cityname_lawrenceville     1.485e-02  8.434e-02   0.176  0.86025
## cityname_lexington        -1.174e-01  8.278e-02  -1.418  0.15616
## cityname_lincoln          -8.589e-02  9.016e-02  -0.953  0.34085
## cityname_lithonia         -5.563e-02  8.765e-02  -0.635  0.52571
## cityname_long_beach        6.840e-02  8.081e-02   0.846  0.39738
## cityname_longview         -1.077e-01  9.384e-02  -1.148  0.25110
## cityname_los_angeles      -1.381e-02  7.614e-02  -0.181  0.85611
## cityname_lubbock          -7.980e-02  8.664e-02  -0.921  0.35710
## cityname_lynnwood          1.181e-01  8.157e-02   1.448  0.14768
## cityname_madison           7.785e-02  7.591e-02   1.026  0.30513
## cityname_manchester        9.972e-02  8.063e-02   1.237  0.21620
## cityname_mandan           -8.964e-02  9.641e-02  -0.930  0.35251
## cityname_mansfield         6.789e-02  8.662e-02   0.784  0.43324
## cityname_maple_valley      1.773e-01  8.588e-02   2.065  0.03901 *
## cityname_marietta          3.833e-03  7.778e-02   0.049  0.96070
## cityname_miami            -5.624e-02  9.646e-02  -0.583  0.55988
## cityname_midland          -2.479e-02  9.181e-02  -0.270  0.78717
## cityname_milford           9.822e-02  9.383e-02   1.047  0.29525
## cityname_milwaukee         3.051e-02  7.776e-02   0.392  0.69480
## cityname_minneapolis       9.021e-03  7.835e-02   0.115  0.90834
## cityname_minot            -1.660e-01  7.950e-02  -2.089  0.03678 *
## cityname_monroe           -7.893e-02  9.024e-02  -0.875  0.38180
## cityname_monroeville      -3.730e-02  8.759e-02  -0.426  0.67022
## cityname_morrisville       2.800e-02  8.880e-02   0.315  0.75251
## cityname_nashville        -3.996e-02  8.284e-02  -0.482  0.62952
## cityname_new_braunfels    -2.033e-04  8.574e-02  -0.002  0.99811
## cityname_new_london        1.471e-02  8.432e-02   0.174  0.86153
## cityname_newport_news     -7.975e-02  1.041e-01  -0.766  0.44355
## cityname_norfolk           4.680e-02  8.761e-02   0.534  0.59326
## cityname_norman           -5.624e-02  7.905e-02  -0.711  0.47685
## cityname_oakland          -1.071e-01  8.625e-02  -1.242  0.21444
## cityname_odenton           1.456e-01  9.645e-02   1.510  0.13111
## cityname_oklahoma_city    -3.451e-02  7.854e-02  -0.439  0.66044
## cityname_omaha            -6.178e-02  7.749e-02  -0.797  0.42535
## cityname_orange            5.357e-02  8.764e-02   0.611  0.54110
## cityname_orlando          -5.365e-02  8.232e-02  -0.652  0.51460
## citynameother             -2.794e-02  7.370e-02  -0.379  0.70467
```

```
## cityname_overland_park    -7.178e-02  9.965e-02  -0.720  0.47135
## cityname_owings_mills      1.079e-01  9.966e-02   1.083  0.27898
## cityname_pasadena          2.472e-02  8.040e-02   0.307  0.75855
## cityname_pflugerville      1.314e-02  8.878e-02   0.148  0.88231
## cityname_philadelphia      9.889e-02  8.067e-02   1.226  0.22026
## cityname_phoenix          -3.875e-02  7.842e-02  -0.494  0.62127
## cityname_pittsburgh        1.565e-02  7.990e-02   0.196  0.84472
## cityname_plainsboro        1.228e-01  8.577e-02   1.431  0.15238
## cityname_plymouth          7.102e-02  8.878e-02   0.800  0.42377
## cityname_portland         -3.062e-02  7.660e-02  -0.400  0.68936
## cityname_princeton         1.517e-01  8.677e-02   1.749  0.08037 .
## cityname_raleigh          -7.160e-04  7.712e-02  -0.009  0.99259
## cityname_randallstown      5.625e-02  9.965e-02   0.564  0.57246
## cityname_redmond           1.121e-01  9.392e-02   1.194  0.23267
## cityname_reno              2.402e-02  8.662e-02   0.277  0.78158
## cityname_renton            1.081e-01  7.956e-02   1.358  0.17443
## cityname_richmond          1.593e-02  8.880e-02   0.179  0.85759
## cityname_rochester        -3.360e-02  8.875e-02  -0.379  0.70501
## cityname_rockville         1.225e-01  8.883e-02   1.379  0.16787
## cityname_roswell           2.544e-02  8.661e-02   0.294  0.76901
## cityname_round_rock        3.125e-02  8.272e-02   0.378  0.70558
## cityname_saint_george     -1.411e-01  1.203e-01  -1.173  0.24076
## cityname_saint_louis      -6.647e-02  7.898e-02  -0.842  0.40003
## cityname_saint_louis_park  6.508e-02  8.880e-02   0.733  0.46366
## cityname_saint_paul        5.485e-02  8.436e-02   0.650  0.51561
## cityname_saint_petersburg -7.301e-02  9.181e-02  -0.795  0.42649
## cityname_salt_lake_city    4.245e-02  8.154e-02   0.521  0.60266
## cityname_san_antonio      -3.093e-02  7.522e-02  -0.411  0.68100
## cityname_san_diego         8.679e-02  7.887e-02   1.100  0.27120
## cityname_san_francisco    -8.057e-01  8.001e-02 -10.071  < 2e-16 ***
## cityname_san_jose         -2.953e-02  9.205e-02  -0.321  0.74834
## cityname_san_marcos        2.649e-02  9.636e-02   0.275  0.78342
## cityname_santa_monica     -1.721e-01  9.442e-02  -1.823  0.06840 .
## cityname_sarasota          8.753e-02  9.184e-02   0.953  0.34054
## cityname_seattle           1.276e-02  7.768e-02   0.164  0.86954
## cityname_silver_spring     1.444e-01  8.275e-02   1.744  0.08113 .
## cityname_sioux_falls      -1.124e-01  7.868e-02  -1.429  0.15308
## cityname_smyrna            1.918e-02  9.184e-02   0.209  0.83456
## cityname_spring           -1.207e-01  9.022e-02  -1.338  0.18090
## cityname_springfield      -1.262e-01  8.152e-02  -1.547  0.12181
## cityname_suwanee           7.367e-02  9.183e-02   0.802  0.42247
## cityname_tallahassee      -1.522e-02  8.876e-02  -0.171  0.86386
## cityname_tampa             6.343e-03  8.157e-02   0.078  0.93802
## cityname_toledo           -1.612e-01  8.877e-02  -1.816  0.06939 .
## cityname_tucson           -9.084e-02  8.660e-02  -1.049  0.29423
## cityname_tulsa            -2.560e-01  8.372e-02  -3.057  0.00224 **
## cityname_tuscaloosa       -6.935e-02  8.587e-02  -0.808  0.41935
## cityname_tyler            -9.035e-02  9.637e-02  -0.938  0.34850
## cityname_urbana           -2.752e-02  8.434e-02  -0.326  0.74424
## cityname_vancouver         3.273e-02  8.229e-02   0.398  0.69082
## cityname_virginia_beach    7.015e-02  9.013e-02   0.778  0.43643
## cityname_washington       -4.912e-02  7.901e-02  -0.622  0.53416
## cityname_west_des_moines  -8.990e-02  9.965e-02  -0.902  0.36697
## cityname_west_hollywood    9.172e-02  1.000e-01   0.917  0.35915
```

```
## cityname_west_lafayette    3.773e-03  7.810e-02   0.048  0.96147
## cityname_west_new_york     9.639e-02  9.412e-02   1.024  0.30584
## cityname_winston_salem    -3.852e-02  9.014e-02  -0.427  0.66911
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1645 on 6094 degrees of freedom
## Multiple R-squared:  0.8514, Adjusted R-squared:  0.847
## F-statistic:   193 on 181 and 6094 DF,  p-value: < 2.2e-16
```

Bảng thống kê tổng hợp cho thấy với Multiple R-squared là 0.8514 và Adjusted R-squared là 0.847, mô hình hồi quy tuyến tính giải thích được khoảng 85% biến thiên của giá thuê căn hộ. Đây là mức độ giải thích rất tốt, cho thấy mô hình phù hợp với dữ liệu. Residual standard error (RSE) là 0.1645, cho biết sai số trung bình của dự đoán so với giá trị thực tế. Mặc dù không phải là hoàn hảo, nhưng độ lệch chuẩn này vẫn cho thấy mô hình có khả năng dự đoán tương đối chính xác. Các biến như **bathrooms, bedrooms, square_feet** và **price_per_sqft** có ảnh hưởng mạnh và có ý nghĩa thống kê cao đến giá căn hộ vì giá trị Pr(>|t|) của các features này rất nhỏ (đều dưới 0.001). Các thành phố cũng có tác động tích cực hoặc tiêu cực đáng kể đối với giá căn hộ do bởi mỗi thành phố có các tiện ích hạ tầng khác nhau (trung tâm thương mại, bệnh viện trung ương,...)

```r
# Dự đoán trên tập test
predictions <- predict(lm_model, newdata = test_data)

# Đánh giá model
RMSE <- sqrt(mean((test_data$price - predictions)^2))
cat("Root Mean Squared Error (RMSE):", RMSE, "\n")
```

```
## Root Mean Squared Error (RMSE): 0.1663875
```

RMSE bằng 0.1663875 là một con số tương đối nhỏ, chỉ hơi cao hơn một chút so với RSE trên tập huấn luyện (0.1645). Điều này cho thấy mô hình có khả năng dự đoán chính xác và không bị overfitting nhiều. Tuy nhiên, ta sẽ thực hiện cross validation để đảm bảo tính chính xác và tin cậy và đánh giá mô hình một cách toàn diện hơn.

4.2 Hồi quy từng bước với cross validation

Hàm predict.regsubsets() để tính giá trị tiên đoán Y dựa trên mô hình M_j (kết quả từ regsubsets()) và dữ liệu trong fold thứ r.

```r
predict.regsubsets <- function(object, newdata, id_model){
  form <- as.formula(object$call[[2]])
  x_mat <- model.matrix(form, newdata)
  coef_est <- coef(object, id = id_model)
  x_vars <- names(coef_est)
  res <- x_mat[, x_vars] %*% coef_est
  return(as.numeric(res))
}
```

Chọn số fold là 10:

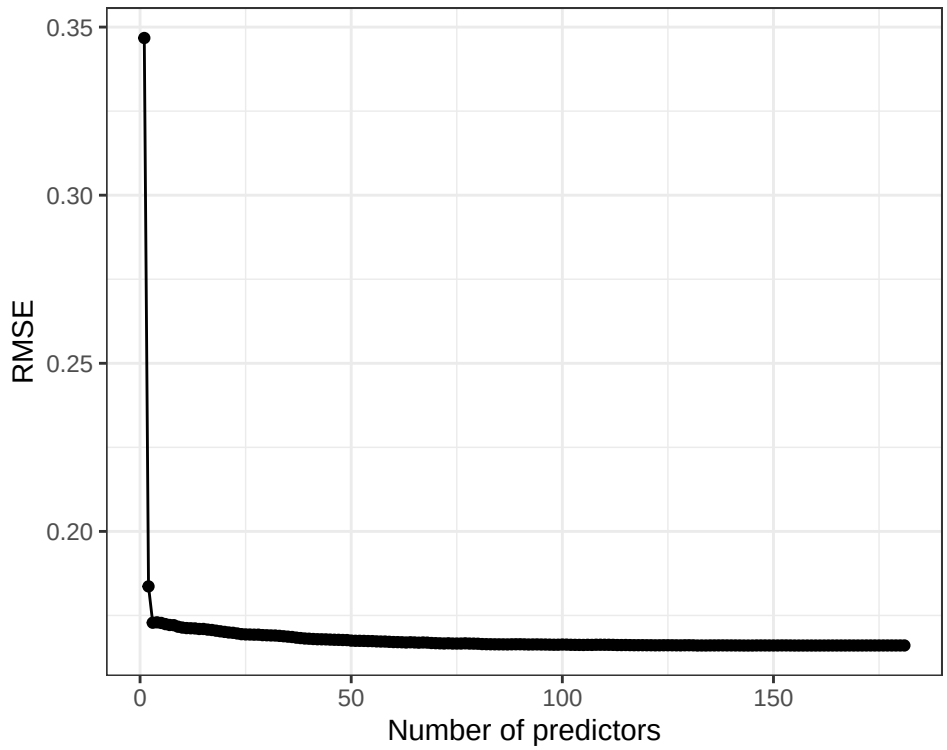
```r
n_df <- nrow(final_df)
k <- 10
set.seed(21)
folds <- sample(rep(1:k, length = n_df))
```

Tính 10-fold cross-validated error cho 181 mô hình con sử dụng phương pháp "forward":

```r
cv_error_df_rj <- matrix(0, nrow = k, ncol = 181)
for(r in 1:k){
  df_train_r <- final_df[folds != r, ]
  df_test_r <- final_df[folds == r, ]
  out_subset_df_folds <- regsubsets(x = price ~ ., data = df_train_r,
                                    method = "forward", nvmax = 181, really.big = T)
  for(j in 1:181){
    pred_rj <- predict(out_subset_df_folds,
                       newdata = df_test_r, id_model = j)
    cv_error_df_rj[r, j] <- sqrt(mean((df_test_r$price - pred_rj)^2))
  }
}
cv_error_df <- colMeans(cv_error_df_rj)
```

Biểu diễn kết quả 10-fold cross-validated error cho 181 mô hình con:

```r
ggplot(data = data.frame(x = c(1:181), y = cv_error_df),
       mapping = aes(x = x, y = y)) +
  geom_point() +
  geom_line() +
  labs(x = "Number of predictors", y = "RMSE") +
  theme_bw()
```



Trong số 181 mô hình con được xây dựng bằng phương pháp "forward", khi số lượng biến dự báo tăng lên, thì giá trị RMSE ban đầu giảm nhanh, nhưng sau một số lượng biến nhất định thì RMSE không giảm nhiều hơn nữa. Sau khoảng 100 biến dự báo, đường cong RMSE không giảm nhiều nữa, thậm chí có xu hướng tăng nhẹ. Điều này gợi ý rằng việc thêm quá nhiều biến dự báo vào mô hình có thể không cải thiện đáng kể độ chính xác dự báo mà còn có thể dẫn đến hiện tượng overfitting.

```
which.min(cv_error_df)
```

```
## [1] 143
```

Giá trị nhỏ nhất của 10-fold cross-validated error xảy ra khi số lượng biến dự báo trong mô hình là 142. Đây sẽ là giá trị tối ưu về độ chính xác dự báo đối với tập dữ liệu này. Vậy nên ta sẽ sử dụng 142 biến dự báo quan trọng nhất để xây dựng mô hình mới. ### Thực hiện regsubsets() và xác định mô hình con với 142 biến hồi quy

```
out_subset_df_2 <- regsubsets(x = price ~ ., data = final_df,
                              method = "forward", nvmax = 181)
# Xác định số lượng biến hồi quy tối ưu (ở đây giả định là 142 biến hồi quy)
optimal_model_id <- which.min(cv_error_df)  # Chọn mô hình có RMSE nhỏ nhất

# Lấy các hệ số của mô hình tối ưu
optimal_model_coefficients <- coef(out_subset_df_2, id = optimal_model_id)
optimal_model_coefficients
```

```
##               (Intercept)                 bathrooms                  bedrooms
##              5.603150e+00              3.196775e-02             -2.802109e-02
##               square_feet             price_per_sqft       cityname_alpharetta
##              8.621279e-04              4.726762e-06              7.810672e-02
##             cityname_ames        cityname_ann_arbor             cityname_apex
##             -6.311531e-02              6.016722e-02             -4.182860e-02
##           cityname_aurora           cityname_austin             cityname_azle
##             -2.870083e-01              7.292638e-03              5.410154e-02
##        cityname_baltimore      cityname_baton_rouge          cityname_bedford
##             -2.471642e-02             -1.143277e-01              6.888333e-02
##         cityname_bellevue      cityname_bloomington           cityname_boston
##              4.442522e-02             -4.488089e-02             -1.315809e-01
##          cityname_bothell              cityname_cary      cityname_cedar_falls
##              1.136448e-01             -4.620085e-02             -6.386062e-02
##        cityname_champaign      cityname_chapel_hill        cityname_charlotte
##             -3.640094e-02              3.193059e-02             -1.219308e-01
##      cityname_cherry_hill          cityname_chicago          cityname_clayton
##              1.137003e-01              7.126458e-03             -8.521228e-02
## cityname_colorado_springs         cityname_columbia          cityname_concord
##              4.344043e-02             -5.806545e-02             -3.511618e-02
##       cityname_coraopolis          cityname_cumming           cityname_dallas
##              5.736932e-02              2.704639e-02              1.703386e-02
##            cityname_dayton          cityname_decatur           cityname_denver
##             -2.150941e-01             -5.535362e-02              6.818333e-02
##       cityname_des_moines          cityname_detroit          cityname_edmond
##             -7.249127e-02             -1.267613e-01             -1.076408e-01
##    cityname_ellicott_city          cityname_everett           cityname_fargo
##              1.232794e-01              8.481971e-02             -1.171997e-01
##     cityname_fayetteville      cityname_federal_way      cityname_forest_lake
```

```
##             -1.522555e-01              1.008590e-01              9.293756e-02
##       cityname_fort_worth     cityname_gaithersburg         cityname_glendale
##             -2.419299e-02              1.218187e-01             -6.157227e-02
##      cityname_grand_forks        cityname_grapevine       cityname_greensboro
##             -1.760703e-01              1.001255e-01             -7.643667e-02
##       cityname_greenville       cityname_hackensack          cityname_hanover
##             -6.736051e-02              1.156862e-01              1.480809e-01
##          cityname_houston           cityname_humble      cityname_hyattsville
##             -5.153634e-02             -1.307281e-01              1.235241e-01
##     cityname_indianapolis         cityname_issaquah     cityname_jacksonville
##             -1.485436e-01              1.581045e-01             -1.321019e-01
##      cityname_jersey_city      cityname_kansas_city             cityname_kent
##             -3.353723e-02             -5.811972e-02              8.923014e-02
##          cityname_kirkland        cityname_lafayette           cityname_laurel
##              7.755156e-02             -1.266041e-01              1.205312e-01
##        cityname_lexington          cityname_lincoln         cityname_lithonia
##             -1.013052e-01             -1.065498e-01             -4.908784e-02
##       cityname_long_beach         cityname_longview      cityname_los_angeles
##              6.122599e-02             -1.295815e-01             -1.339711e-02
##          cityname_lubbock         cityname_lynnwood          cityname_madison
##             -1.057001e-01              1.229610e-01              7.960202e-02
##       cityname_manchester           cityname_mandan        cityname_mansfield
##              8.495236e-02             -1.012388e-01              6.174568e-02
##     cityname_maple_valley            cityname_miami          cityname_midland
##              1.679883e-01             -3.214921e-02             -2.558890e-02
##          cityname_milford      cityname_minneapolis            cityname_minot
##              6.772080e-02              2.556172e-02             -1.744612e-01
##           cityname_monroe      cityname_monroeville      cityname_morrisville
##             -9.034986e-02             -4.503716e-02              2.404685e-02
##     cityname_newport_news          cityname_norfolk           cityname_norman
##             -7.557629e-02              5.729542e-02             -7.410939e-02
##          cityname_oakland          cityname_odenton    cityname_oklahoma_city
##             -1.291155e-01              1.498433e-01             -3.212768e-02
##            cityname_omaha           cityname_orange          cityname_orlando
##             -5.900030e-02              4.221983e-02             -4.691175e-02
##             citynameother     cityname_overland_park      cityname_owings_mills
##             -3.412375e-02             -7.301870e-02              7.684025e-02
##         cityname_pasadena      cityname_philadelphia          cityname_phoenix
##              3.136133e-02              6.916738e-02             -4.148845e-02
##       cityname_plainsboro         cityname_plymouth         cityname_portland
##              1.138956e-01              6.644615e-02             -2.056853e-02
##        cityname_princeton          cityname_redmond             cityname_reno
##              1.434642e-01              8.469586e-02             -6.295908e-02
##           cityname_renton        cityname_rochester        cityname_rockville
##              9.668327e-02             -4.562470e-02              1.187405e-01
##       cityname_round_rock      cityname_saint_george      cityname_saint_louis
##              2.461314e-02             -1.741556e-01             -4.568097e-02
## cityname_saint_louis_park       cityname_saint_paul cityname_saint_petersburg
##              8.209489e-02              5.198960e-02             -5.927253e-02
##    cityname_salt_lake_city      cityname_san_antonio         cityname_san_diego
##              2.847378e-02             -2.953863e-02              7.187848e-02
##     cityname_san_francisco       cityname_san_marcos     cityname_santa_monica
##             -8.281282e-01             -5.593427e-02             -1.575944e-01
##           cityname_seattle     cityname_silver_spring       cityname_sioux_falls
```

```
##              1.982596e-02              1.299988e-01             -1.144610e-01
##           cityname_spring     cityname_springfield          cityname_suwanee
##             -8.518126e-02             -1.186779e-01              8.604969e-02
##      cityname_tallahassee            cityname_tampa           cityname_toledo
##             -4.267268e-02             -1.655359e-02             -1.483703e-01
##           cityname_tucson            cityname_tulsa       cityname_tuscaloosa
##             -1.412146e-01             -2.153480e-01             -8.560158e-02
##            cityname_tyler           cityname_urbana        cityname_vancouver
##             -1.265066e-01             -5.076934e-02              3.794417e-02
##   cityname_virginia_beach       cityname_washington  cityname_west_des_moines
##              4.898191e-02             -7.341129e-02             -6.261129e-02
##   cityname_west_lafayette    cityname_west_new_york    cityname_winston_salem
##             -2.675503e-02              9.961173e-02             -5.046935e-02
```

4.3 Xây dựng model từ kết quả của cross validation

Tạo công thức hồi quy tuyến tính từ các hệ số

```
selected_variables <- names(optimal_model_coefficients)[-1]  # Loại bỏ intercept
formula_str <- paste("price ~", paste(selected_variables, collapse = " + "))
formula_optimal <- as.formula(formula_str)

cv_model <- lm(formula_optimal, data = train_data)
summary(cv_model)
```

```
##
## Call:
## lm(formula = formula_optimal, data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.10366 -0.05914  0.02460  0.09951  0.95303
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 5.603e+00  1.163e-02 481.794  < 2e-16 ***
## bathrooms                   3.061e-02  6.291e-03   4.865 1.17e-06 ***
## bedrooms                   -2.830e-02  4.408e-03  -6.420 1.46e-10 ***
## square_feet                 8.598e-04  1.097e-05  78.348  < 2e-16 ***
## price_per_sqft              4.748e-06  3.934e-08 120.699  < 2e-16 ***
## cityname_alpharetta         7.695e-02  4.152e-02   1.853 0.063867 .
## cityname_ames              -5.904e-02  2.366e-02  -2.496 0.012601 *
## cityname_ann_arbor          3.769e-02  4.597e-02   0.820 0.412349
## cityname_apex              -4.929e-02  4.596e-02  -1.073 0.283537
## cityname_aurora            -1.472e-01  6.241e-02  -2.358 0.018402 *
## cityname_austin             6.494e-03  1.099e-02   0.591 0.554507
## cityname_azle               6.209e-02  5.506e-02   1.128 0.259553
## cityname_baltimore         -1.661e-02  4.028e-02  -0.412 0.680085
## cityname_baton_rouge       -1.315e-01  3.723e-02  -3.533 0.000414 ***
## cityname_bedford            7.114e-02  3.633e-02   1.958 0.050255 .
## cityname_bellevue           2.297e-02  3.489e-02   0.658 0.510281
## cityname_bloomington       -4.615e-02  2.920e-02  -1.581 0.114040
## cityname_boston            -1.537e-01  4.679e-02  -3.285 0.001025 **
```

```
## cityname_bothell           1.233e-01  4.599e-02   2.682 0.007332 **
## cityname_cary             -4.203e-02  4.286e-02  -0.981 0.326805
## cityname_cedar_falls      -6.942e-02  5.509e-02  -1.260 0.207662
## cityname_champaign        -3.996e-02  3.020e-02  -1.323 0.185801
## cityname_chapel_hill       4.482e-02  5.506e-02   0.814 0.415665
## cityname_charlotte        -1.173e-01  3.120e-02  -3.760 0.000172 ***
## cityname_cherry_hill       1.263e-01  6.731e-02   1.876 0.060691 .
## cityname_chicago           1.526e-02  1.849e-02   0.825 0.409200
## cityname_clayton          -6.984e-02  4.301e-02  -1.624 0.104461
## cityname_colorado_springs  3.748e-02  2.333e-02   1.606 0.108296
## cityname_columbia         -5.630e-02  2.880e-02  -1.955 0.050612 .
## cityname_concord          -2.106e-02  4.435e-02  -0.475 0.634881
## cityname_coraopolis        6.793e-02  4.988e-02   1.362 0.173273
## cityname_cumming           1.410e-02  6.241e-02   0.226 0.821262
## cityname_dallas            2.337e-02  1.483e-02   1.575 0.115252
## cityname_dayton           -2.248e-01  4.035e-02  -5.571 2.64e-08 ***
## cityname_decatur          -2.980e-02  4.431e-02  -0.673 0.501255
## cityname_denver            7.158e-02  2.027e-02   3.531 0.000417 ***
## cityname_des_moines       -8.979e-02  5.842e-02  -1.537 0.124391
## cityname_detroit          -8.808e-02  5.861e-02  -1.503 0.132961
## cityname_edmond           -9.663e-02  3.281e-02  -2.946 0.003236 **
## cityname_ellicott_city     1.336e-01  4.287e-02   3.117 0.001834 **
## cityname_everett           4.458e-02  6.734e-02   0.662 0.508006
## cityname_fargo            -7.597e-02  5.839e-02  -1.301 0.193310
## cityname_fayetteville     -1.482e-01  5.516e-02  -2.687 0.007235 **
## cityname_federal_way       1.112e-01  5.838e-02   1.905 0.056870 .
## cityname_forest_lake       1.052e-01  5.520e-02   1.905 0.056816 .
## cityname_fort_worth       -1.875e-02  5.233e-02  -0.358 0.720120
## cityname_gaithersburg      1.429e-01  4.990e-02   2.864 0.004203 **
## cityname_glendale         -3.126e-02  4.597e-02  -0.680 0.496585
## cityname_grand_forks      -2.102e-01  4.161e-02  -5.052 4.51e-07 ***
## cityname_grapevine         1.066e-01  5.507e-02   1.936 0.052950 .
## cityname_greensboro       -6.253e-02  3.482e-02  -1.796 0.072552 .
## cityname_greenville       -3.410e-02  5.845e-02  -0.583 0.559666
## cityname_hackensack        9.876e-02  5.516e-02   1.791 0.073414 .
## cityname_hanover           1.400e-01  4.289e-02   3.263 0.001107 **
## cityname_houston          -4.587e-02  1.647e-02  -2.785 0.005370 **
## cityname_humble           -1.749e-01  5.847e-02  -2.991 0.002792 **
## cityname_hyattsville       1.280e-01  4.988e-02   2.566 0.010312 *
## cityname_indianapolis     -1.467e-01  3.725e-02  -3.939 8.29e-05 ***
## cityname_issaquah          1.643e-01  4.995e-02   3.289 0.001013 **
## cityname_jacksonville     -1.391e-01  4.595e-02  -3.027 0.002479 **
## cityname_jersey_city      -2.093e-02  3.487e-02  -0.600 0.548463
## cityname_kansas_city      -4.021e-02  2.128e-02  -1.890 0.058786 .
## cityname_kent              9.091e-02  3.013e-02   3.018 0.002558 **
## cityname_kirkland          9.421e-02  3.929e-02   2.398 0.016523 *
## cityname_lafayette        -1.138e-01  4.152e-02  -2.741 0.006146 **
## cityname_laurel            1.278e-01  4.281e-02   2.984 0.002857 **
## cityname_lexington        -1.227e-01  3.824e-02  -3.210 0.001333 **
## cityname_lincoln          -9.123e-02  5.232e-02  -1.744 0.081279 .
## cityname_lithonia         -6.049e-02  4.783e-02  -1.265 0.206071
## cityname_long_beach        6.293e-02  3.368e-02   1.868 0.061743 .
## cityname_longview         -1.124e-01  5.842e-02  -1.925 0.054307 .
## cityname_los_angeles      -1.935e-02  1.984e-02  -0.975 0.329404
```

```
## cityname_lubbock          -8.489e-02  4.599e-02  -1.846 0.064947 .
## cityname_lynnwood          1.129e-01  3.553e-02   3.178 0.001488 **
## cityname_madison           7.220e-02  1.955e-02   3.693 0.000223 ***
## cityname_manchester        9.434e-02  3.341e-02   2.824 0.004758 **
## cityname_mandan           -9.473e-02  6.241e-02  -1.518 0.129105
## cityname_mansfield         6.288e-02  4.594e-02   1.369 0.171101
## cityname_maple_valley      1.725e-01  4.444e-02   3.882 0.000105 ***
## cityname_miami            -6.138e-02  6.243e-02  -0.983 0.325568
## cityname_midland          -3.002e-02  5.508e-02  -0.545 0.585772
## cityname_milford           9.278e-02  5.840e-02   1.589 0.112202
## cityname_minneapolis       3.647e-03  2.741e-02   0.133 0.894131
## cityname_minot            -1.711e-01  3.066e-02  -5.580 2.51e-08 ***
## cityname_monroe           -8.412e-02  5.242e-02  -1.605 0.108607
## cityname_monroeville      -4.240e-02  4.781e-02  -0.887 0.375176
## cityname_morrisville       2.311e-02  4.990e-02   0.463 0.643326
## cityname_newport_news     -8.489e-02  7.369e-02  -1.152 0.249359
## cityname_norfolk           4.174e-02  4.777e-02   0.874 0.382300
## cityname_norman           -6.089e-02  2.933e-02  -2.076 0.037933 *
## cityname_oakland          -1.128e-01  4.501e-02  -2.506 0.012237 *
## cityname_odenton           1.408e-01  6.241e-02   2.256 0.024083 *
## cityname_oklahoma_city    -3.959e-02  2.803e-02  -1.413 0.157826
## cityname_omaha            -6.688e-02  2.497e-02  -2.678 0.007417 **
## cityname_orange            4.818e-02  4.781e-02   1.008 0.313549
## cityname_orlando          -5.874e-02  3.720e-02  -1.579 0.114435
## citynameother             -3.313e-02  6.890e-03  -4.809 1.55e-06 ***
## cityname_overland_park    -7.692e-02  6.733e-02  -1.142 0.253335
## cityname_owings_mills      1.028e-01  6.731e-02   1.527 0.126783
## cityname_pasadena          1.928e-02  3.258e-02   0.592 0.554062
## cityname_philadelphia      9.363e-02  3.340e-02   2.803 0.005082 **
## cityname_phoenix          -4.383e-02  2.764e-02  -1.586 0.112871
## cityname_plainsboro        1.176e-01  4.431e-02   2.655 0.007962 **
## cityname_plymouth          6.554e-02  4.989e-02   1.314 0.189023
## cityname_portland         -3.604e-02  2.184e-02  -1.650 0.098969 .
## cityname_princeton         1.464e-01  4.609e-02   3.176 0.001500 **
## cityname_redmond           1.068e-01  5.843e-02   1.828 0.067554 .
## cityname_reno              1.859e-02  4.596e-02   0.405 0.685833
## cityname_renton            1.029e-01  3.061e-02   3.363 0.000775 ***
## cityname_rochester        -3.883e-02  4.989e-02  -0.778 0.436391
## cityname_rockville         1.173e-01  4.991e-02   2.350 0.018821 *
## cityname_round_rock        2.628e-02  3.816e-02   0.689 0.491012
## cityname_saint_george     -1.468e-01  9.509e-02  -1.544 0.122749
## cityname_saint_louis      -7.167e-02  2.922e-02  -2.453 0.014210 *
## cityname_saint_louis_park  5.981e-02  4.990e-02   1.199 0.230748
## cityname_saint_paul        4.964e-02  4.153e-02   1.195 0.231972
## cityname_saint_petersburg -7.813e-02  5.508e-02  -1.418 0.156131
## cityname_salt_lake_city    3.724e-02  3.552e-02   1.048 0.294555
## cityname_san_antonio      -3.601e-02  1.664e-02  -2.164 0.030465 *
## cityname_san_diego         8.144e-02  2.867e-02   2.841 0.004515 **
## cityname_san_francisco    -8.120e-01  3.102e-02 -26.177  < 2e-16 ***
## cityname_san_marcos        2.139e-02  6.236e-02   0.343 0.731601
## cityname_santa_monica     -1.781e-01  5.910e-02  -3.013 0.002593 **
## cityname_seattle           7.262e-03  2.527e-02   0.287 0.773868
## cityname_silver_spring     1.392e-01  3.815e-02   3.648 0.000266 ***
## cityname_sioux_falls      -1.175e-01  2.848e-02  -4.128 3.71e-05 ***
```

```
## cityname_spring           -1.261e-01  5.239e-02  -2.406 0.016154 *
## cityname_springfield      -1.313e-01  3.553e-02  -3.695 0.000222 ***
## cityname_suwanee           6.847e-02  5.508e-02   1.243 0.213817
## cityname_tallahassee      -2.025e-02  4.989e-02  -0.406 0.684790
## cityname_tampa             1.007e-03  3.555e-02   0.028 0.977412
## cityname_toledo           -1.661e-01  4.994e-02  -3.327 0.000884 ***
## cityname_tucson           -9.582e-02  4.598e-02  -2.084 0.037218 *
## cityname_tulsa            -2.612e-01  4.034e-02  -6.474 1.03e-10 ***
## cityname_tuscaloosa       -7.398e-02  4.445e-02  -1.664 0.096085 .
## cityname_tyler            -9.534e-02  6.240e-02  -1.528 0.126600
## cityname_urbana           -3.253e-02  4.156e-02  -0.783 0.433844
## cityname_vancouver         2.742e-02  3.721e-02   0.737 0.461213
## cityname_virginia_beach    6.504e-02  5.227e-02   1.244 0.213388
## cityname_washington       -5.453e-02  2.898e-02  -1.882 0.059903 .
## cityname_west_des_moines  -9.512e-02  6.736e-02  -1.412 0.157971
## cityname_west_lafayette   -1.305e-03  2.678e-02  -0.049 0.961156
## cityname_west_new_york     9.100e-02  5.868e-02   1.551 0.120976
## cityname_winston_salem    -4.356e-02  5.228e-02  -0.833 0.404746
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1642 on 6132 degrees of freedom
## Multiple R-squared:  0.8511, Adjusted R-squared:  0.8476
## F-statistic: 245.1 on 143 and 6132 DF,  p-value: < 2.2e-16
```

Mô hình hồi quy tuyến tính với các biến tối ưu có độ chính xác cao, với Multiple R-squared là 0.8511 và Adjusted R-squared là 0.8476, giải thích được khoảng 85% biến thiên của giá thuê căn hộ. Giá trị F-statistic lớn và p-value rất nhỏ ($< 2.2e\text{-}16$) cho thấy mô hình có ý nghĩa thống kê cao. Residual standard error (RSE) là 0.1642, cho thấy sai số dự đoán trung bình rất thấp.

```
# Dự đoán trên tập test
predictions <- predict(cv_model, newdata = test_data)
RMSE <- sqrt(mean((test_data$price - predictions)^2))
cat("Root Mean Squared Error (RMSE):", RMSE, "\n")
```

```
## Root Mean Squared Error (RMSE): 0.1656222
```

RMSE (Root Mean Squared Error) trên tập kiểm tra là 0.1656222. Đây là một giá trị khá thấp, cho thấy độ chính xác dự đoán của mô hình là tốt. Với RMSE thấp trên cả tập huấn luyện và tập kiểm tra, ta có thể kết luận rằng mô hình hồi quy tuyến tính với 142 biến dự báo tối ưu đã được xây dựng khá tốt và có khả năng dự đoán chính xác trên dữ liệu mới.

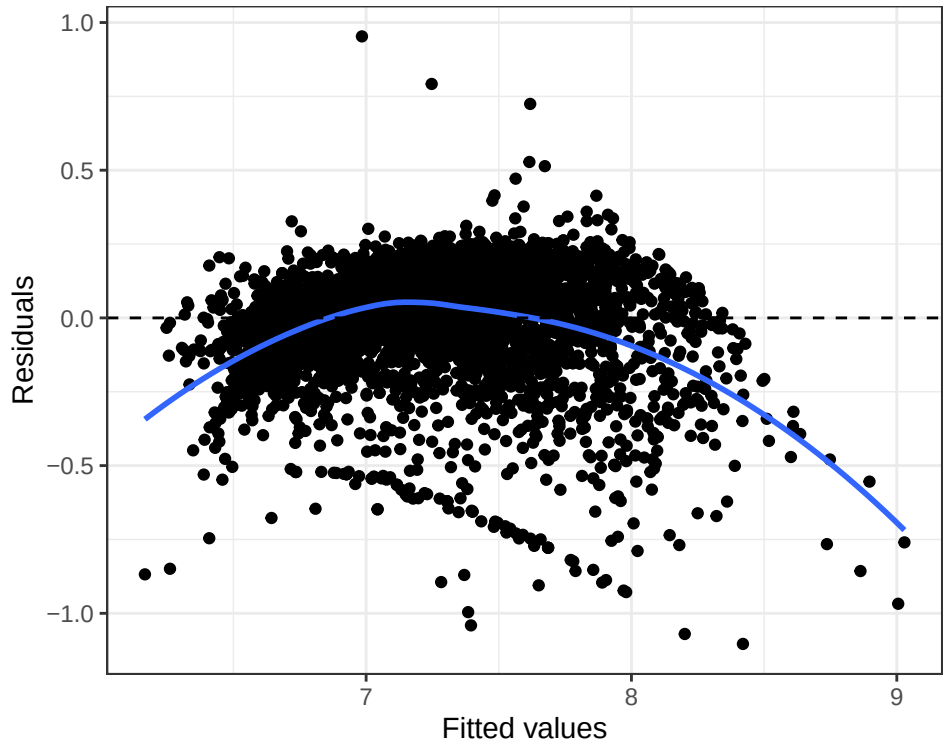
5. Chuẩn đoán thặng dư của mô hình

Dựa vào mô hình cross validation, ta tiến hành chuẩn đoán thặng dư của mô hình để đánh giá hiệu suất một cách cụ thể hơn.

5.1 Biểu đồ thặng dư của mô hình

```
ggplot(data = cv_model, mapping = aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_smooth(method = "loess", se = FALSE) +
  geom_hline(yintercept = 0, linetype = "dashed") +
  labs(x = "Fitted values", y = "Residuals") +
  theme_bw()
```

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



Biểu đồ thặng dư cho thấy các điểm dữ liệu (phần dư) được phân tán ngẫu nhiên xung quanh đường thẳng ngang ở $y = 0$. Điều này cho thấy mô hình đã được xây dựng phù hợp và không vi phạm các giả định cơ bản của hồi quy tuyến tính. Đường cong trơn (smooth line) được vẽ qua các điểm dữ liệu không có xu hướng lên hoặc xuống, chứng tỏ mô hình không có sai sót hệ thống. Độ phân tán của các điểm dữ liệu xung quanh đường $y = 0$ tương đối đồng đều, không có xu hướng thay đổi theo giá trị dự đoán (fitted values).

5.2 Kiểm tra tính tuyến tính từng phần

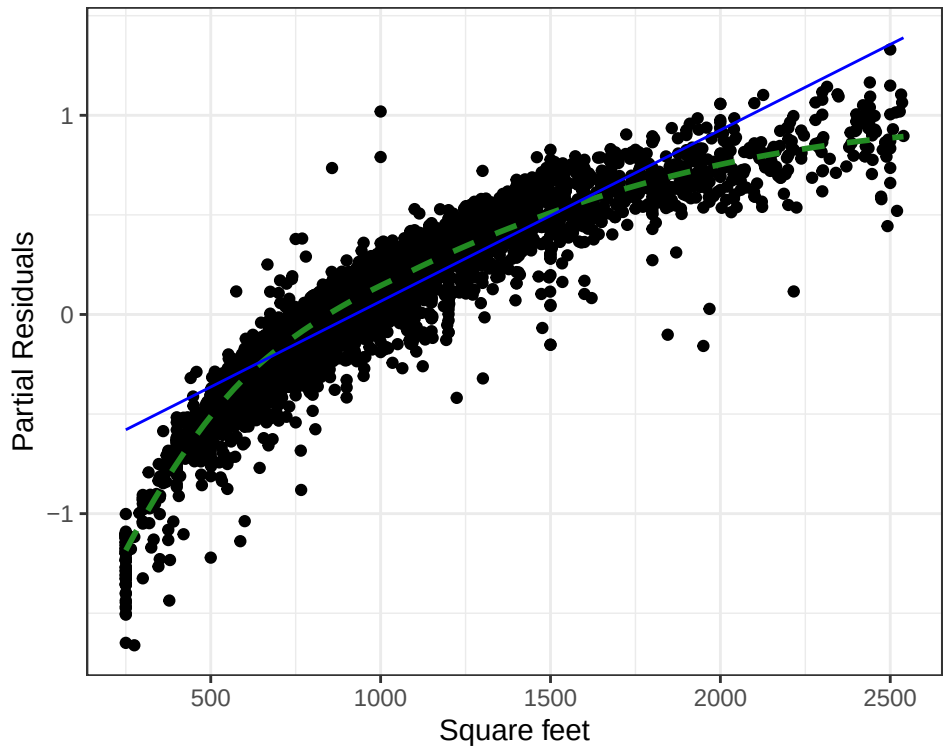
```
terms_df <- predict(cv_model, type = "terms")
partial_resid_df <- residuals(cv_model, type = "partial")
```

Ta sẽ đánh giá tính tuyến tính của thành phần square_feet trong mô hình:

```
data_part_resid_square_feet_df <- tibble(
  square_feet = train_data$square_feet,
  terms_square_feet = terms_df[, "square_feet"],
```

```
  partial_resid_square_feet = partial_resid_df[, "square_feet"]
)

ggplot(data_part_resid_square_feet_df, mapping = aes(square_feet, partial_resid_square_feet)) +
  geom_point() +
  geom_smooth(method = "loess", se = FALSE, linetype = "dashed",
              color = "forestgreen") +
  geom_line(aes(x = square_feet, y = terms_square_feet), color = "blue") +
  labs(x = "Square feet", y = "Partial Residuals") +
  theme_bw()
```

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

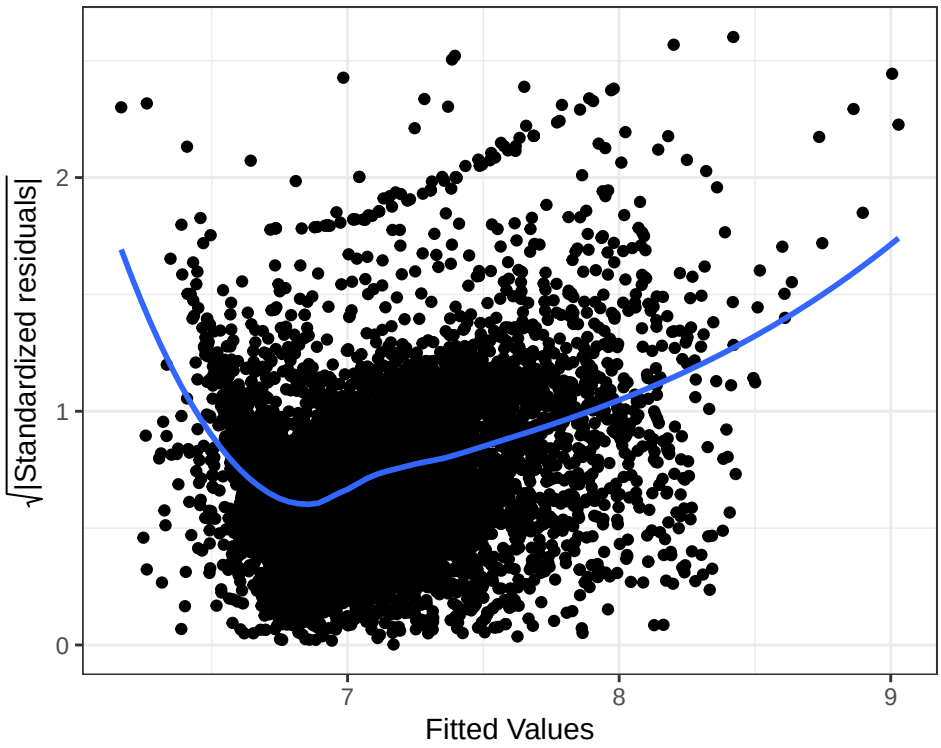


Đường cong trơn (dashed green line) có xu hướng tăng dần khi diện tích căn hộ (square feet) tăng lên, cho thấy mối quan hệ giữa diện tích căn hộ và biến phụ thuộc không hoàn toàn tuyến tính, gợi ý rằng mô hình hồi quy tuyến tính hiện tại chưa phù hợp hoàn toàn.

5.3 Kiểm tra tính đồng nhất phương sai

```
ggplot(cv_model, aes(.fitted, sqrt(abs(.stdresid)))) +
  geom_point(na.rm = TRUE) +
  geom_smooth(method = "loess", na.rm = TRUE, se = FALSE) +
  labs(x = "Fitted Values", y = expression(sqrt("|Standardized residuals|"))) +
  theme_bw()
```

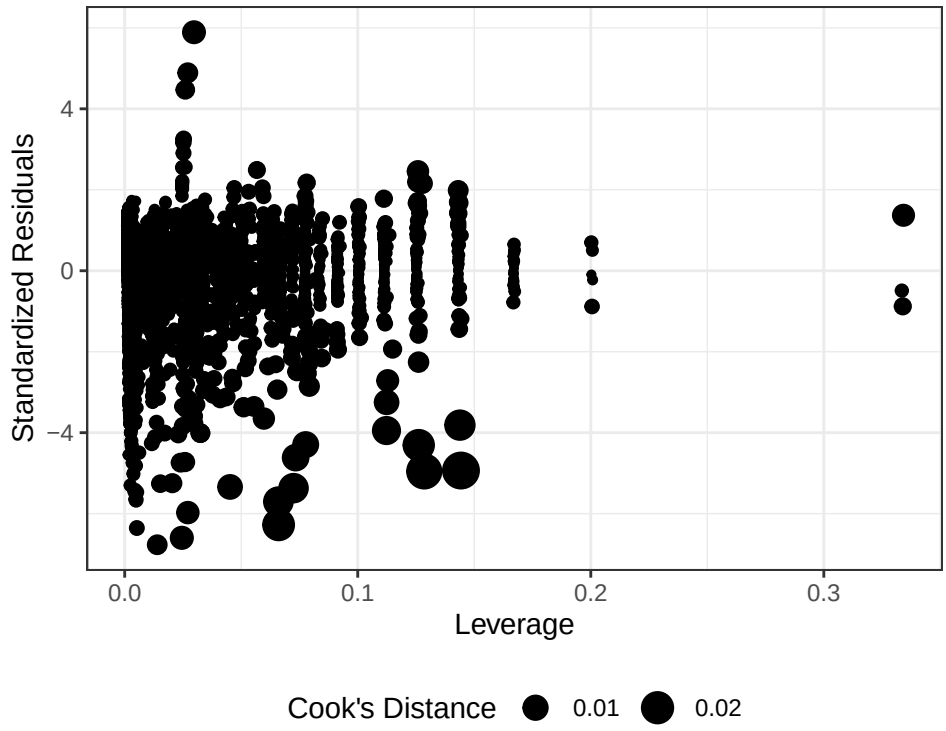
```
## `geom_smooth()` using formula = 'y ~ x'
```



Mối quan hệ giữa các giá trị được ước lượng (Fitted Values) và giá trị của căn bậc hai giá trị tuyệt đối của các residual chuẩn hóa (sqrt(|Standardized residuals|)) không hoàn toàn ổn định (không đồng nhất). Có một trend rõ ràng, với giá trị căn bậc hai của residual chuẩn hóa giảm dần khi các giá trị được ước lượng tăng lên. Điều này gợi ý rằng phương sai của residual có thể không đồng nhất, mà phụ thuộc vào giá trị được ước lượng.

5.4 Kiểm tra điểm ngoại lai trong mô hình

```
ggplot(cv_model, aes(.hat, .stdresid)) +
  geom_point(aes(size = .cooksd)) +
  xlab("Leverage") + ylab("Standardized Residuals") +
  scale_size_continuous("Cook's Distance", range = c(1, 6)) +
  theme_bw() +
  theme(legend.position = "bottom")
```



Có một số quan sát có leverage và/hoặc ảnh hưởng lớn (Cook's distance lớn) so với các quan sát khác. Những quan sát này có thể là điểm ngoại lai và cần được kiểm tra kỹ lưỡng. Hầu hết các quan sát nằm trong phạm vi của mô hình, với standardized residuals và leverage ở mức độ chấp nhận được. Một số quan sát có standardized residuals lớn (nằm xa trục ngang), nhưng leverage không quá lớn.

```
std_resid_df <- rstandard(cv_model)
hat_values_df <- hatvalues(cv_model)
cooks_D_df <- cooks.distance(cv_model)

data_cooks_df <- tibble(id_point = 1:nrow(train_data),
                        rstand = std_resid_df, hats = hat_values_df,
                        cooks = cooks_D_df, sales = train_data$price)
data_cooks_df |> arrange(desc(cooks))
```

```
## # A tibble: 6,276 x 5
##    id_point rstand   hats  cooks sales
##       <int>  <dbl>  <dbl>  <dbl> <dbl>
##  1     2545  -4.94 0.144  0.0285  6.91
##  2     4446  -4.96 0.129  0.0252  8.27
##  3     5897  -6.28 0.0661 0.0193  6.39
##  4     4445  -4.31 0.126  0.0186  7.59
##  5     2401  -3.81 0.144  0.0170  6.80
##  6     4444  -5.70 0.0660 0.0160  6.75
##  7     4053  -5.37 0.0725 0.0156  5.41
##  8     2730  -3.95 0.112  0.0137  6.75
##  9     2591  -4.62 0.0734 0.0117  6.84
## 10     2808  -4.29 0.0777 0.0108  5.97
## # i 6,266 more rows
```

```r
data_cooks_df_sorted <- data_cooks_df |> arrange(desc(cooks))
outliers <- data_cooks_df_sorted[data_cooks_df_sorted$cooks > 4/5044, ]
print(outliers)
```

```
## # A tibble: 221 x 5
##    id_point rstand   hats  cooks sales
##       <int>  <dbl>  <dbl>  <dbl> <dbl>
##  1     2545  -4.94 0.144  0.0285  6.91
##  2     4446  -4.96 0.129  0.0252  8.27
##  3     5897  -6.28 0.0661 0.0193  6.39
##  4     4445  -4.31 0.126  0.0186  7.59
##  5     2401  -3.81 0.144  0.0170  6.80
##  6     4444  -5.70 0.0660 0.0160  6.75
##  7     4053  -5.37 0.0725 0.0156  5.41
##  8     2730  -3.95 0.112  0.0137  6.75
##  9     2591  -4.62 0.0734 0.0117  6.84
## 10     2808  -4.29 0.0777 0.0108  5.97
## # i 211 more rows
```

6. Mở rộng mô hình hồi quy

Dựa vào kết quả từ biểu đồ cooks, ta sẽ thử loại bỏ các điểm ngoại lai từ train_data để xây dựng mô hình dựa trên dữ liệu mới này và đánh giá hiệu suất.

```r
# Thêm cột chỉ số (index) vào train_data
train_data$index <- seq_len(nrow(train_data))

# Loại bỏ các điểm ngoại lai từ train_data
train_data_clean <- train_data[!train_data$index %in% outliers$id_point, ]

# Xóa cột index
train_data_clean <- train_data_clean[, -ncol(train_data_clean)]

# Xây dựng model dựa trên tập train đã loại bỏ outliers
lm_model_clean <- lm(formula_optimal, data = train_data_clean)
summary(lm_model_clean)
```

```
##
## Call:
## lm(formula = formula_optimal, data = train_data_clean)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.89909 -0.05391  0.02196  0.08777  0.35499
##
## Coefficients: (1 not defined because of singularities)
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 5.527e+00  1.038e-02 532.743  < 2e-16 ***
## bathrooms                   2.755e-02  5.629e-03   4.893 1.02e-06 ***
## bedrooms                   -2.782e-02  3.909e-03  -7.118 1.23e-12 ***
## square_feet                 8.899e-04  1.002e-05  88.776  < 2e-16 ***
```

```
## price_per_sqft             5.103e-06  3.592e-08 142.087  < 2e-16 ***
## cityname_alpharetta        7.251e-02  3.584e-02   2.023 0.043078 *
## cityname_ames             -4.504e-02  2.043e-02  -2.205 0.027490 *
## cityname_ann_arbor         5.423e-02  4.126e-02   1.314 0.188801
## cityname_apex             -2.627e-03  4.514e-02  -0.058 0.953592
## cityname_aurora           -2.189e-01  1.419e-01  -1.543 0.122772
## cityname_austin           -1.948e-03  9.501e-03  -0.205 0.837576
## cityname_azle              6.512e-02  4.752e-02   1.370 0.170667
## cityname_baltimore         1.730e-02  3.581e-02   0.483 0.629009
## cityname_baton_rouge      -8.897e-02  3.383e-02  -2.629 0.008573 **
## cityname_bedford           7.118e-02  3.136e-02   2.270 0.023244 *
## cityname_bellevue          3.331e-02  3.077e-02   1.083 0.278966
## cityname_bloomington      -2.405e-02  2.597e-02  -0.926 0.354445
## cityname_boston           -2.279e-01  6.406e-02  -3.558 0.000376 ***
## cityname_bothell           1.281e-01  4.128e-02   3.104 0.001917 **
## cityname_cary             -1.709e-02  3.825e-02  -0.447 0.655007
## cityname_cedar_falls      -5.736e-02  4.755e-02  -1.206 0.227768
## cityname_champaign        -2.219e-02  2.647e-02  -0.838 0.401945
## cityname_chapel_hill       4.355e-02  4.752e-02   0.916 0.359533
## cityname_charlotte        -9.214e-02  2.785e-02  -3.309 0.000943 ***
## cityname_cherry_hill       1.172e-01  5.809e-02   2.017 0.043702 *
## cityname_chicago          -3.984e-03  1.606e-02  -0.248 0.804077
## cityname_clayton          -3.293e-02  3.981e-02  -0.827 0.408140
## cityname_colorado_springs  4.017e-02  2.014e-02   1.994 0.046163 *
## cityname_columbia         -1.295e-02  2.598e-02  -0.498 0.618156
## cityname_concord           1.740e-02  4.128e-02   0.422 0.673400
## cityname_coraopolis        7.160e-02  4.305e-02   1.663 0.096313 .
## cityname_cumming           1.679e-02  5.387e-02   0.312 0.755216
## cityname_dallas            2.400e-02  1.285e-02   1.868 0.061821 .
## cityname_dayton           -1.640e-01  3.832e-02  -4.281 1.89e-05 ***
## cityname_decatur           3.544e-02  3.966e-02   0.894 0.371487
## cityname_denver            7.450e-02  1.773e-02   4.202 2.68e-05 ***
## cityname_des_moines       -1.003e-01  5.388e-02  -1.862 0.062637 .
## cityname_detroit          -2.010e-01  8.210e-02  -2.448 0.014382 *
## cityname_edmond           -8.686e-02  2.832e-02  -3.067 0.002173 **
## cityname_ellicott_city     1.164e-01  3.701e-02   3.144 0.001675 **
## cityname_everett           5.058e-02  6.362e-02   0.795 0.426631
## cityname_fargo            -5.825e-02  5.811e-02  -1.002 0.316168
## cityname_fayetteville     -1.606e-01  5.815e-02  -2.762 0.005766 **
## cityname_federal_way       1.062e-01  5.038e-02   2.108 0.035049 *
## cityname_forest_lake       1.085e-01  4.765e-02   2.276 0.022852 *
## cityname_fort_worth       -1.751e-02  4.517e-02  -0.388 0.698253
## cityname_gaithersburg      1.261e-01  4.307e-02   2.928 0.003429 **
## cityname_glendale         -4.654e-02  5.038e-02  -0.924 0.355617
## cityname_grand_forks      -1.337e-01  3.832e-02  -3.488 0.000490 ***
## cityname_grapevine         1.034e-01  4.753e-02   2.176 0.029582 *
## cityname_greensboro       -5.134e-02  3.005e-02  -1.708 0.087643 .
## cityname_greenville       -4.428e-02  5.393e-02  -0.821 0.411686
## cityname_hackensack        9.005e-02  5.046e-02   1.785 0.074385 .
## cityname_hanover           1.169e-01  3.703e-02   3.156 0.001606 **
## cityname_houston          -3.857e-02  1.422e-02  -2.713 0.006695 **
## cityname_humble           -1.596e-01  6.368e-02  -2.507 0.012197 *
## cityname_hyattsville       1.138e-01  4.305e-02   2.642 0.008261 **
## cityname_indianapolis     -1.187e-01  3.295e-02  -3.601 0.000320 ***
```

```
## cityname_issaquah          1.354e-01  4.312e-02   3.140 0.001697 **
## cityname_jacksonville     -7.974e-02  4.754e-02  -1.677 0.093516 .
## cityname_jersey_city      -8.169e-02  3.143e-02  -2.599 0.009360 **
## cityname_kansas_city      -1.744e-02  1.850e-02  -0.943 0.345838
## cityname_kent              8.578e-02  2.642e-02   3.246 0.001175 **
## cityname_kirkland          6.390e-02  3.393e-02   1.884 0.059679 .
## cityname_lafayette        -8.249e-02  3.699e-02  -2.230 0.025785 *
## cityname_laurel            1.214e-01  3.695e-02   3.287 0.001020 **
## cityname_lexington        -1.164e-01  4.307e-02  -2.703 0.006887 **
## cityname_lincoln          -6.508e-02  5.385e-02  -1.209 0.226883
## cityname_lithonia         -2.936e-02  4.308e-02  -0.682 0.495520
## cityname_long_beach        4.964e-02  3.028e-02   1.639 0.101213
## cityname_longview         -9.375e-02  5.043e-02  -1.859 0.063062 .
## cityname_los_angeles      -4.826e-02  1.741e-02  -2.772 0.005591 **
## cityname_lubbock          -7.990e-02  3.969e-02  -2.013 0.044163 *
## cityname_lynnwood          9.936e-02  3.067e-02   3.239 0.001204 **
## cityname_madison           7.713e-02  1.707e-02   4.519 6.34e-06 ***
## cityname_manchester        1.022e-01  2.941e-02   3.475 0.000514 ***
## cityname_mandan           -5.392e-02  5.811e-02  -0.928 0.353550
## cityname_mansfield         8.250e-02  4.124e-02   2.000 0.045515 *
## cityname_maple_valley      1.525e-01  3.837e-02   3.974 7.16e-05 ***
## cityname_miami             1.467e-02  1.418e-01   0.103 0.917649
## cityname_midland           5.315e-02  5.812e-02   0.915 0.360435
## cityname_milford           8.279e-02  5.041e-02   1.642 0.100576
## cityname_minneapolis       4.770e-03  2.428e-02   0.196 0.844261
## cityname_minot            -1.358e-01  2.691e-02  -5.046 4.65e-07 ***
## cityname_monroe           -1.325e-01  5.052e-02  -2.622 0.008763 **
## cityname_monroeville      -2.827e-02  4.127e-02  -0.685 0.493421
## cityname_morrisville       3.065e-02  4.307e-02   0.712 0.476658
## cityname_newport_news     -6.333e-02  8.202e-02  -0.772 0.440089
## cityname_norfolk           5.731e-02  4.304e-02   1.331 0.183078
## cityname_norman            1.362e-02  2.690e-02   0.506 0.612603
## cityname_oakland          -1.363e-01  4.799e-02  -2.840 0.004520 **
## cityname_odenton           1.271e-01  5.386e-02   2.359 0.018336 *
## cityname_oklahoma_city    -3.157e-02  2.419e-02  -1.305 0.191959
## cityname_omaha            -5.592e-02  2.155e-02  -2.594 0.009503 **
## cityname_orange            1.019e-01  5.045e-02   2.021 0.043351 *
## cityname_orlando          -4.653e-03  3.580e-02  -0.130 0.896586
## citynameother             -3.712e-02  5.956e-03  -6.233 4.89e-10 ***
## cityname_overland_park    -6.601e-02  5.811e-02  -1.136 0.256076
## cityname_owings_mills      1.012e-01  5.809e-02   1.742 0.081623 .
## cityname_pasadena         -2.235e-03  2.915e-02  -0.077 0.938887
## cityname_philadelphia      1.004e-01  2.940e-02   3.416 0.000640 ***
## cityname_phoenix           2.045e-02  2.520e-02   0.812 0.417108
## cityname_plainsboro        1.037e-01  3.825e-02   2.711 0.006738 **
## cityname_plymouth          7.004e-02  4.753e-02   1.474 0.140643
## cityname_portland         -5.021e-02  1.900e-02  -2.642 0.008264 **
## cityname_princeton         1.116e-01  3.980e-02   2.805 0.005054 **
## cityname_redmond           1.055e-01  5.387e-02   1.958 0.050300 .
## cityname_reno              8.548e-02  4.305e-02   1.986 0.047115 *
## cityname_renton            8.576e-02  2.643e-02   3.245 0.001180 **
## cityname_rochester        -3.055e-02  4.306e-02  -0.710 0.478021
## cityname_rockville         1.206e-01  4.515e-02   2.672 0.007563 **
## cityname_round_rock        3.301e-02  3.294e-02   1.002 0.316244
```

```
## cityname_saint_george             NA         NA      NA       NA
## cityname_saint_louis      -4.997e-02  2.559e-02  -1.953 0.050926 .
## cityname_saint_louis_park  7.232e-02  4.514e-02   1.602 0.109166
## cityname_saint_paul        3.435e-02  3.585e-02   0.958 0.338023
## cityname_saint_petersburg -1.874e-02  5.383e-02  -0.348 0.727715
## cityname_salt_lake_city    3.605e-02  3.066e-02   1.176 0.239746
## cityname_san_antonio      -3.113e-02  1.436e-02  -2.168 0.030234 *
## cityname_san_diego         7.565e-02  2.543e-02   2.974 0.002947 **
## cityname_san_francisco    -9.618e-01  3.485e-02 -27.600  < 2e-16 ***
## cityname_san_marcos       -3.067e-02  6.361e-02  -0.482 0.629717
## cityname_santa_monica     -2.037e-01  1.007e-01  -2.023 0.043088 *
## cityname_seattle           2.920e-02  2.248e-02   1.299 0.193982
## cityname_silver_spring     1.275e-01  3.293e-02   3.871 0.000109 ***
## cityname_sioux_falls      -9.967e-02  2.459e-02  -4.054 5.10e-05 ***
## cityname_spring           -1.116e-01  5.047e-02  -2.212 0.027008 *
## cityname_springfield      -1.030e-01  3.296e-02  -3.126 0.001778 **
## cityname_suwanee           1.060e-01  5.039e-02   2.103 0.035506 *
## cityname_tallahassee      -7.880e-03  4.306e-02  -0.183 0.854815
## cityname_tampa             9.243e-03  3.138e-02   0.295 0.768347
## cityname_toledo           -1.236e-01  4.517e-02  -2.736 0.006232 **
## cityname_tucson           -1.207e-01  4.310e-02  -2.801 0.005109 **
## cityname_tulsa            -1.868e-01  3.975e-02  -4.698 2.69e-06 ***
## cityname_tuscaloosa       -4.201e-02  4.527e-02  -0.928 0.353478
## cityname_tyler            -1.201e-01  5.815e-02  -2.065 0.038980 *
## cityname_urbana           -2.577e-02  3.828e-02  -0.673 0.500893
## cityname_vancouver         3.697e-02  3.293e-02   1.123 0.261607
## cityname_virginia_beach    6.919e-02  4.511e-02   1.534 0.125132
## cityname_washington       -7.971e-02  2.572e-02  -3.099 0.001949 **
## cityname_west_des_moines  -7.575e-02  5.814e-02  -1.303 0.192676
## cityname_west_lafayette   -5.547e-04  2.313e-02  -0.024 0.980868
## cityname_west_new_york     3.497e-02  5.067e-02   0.690 0.490188
## cityname_winston_salem    -3.118e-02  5.038e-02  -0.619 0.536022
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1417 on 5912 degrees of freedom
## Multiple R-squared:  0.8853, Adjusted R-squared:  0.8825
## F-statistic: 321.2 on 142 and 5912 DF,  p-value: < 2.2e-16
```

Khi xây dựng mô hình hồi quy tuyến tính mới trên tập train_data_clean, ta thấy:

- Residual standard error giảm từ 0.1427 xuống 0.1417, chỉ ra mô hình dự báo tốt hơn.
- Multiple R-squared tăng từ 0.8797 lên 0.8853, tức mô hình giải thích được nhiều phần biến thiên của biến phụ thuộc hơn.
- Adjusted R-squared tăng từ 0.8768 lên 0.8825, chỉ ra mô hình có khả năng tổng quát hóa tốt hơn.
- F-statistic cũng tăng, chứng tỏ mô hình có ý nghĩa thống kê tốt hơn.

Như vậy, việc loại bỏ các outliers đã giúp cải thiện đáng kể hiệu suất của mô hình hồi quy tuyến tính. Bên cạnh đó, thông qua biểu đồ thặng dư từng phần cho biến **square_feet**, ước lượng tuyến tính là không phù hợp với dữ liệu. Vậy ta có thể dùng tập dữ liệu này mở rộng thành phần **square_feet** lên bậc 2 để ước lượng mô hình:

```r
lm_model_poly <- lm(price ~ poly(square_feet, 2) + ., data = train_data_clean)
summary(lm_model_poly)
```

```
##
## Call:
## lm(formula = price ~ poly(square_feet, 2) + ., data = train_data_clean)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.68649 -0.03723  0.01657  0.05987  0.56900
##
## Coefficients: (2 not defined because of singularities)
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                6.353e+00  4.605e-02 137.963  < 2e-16 ***
## poly(square_feet, 2)1      2.853e+01  2.136e-01 133.542  < 2e-16 ***
## poly(square_feet, 2)2     -8.131e+00  1.095e-01 -74.284  < 2e-16 ***
## bathrooms                 -3.040e-03  4.104e-03  -0.741 0.458798
## bedrooms                  -4.042e-02  2.841e-03 -14.229  < 2e-16 ***
## square_feet                       NA         NA      NA       NA
## price_per_sqft             5.503e-06  2.704e-08 203.527  < 2e-16 ***
## cityname_alexandria        6.585e-02  5.688e-02   1.158 0.246967
## cityname_alpharetta       -1.009e-02  5.225e-02  -0.193 0.846922
## cityname_ames             -5.524e-02  4.772e-02  -1.158 0.247066
## cityname_anaheim          -2.748e-02  5.317e-02  -0.517 0.605328
## cityname_anchorage         2.855e-02  5.096e-02   0.560 0.575315
## cityname_ann_arbor         1.915e-02  5.427e-02   0.353 0.724240
## cityname_apex             -5.437e-02  5.584e-02  -0.974 0.330245
## cityname_arlington        -2.334e-02  4.745e-02  -0.492 0.622745
## cityname_atlanta           1.552e-02  4.903e-02   0.317 0.751574
## cityname_aurora           -1.719e-02  1.117e-01  -0.154 0.877652
## cityname_austin            3.050e-02  4.595e-02   0.664 0.506846
## cityname_azle              9.667e-03  5.685e-02   0.170 0.864986
## cityname_baltimore         2.371e-02  5.223e-02   0.454 0.649857
## cityname_baton_rouge      -8.184e-02  5.153e-02  -1.588 0.112297
## cityname_bedford           2.086e-02  5.073e-02   0.411 0.680936
## cityname_bellevue         -4.764e-03  5.057e-02  -0.094 0.924942
## cityname_bismarck         -5.596e-02  5.821e-02  -0.961 0.336463
## cityname_bloomington       1.822e-02  4.912e-02   0.371 0.710742
## cityname_boise             1.361e-02  5.499e-02   0.248 0.804520
## cityname_boston           -1.751e-01  6.477e-02  -2.703 0.006886 **
## cityname_bothell           6.038e-02  5.429e-02   1.112 0.266141
## cityname_burlington       -4.957e-02  5.811e-02  -0.853 0.393601
## cityname_cary             -5.891e-02  5.311e-02  -1.109 0.267434
## cityname_cedar_falls      -5.719e-02  5.684e-02  -1.006 0.314397
## cityname_cedar_park        2.475e-02  5.809e-02   0.426 0.670067
## cityname_champaign         3.343e-02  4.924e-02   0.679 0.497213
## cityname_chapel_hill       1.994e-02  5.684e-02   0.351 0.725697
## cityname_charlotte        -6.630e-02  4.965e-02  -1.335 0.181814
## cityname_cherry_hill       3.691e-02  6.173e-02   0.598 0.549885
## cityname_chicago           1.367e-02  4.690e-02   0.292 0.770678
## cityname_cincinnati       -1.522e-02  4.753e-02  -0.320 0.748899
## cityname_clayton          -6.498e-02  5.370e-02  -1.210 0.226306
## cityname_cleveland        -6.203e-02  5.364e-02  -1.156 0.247536
```

```
## cityname_colorado_springs  2.998e-02  4.767e-02   0.629 0.529514
## cityname_columbia         -4.796e-02  4.913e-02  -0.976 0.328970
## cityname_columbus          6.045e-03  4.963e-02   0.122 0.903064
## cityname_concord          -3.795e-02  5.428e-02  -0.699 0.484501
## cityname_coraopolis        3.266e-02  5.496e-02   0.594 0.552310
## cityname_cumming          -7.493e-02  5.973e-02  -1.254 0.209710
## cityname_dallas            1.367e-02  4.637e-02   0.295 0.768095
## cityname_davenport         1.890e-02  5.810e-02   0.325 0.744933
## cityname_dayton           -2.049e-01  5.310e-02  -3.859 0.000115 ***
## cityname_decatur          -2.924e-02  5.364e-02  -0.545 0.585644
## cityname_denver            4.993e-02  4.720e-02   1.058 0.290137
## cityname_des_moines       -5.968e-02  5.967e-02  -1.000 0.317305
## cityname_detroit          -1.340e-01  7.452e-02  -1.799 0.072111 .
## cityname_durham           -2.919e-02  4.963e-02  -0.588 0.556423
## cityname_edmond           -1.037e-01  4.978e-02  -2.083 0.037298 *
## cityname_ellicott_city     2.543e-02  5.270e-02   0.483 0.629451
## cityname_euless           -2.616e-02  4.882e-02  -0.536 0.592125
## cityname_everett           1.804e-02  6.448e-02   0.280 0.779658
## cityname_fargo            -9.143e-02  6.171e-02  -1.482 0.138481
## cityname_fayetteville     -1.780e-01  6.174e-02  -2.883 0.003955 **
## cityname_federal_way       5.769e-02  5.812e-02   0.993 0.320933
## cityname_forest_lake       2.826e-02  5.688e-02   0.497 0.619346
## cityname_fort_collins      4.077e-03  5.365e-02   0.076 0.939426
## cityname_fort_worth       -3.088e-02  5.583e-02  -0.553 0.580245
## cityname_gaithersburg      4.334e-02  5.502e-02   0.788 0.430855
## cityname_georgetown        3.902e-02  5.426e-02   0.719 0.472109
## cityname_glen_burnie      -1.243e-02  5.072e-02  -0.245 0.806409
## cityname_glendale         -5.787e-02  5.810e-02  -0.996 0.319221
## cityname_grand_forks      -1.671e-01  5.310e-02  -3.148 0.001654 **
## cityname_grapevine         3.398e-02  5.687e-02   0.598 0.550187
## cityname_greensboro       -9.448e-02  5.031e-02  -1.878 0.060448 .
## cityname_greenville       -7.860e-02  5.974e-02  -1.316 0.188337
## cityname_hackensack        1.198e-02  5.818e-02   0.206 0.836810
## cityname_hanover           3.774e-02  5.270e-02   0.716 0.473951
## cityname_hartford         -2.973e-03  5.967e-02  -0.050 0.960257
## cityname_houston          -1.113e-02  4.656e-02  -0.239 0.811041
## cityname_humble           -1.167e-01  6.450e-02  -1.810 0.070358 .
## cityname_hyattsville       6.196e-02  5.499e-02   1.127 0.259883
## cityname_indianapolis     -9.689e-02  5.123e-02  -1.891 0.058639 .
## cityname_issaquah          2.537e-02  5.506e-02   0.461 0.644924
## cityname_jacksonville     -7.804e-02  5.684e-02  -1.373 0.169830
## cityname_jersey_city      -2.125e-01  5.086e-02  -4.178 2.99e-05 ***
## cityname_kansas_city      -2.847e-02  4.731e-02  -0.602 0.547357
## cityname_kent              5.098e-02  4.925e-02   1.035 0.300727
## cityname_kirkland          1.456e-02  5.160e-02   0.282 0.777871
## cityname_lafayette        -4.836e-02  5.262e-02  -0.919 0.358139
## cityname_lakewood         -1.218e-02  5.499e-02  -0.221 0.824761
## cityname_las_vegas        -2.549e-02  4.773e-02  -0.534 0.593288
## cityname_laurel            4.529e-02  5.265e-02   0.860 0.389663
## cityname_lawrence         -3.753e-02  5.363e-02  -0.700 0.484040
## cityname_lawrenceville    -2.667e-02  5.223e-02  -0.511 0.609634
## cityname_lexington        -1.118e-01  5.499e-02  -2.033 0.042133 *
## cityname_lincoln          -1.271e-01  5.968e-02  -2.129 0.033308 *
## cityname_lithonia         -1.010e-01  5.499e-02  -1.837 0.066216 .
```

```
## cityname_long_beach       -9.457e-03  5.042e-02  -0.188 0.851233
## cityname_longview         -9.498e-02  5.811e-02  -1.634 0.102228
## cityname_los_angeles      -1.426e-01  4.722e-02  -3.019 0.002548 **
## cityname_lubbock          -9.087e-02  5.365e-02  -1.694 0.090375 .
## cityname_lynnwood          6.015e-02  5.051e-02   1.191 0.233790
## cityname_madison           4.767e-02  4.704e-02   1.013 0.310887
## cityname_manchester        5.780e-02  5.010e-02   1.154 0.248683
## cityname_mandan           -9.525e-02  6.172e-02  -1.543 0.122838
## cityname_mansfield         2.092e-02  5.427e-02   0.385 0.699911
## cityname_maple_valley      7.563e-02  5.320e-02   1.422 0.155143
## cityname_marietta         -4.264e-02  4.817e-02  -0.885 0.376119
## cityname_miami             3.695e-02  1.116e-01   0.331 0.740703
## cityname_midland           3.504e-02  6.173e-02   0.568 0.570325
## cityname_milford           4.061e-02  5.811e-02   0.699 0.484654
## cityname_milwaukee         9.839e-03  4.815e-02   0.204 0.838103
## cityname_minneapolis       2.620e-02  4.868e-02   0.538 0.590471
## cityname_minot            -1.272e-01  4.935e-02  -2.578 0.009975 **
## cityname_monroe           -9.633e-02  5.817e-02  -1.656 0.097800 .
## cityname_monroeville      -4.555e-02  5.424e-02  -0.840 0.401029
## cityname_morrisville      -5.398e-02  5.500e-02  -0.981 0.326402
## cityname_nashville        -6.875e-02  5.131e-02  -1.340 0.180278
## cityname_new_braunfels    -6.578e-03  5.309e-02  -0.124 0.901404
## cityname_new_london        2.537e-02  5.221e-02   0.486 0.626987
## cityname_newport_news     -1.093e-01  7.442e-02  -1.468 0.142124
## cityname_norfolk           2.622e-02  5.498e-02   0.477 0.633408
## cityname_norman           -2.521e-02  4.939e-02  -0.510 0.609748
## cityname_oakland          -1.972e-01  5.712e-02  -3.452 0.000559 ***
## cityname_odenton           4.537e-02  5.974e-02   0.759 0.447587
## cityname_oklahoma_city    -3.549e-02  4.863e-02  -0.730 0.465605
## cityname_omaha            -4.432e-02  4.798e-02  -0.924 0.355763
## cityname_orange            3.347e-02  5.817e-02   0.575 0.565037
## cityname_orlando           3.243e-02  5.222e-02   0.621 0.534545
## citynameother             -4.365e-02  4.564e-02  -0.956 0.338882
## cityname_overland_park    -6.504e-02  6.171e-02  -1.054 0.291934
## cityname_owings_mills      1.610e-02  6.172e-02   0.261 0.794169
## cityname_pasadena         -8.985e-02  5.011e-02  -1.793 0.073004 .
## cityname_pflugerville      2.516e-02  5.498e-02   0.458 0.647177
## cityname_philadelphia      4.032e-02  5.013e-02   0.804 0.421218
## cityname_phoenix           2.007e-02  4.891e-02   0.410 0.681533
## cityname_pittsburgh        5.594e-03  4.948e-02   0.113 0.909988
## cityname_plainsboro        6.708e-02  5.312e-02   1.263 0.206698
## cityname_plymouth          2.060e-02  5.685e-02   0.362 0.717098
## cityname_portland          8.299e-03  4.747e-02   0.175 0.861234
## cityname_princeton         8.892e-03  5.376e-02   0.165 0.868626
## cityname_raleigh          -2.167e-02  4.776e-02  -0.454 0.650016
## cityname_randallstown     -2.938e-03  6.171e-02  -0.048 0.962032
## cityname_redmond           3.162e-02  5.974e-02   0.529 0.596671
## cityname_reno              4.479e-02  5.497e-02   0.815 0.415195
## cityname_renton            4.288e-02  4.927e-02   0.870 0.384201
## cityname_richmond         -1.592e-02  5.499e-02  -0.289 0.772226
## cityname_rochester        -5.171e-02  5.496e-02  -0.941 0.346760
## cityname_rockville         3.434e-02  5.587e-02   0.615 0.538825
## cityname_roswell          -2.715e-02  5.364e-02  -0.506 0.612804
## cityname_round_rock        2.457e-02  5.122e-02   0.480 0.631554
```

```
## cityname_saint_george             NA         NA      NA       NA
## cityname_saint_louis      -2.948e-02  4.900e-02  -0.602 0.547470
## cityname_saint_louis_park  3.731e-02  5.584e-02   0.668 0.504074
## cityname_saint_paul        4.390e-02  5.224e-02   0.840 0.400742
## cityname_saint_petersburg  3.203e-02  5.967e-02   0.537 0.591482
## cityname_salt_lake_city    3.190e-02  5.049e-02   0.632 0.527572
## cityname_san_antonio       1.169e-03  4.658e-02   0.025 0.979972
## cityname_san_diego         1.334e-02  4.903e-02   0.272 0.785632
## cityname_san_francisco    -1.042e+00  5.211e-02 -20.004  < 2e-16 ***
## cityname_san_jose         -1.774e-02  5.824e-02  -0.305 0.760713
## cityname_san_marcos        2.012e-02  6.446e-02   0.312 0.754976
## cityname_santa_monica     -2.992e-01  8.552e-02  -3.499 0.000471 ***
## cityname_sarasota          1.907e-02  5.687e-02   0.335 0.737396
## cityname_seattle           1.611e-02  4.826e-02   0.334 0.738547
## cityname_silver_spring     4.911e-02  5.125e-02   0.958 0.337977
## cityname_sioux_falls      -1.190e-01  4.872e-02  -2.443 0.014589 *
## cityname_smyrna           -2.045e-03  5.687e-02  -0.036 0.971315
## cityname_spring           -1.286e-01  5.815e-02  -2.212 0.026999 *
## cityname_springfield      -7.505e-02  5.122e-02  -1.465 0.142859
## cityname_suwanee           4.929e-02  5.813e-02   0.848 0.396441
## cityname_tallahassee      -1.767e-02  5.496e-02  -0.321 0.747896
## cityname_tampa             1.725e-02  5.073e-02   0.340 0.733829
## cityname_toledo           -1.232e-01  5.582e-02  -2.206 0.027407 *
## cityname_tucson           -3.288e-02  5.498e-02  -0.598 0.549838
## cityname_tulsa            -1.425e-01  5.363e-02  -2.658 0.007893 **
## cityname_tuscaloosa       -5.971e-02  5.592e-02  -1.068 0.285666
## cityname_tyler            -1.119e-01  6.171e-02  -1.813 0.069855 .
## cityname_urbana            2.053e-02  5.310e-02   0.387 0.699045
## cityname_vancouver         3.809e-02  5.123e-02   0.743 0.457216
## cityname_virginia_beach    2.043e-02  5.582e-02   0.366 0.714363
## cityname_washington       -1.013e-01  4.912e-02  -2.063 0.039139 *
## cityname_west_des_moines  -9.284e-02  6.170e-02  -1.505 0.132468
## cityname_west_hollywood   -1.482e-01  6.200e-02  -2.390 0.016887 *
## cityname_west_lafayette    1.284e-02  4.837e-02   0.265 0.790668
## cityname_west_new_york    -9.254e-02  5.833e-02  -1.586 0.112682
## cityname_winston_salem    -7.468e-02  5.809e-02  -1.285 0.198672
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1019 on 5873 degrees of freedom
## Multiple R-squared:  0.9411, Adjusted R-squared:  0.9393
## F-statistic: 518.4 on 181 and 5873 DF,  p-value: < 2.2e-16
```

Kết quả từ mô hình hồi quy tuyến tính bậc 2 trên tập dữ liệu train_data_clean trông rất khả quan:

- Độ chính xác của mô hình: Với hệ số xác định điều chỉnh (Adjusted R-squared) lên đến 0.9393, có nghĩa là mô hình có thể giải thích được khoảng 93.93% biến thiên của biến phụ thuộc (price) dựa trên các biến dự báo. Đây là một kết quả rất tốt.
- Mức ý nghĩa thống kê của các biến dự báo: Các biến dự báo (bao gồm cả bình phương của square_feet) đều có mức ý nghĩa thống kê rất cao, với hầu hết p-values < 0.001 (được đánh dấu ***). Điều này cho thấy các biến này đóng góp đáng kể vào việc dự báo giá.
- Hiệu suất dự báo của mô hình: Với sai số chuẩn hóa của phần dư (Residual standard error) chỉ 0.1019,

có nghĩa là mô hình có độ lệch chuẩn của phần dư rất thấp, chứng tỏ khả năng dự báo của mô hình rất tốt.

Nhìn chung, việc thêm bậc 2 của biến square_feet đã giúp cải thiện đáng kể chất lượng của mô hình hồi quy tuyến tính. Mô hình hồi quy tuyến tính bậc 2 với biến **square_feet** là mô hình phù hợp nhất để dự báo giá thuê căn hộ:

- Giúp hiểu rõ hơn mối quan hệ giữa diện tích căn hộ và giá thuê. Mô hình cho thấy giá thuê tăng không tuyến tính theo diện tích, với tốc độ tăng nhanh hơn ở các căn hộ có diện tích lớn hơn.
- Từ đó, có thể đề xuất các kích cỡ căn hộ phù hợp với nhu cầu sử dụng và khả năng chi trả của người thuê. Ví dụ, với người thuê cá nhân, căn hộ 50-70 m2 có thể là lựa chọn hợp lý. Còn với gia đình đông người, căn hộ 80-100 m2 sẽ phù hợp hơn.
- Giúp các công ty môi giới có cơ sở khoa học hơn trong việc định giá và tư vấn cho khách hàng, từ đó nâng cao chất lượng dịch vụ.

7. Kết luận

Qua quá trình phân tích, chúng ta đã rút ra được một số kết quả quan trọng về thị trường thuê nhà:

- Vị trí địa lý, diện tích, số lượng phòng và tiện ích là những yếu tố chính ảnh hưởng đến mức giá thuê nhà. Các căn hộ tại các thành phố lớn, có diện tích rộng, nhiều phòng và đi kèm nhiều tiện ích sẽ có giá thuê cao hơn.
- Người thuê nhà cần cân nhắc kỹ các yếu tố ưu tiên như vị trí, diện tích, tiện ích để lựa chọn được căn hộ phù hợp với nhu cầu và khả năng tài chính của mình. Việc nghiên cứu kỹ thị trường cũng rất quan trọng để tìm được thời điểm và căn hộ phù hợp.
- Các công ty môi giới cần tư vấn khách hàng dựa trên phân tích dữ liệu chi tiết về thị trường và các yếu tố ảnh hưởng đến giá thuê, giúp khách hàng đưa ra quyết định tốt nhất. Việc theo dõi sát sao diễn biến thị trường cũng là điều cần thiết.