Thiết lập thư viện dùng cho cả bài:

library(stats)

library(dplyr)

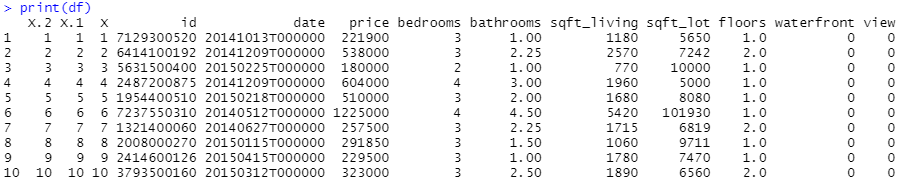
library(magrittr)

1. Đọc dữ liệu

Code:

df = read.csv('gia\_nha.csv');

Output: In datafrace df được kết quả như sau:



Lưu ý: Tổng cộng dataframe có 21572 dòng. Dataframe df còn các cột condition, grade, sqft\_above không đủ vị trí để hiển thị.

1. Làm sạch dữ liệu

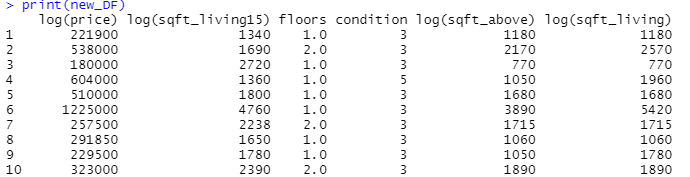
Code:

new\_DF = df %>% select('price', 'sqft\_living15', 'floors','condition','sqft\_above','sqft\_living');

new\_DF = new\_DF %>% filter(!(is.na(price) | is.na(sqft\_living15) | is.na(sqft\_living) | is.na(floors) | is.na(condition) | is.na(sqft\_above)));

new\_DF = new\_DF %>% rename('log(price)'=price, 'log(sqft\_living15)'=sqft\_living15, 'log(sqft\_living)'=sqft\_living, 'log(sqft\_above)'=sqft\_above);

Output: In dataframe new\_DF nhận được kết quả như sau



Lưu ý: Tổng cộng dataframe có 21427 dòng.

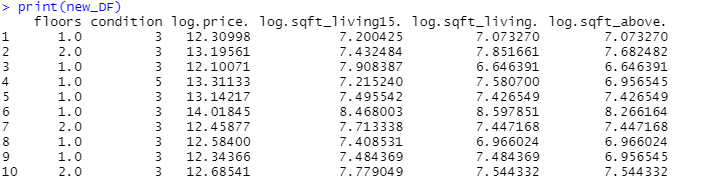
1. Làm rõ dữ liệu
2. Chuyển đổi biến

Code:

Log = function(x) {log(x)};

new\_DF = data.frame(new\_DF %>% select('floors','condition'), apply(new\_DF %>% select('log(price)', 'log(sqft\_living15)','log(sqft\_living)','log(sqft\_above)'),2,Log));

Output: In dataframe new\_DF nhận được kết quả như sau



1. Tính các giá trị thống kê

Code:

table = data.frame(Name = c('mean','median','sd','min','max'),

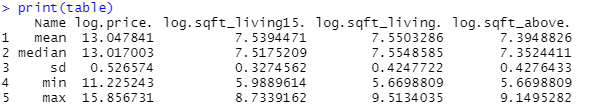
log.price.= c(mean(new\_DF[,'log.price.']), median(new\_DF[,'log.price.']),sd(new\_DF[,'log.price.']),min(new\_DF[,'log.price.']),max(new\_DF[,'log.price.'])),

log.sqft\_living15.= c(mean(new\_DF[,'log.sqft\_living15.']), median(new\_DF[,'log.sqft\_living15.']),sd(new\_DF[,'log.sqft\_living15.']),min(new\_DF[,'log.sqft\_living15.']),max(new\_DF[,'log.sqft\_living15.'])),

log.sqft\_living.= c(mean(new\_DF[,'log.sqft\_living.']), median(new\_DF[,'log.sqft\_living.']),sd(new\_DF[,'log.sqft\_living.']),min(new\_DF[,'log.sqft\_living.']),max(new\_DF[,'log.sqft\_living.'])),

log.sqft\_above.= c(mean(new\_DF[,'log.sqft\_above.']), median(new\_DF[,'log.sqft\_above.']),sd(new\_DF[,'log.sqft\_above.']),min(new\_DF[,'log.sqft\_above.']),max(new\_DF[,'log.sqft\_above.']))Output: In dataframe new\_DF nhận được kết quả như sau

Output:



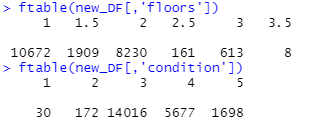
1. Lập bảng tần số:

Code:

ftable(new\_DF[,'floors'])

ftable(new\_DF[,'condition'])

Output:



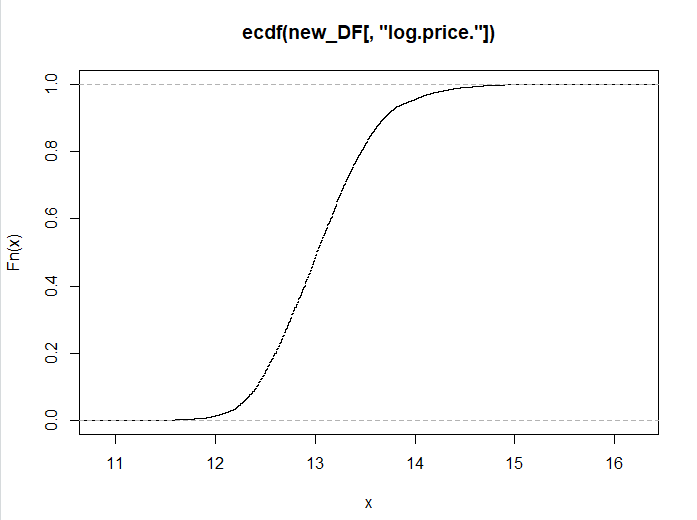
1. Vẽ đồ thị phân phối biến price

Code:

P = ecdf(new\_DF[,'log.price.']);

plot(P)

Output:



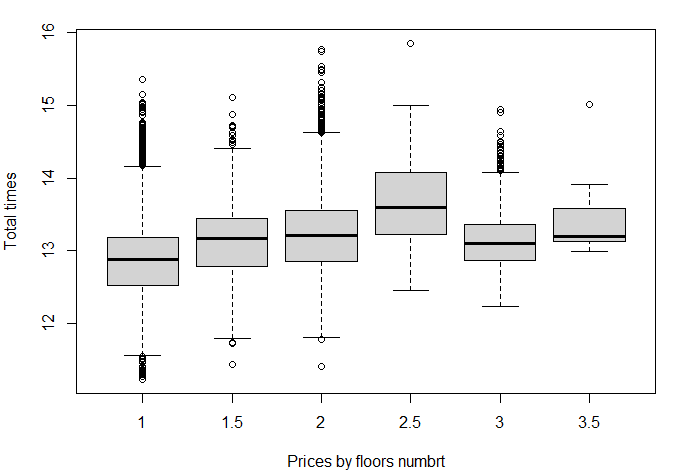
1. Vẽ phân phối của price cho floors và condition

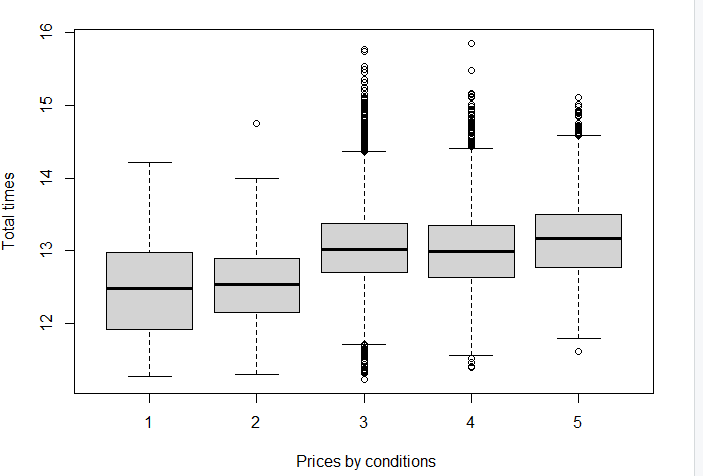
Code:

boxplot(new\_DF$log.price. ~ new\_DF$floors, ylab = "Total times", xlab = "Prices by floors numbrt")

boxplot(new\_DF$log.price. ~ new\_DF$condition, ylab = "Total times", xlab = " Prices by conditions")

Output:





1. Vẽ phân phối của price cho sqft\_living15, sqft\_above, và sqft\_living

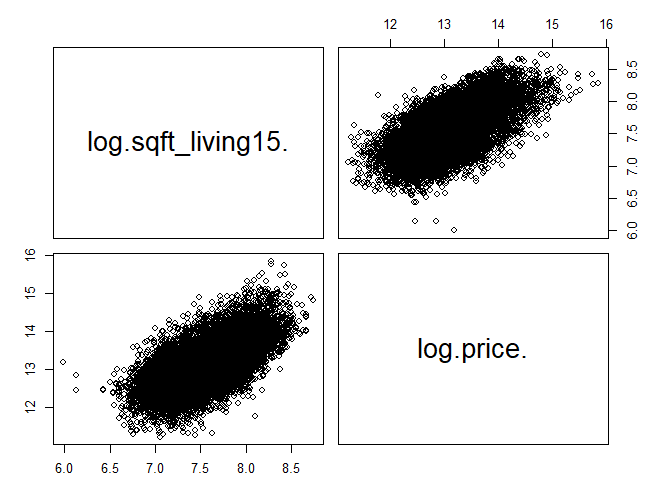
Code:

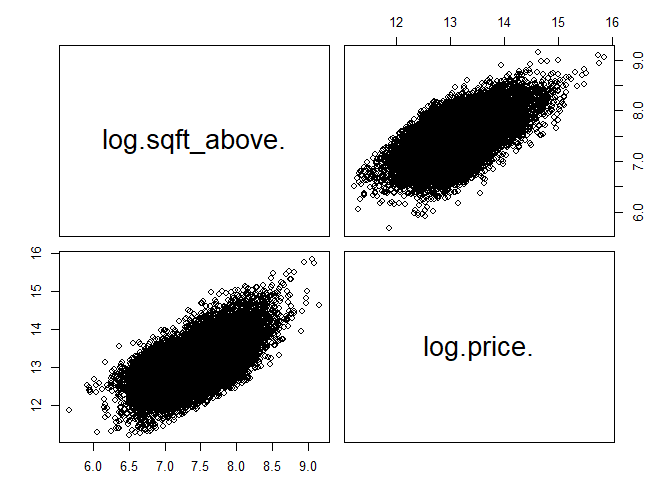
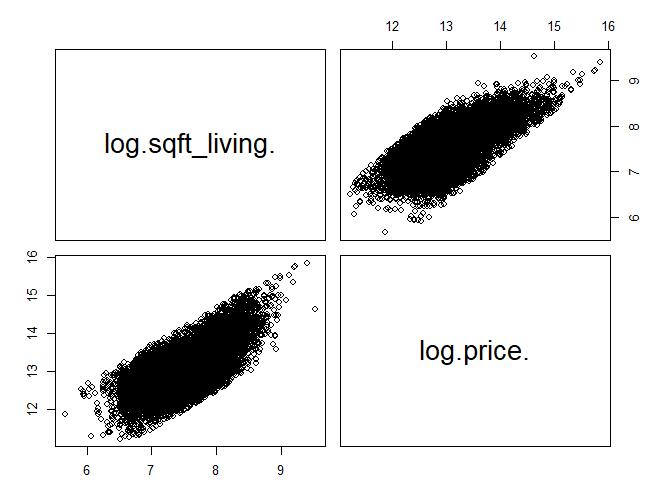
pairs(new\_DF[,c('log.sqft\_living15.','log.price.')])

pairs(new\_DF[,c('log.sqft\_living.','log.price.')])

pairs(new\_DF[,c('log.sqft\_above.','log.price.')])

Output: (tượng trung cho biến log.sqft\_living15)





4)

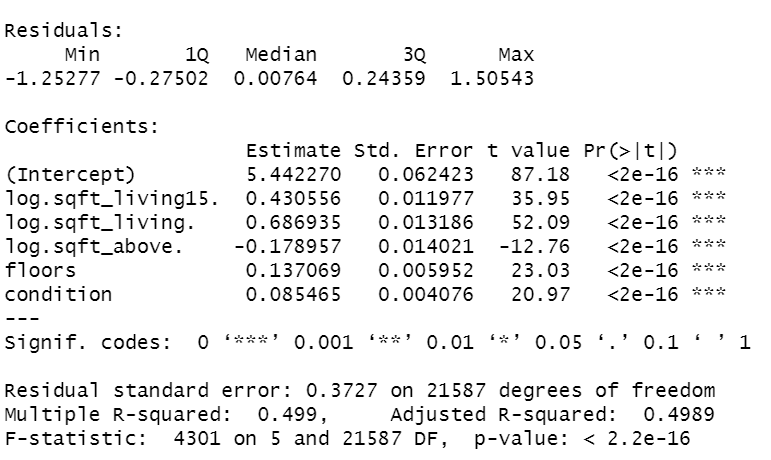
a) Xét mô hình hồi quy tuyến tính bao gồm biến price là một biến phụ thuộc, và tất cả các biến còn lại đều là biến độc lập. Hãy dùng lệnh lm() để thực thi mô hình hồi quy tuyến tính bội.

Phần code:

lmPrice = lm(log.price.~log.sqft\_living15.+log.sqft\_living.+log.sqft\_above.+floors+condition,new\_DF)

summary(lmPrice)

Thu được kết quả lmPrice chứa thông tin, các tham số cho chương trình hồi quy, gọi hàm summary():



Các thông tin:

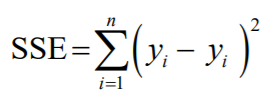
-Residual: thống kê thông tin về độ lệch so sánh giữa hàm giả thiết và dữ liệu cho trước

-Coffecients:

+ Estimate: các hệ số của phương trình hồi quy

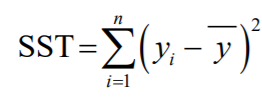
Price = 5.4423 + 0.4306 \* sqft\_living15 + 0.6869 \* sqft\_living – 0.1789 \* sqft\_above + 0.1371 \* floors + 0.085465 \* condition

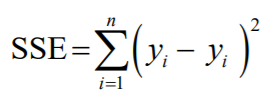
+ Residual Standard Error: Sai số chuẩn ước lượng

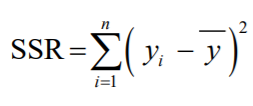


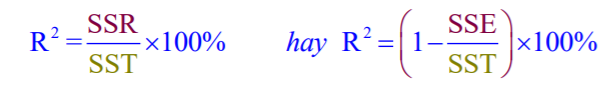
n = 21587; k = 5

+Multiple R-squared: Hệ số tương quan tuyến tính thể hiện mức độ phù hợp của mô hình









Multiple R-sq = 0.499 Adjusted Multiple R-sq = 0.4989

+F-Statistic: Tiêu chuẩn kiểm định

t-stat = 4301

Giả thiết:

H0: R2= 0

H1: R2 0

Miền bác bỏ :

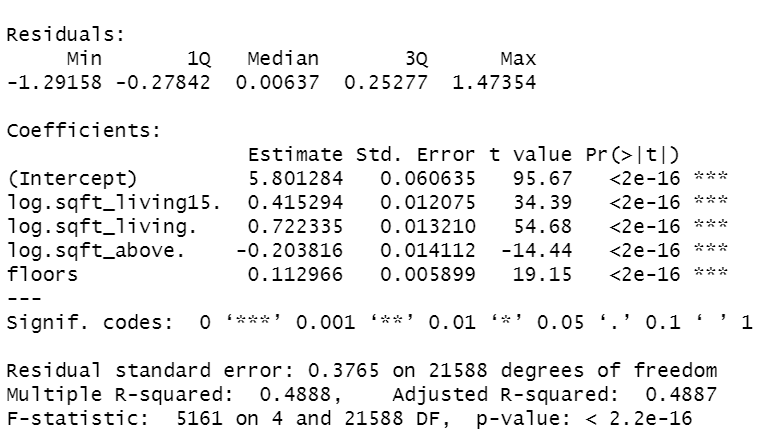
Tiêu chuẩn thuộc miền bác bỏ, bác bỏ H0 thừa nhận H1, biến price có mối quan hệ tuyến tính với tập giá trị input

b) Dựa vào kết quả P-value (Pr > |t|) của 5 feature đều rất nhỏ, vậy rõ ràng khả năng bác bỏ H0 của 5 biến trên đều rất cao (2.2 \* << 0.05), vậy không bác bỏ bất kỳ feature nào ở mức tin cậy 0.05

c)

Mô hình M1 đã được thực hiện ở câu a)

Mô hình M2: loại bỏ ‘condition’



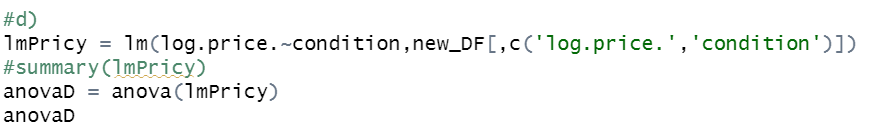
Giả thiết:

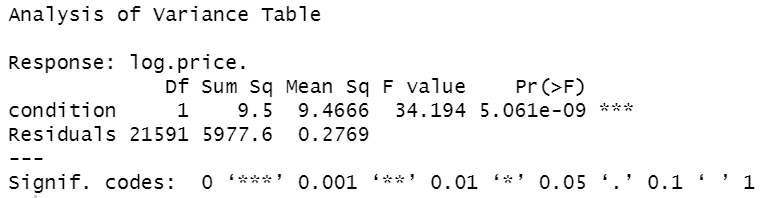
H0: R2= 0

H1: R2 0

Sau khi loại bỏ ‘condition’, thông số R-sq giảm xuống, tuy vẫn bác bỏ mạnh H0 thừa nhận tương quan tuyến tính của mô hình, mức bác bỏ vẫn rất cao (P-value = 2.2 \* << 0.05) nhưng không tốt bằng mô hình ban đầu

Sử dụng lệnh anova() khi xét mô hình price theo condition





Giả thiết:

H0: Price không phụ thuộc vài sự biến thiên của condition

H1: Price phụ thuộc vài sự biến thiên của condition

P-value = 5.061 \* 10-9 << . Vậy bác bỏ H0 tốt, thừa nhận sự phụ thuộc vào sự biến thiên của condition, vậy rõ rang mô hình M1 là mô hình chuẩn xác hơn.

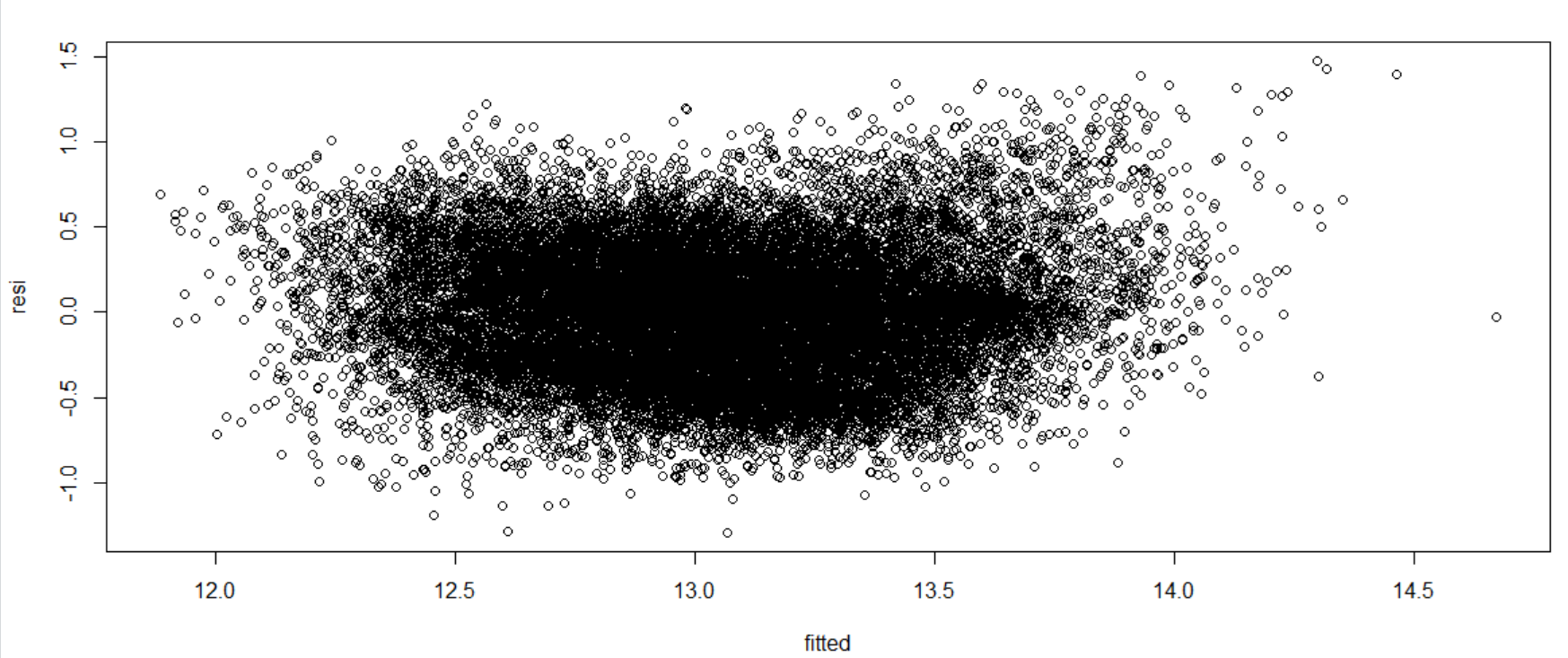
d) Mô hình hợp lý hơn ở câu c là mô hình M1

Các biến của M1 tác động vào giá trị price theo quan hệ tuyến tính xác định bằng hệ số estimated coefficient

3 giá trị sqft\_living, sqft\_living15, floors tác động đáng kể vào giá nhà theo tỷ lệ thuận, conditions cũng tác động theo xu hướng tang dần nhưng mức tác động thấp hơn.

Giá trị sqft\_above có tác động theo chiều tỉ lệ nghịch với giá nhà do hệ số âm

e) Đồ thị biểu diễn sai số hồi quy so với giá trị dự báo:



Giá nhà dao động phần lớn xung quanh giá trị 1012 – 1014, log (sai số dự báo dao động) từ -1 đến 1 tức dự báo chênh lệch lên đến 10 lần, sự chênh lệch do dự báo độc lập với giá trị dự báo được bởi mô hình

5)

Để tiến hành dự đoán giá nhà tại 2 thuộc tính x1 và x2, trước tiên ta cần phải tạo 2 data frame x1 chứa các biến sqft\_living15 = mean(sqft\_living15), sqft\_living = mean(sqft\_living), sqft\_above = mean(sqft\_above), floors = 2, condition = 3 và x2 chứa các biến sqft\_living15 = max(sqft\_living15), sqft\_living = max(sqft\_living), sqft\_above = max(sqft\_above), floors = 2, condition = 3. Sau đó sử dụng lệnh predict với mô hình hồi quy tuyến tính thu được ở câu 4c và độ tin cậy 95%

Phần code:

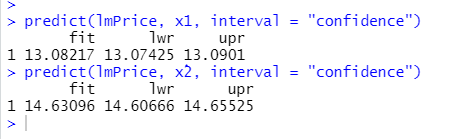
x1 = data.frame(log.sqft\_living15. = mean(new\_DF[,'log.sqft\_living15.']), log.sqft\_living. = mean(new\_DF[,'log.sqft\_living.']), log.sqft\_above. = mean(new\_DF[,'log.sqft\_above.']), floors = 2, condition = 3)

x2 = data.frame(log.sqft\_living15. = max(new\_DF[,'log.sqft\_living15.']), log.sqft\_living. = max(new\_DF[,'log.sqft\_living.']), log.sqft\_above. = max(new\_DF[,'log.sqft\_above.']), floors = 2, condition = 3)

predict(lmPrice, x1, interval = "confidence")

predict(lmPrice, x2, interval = "confidence")

Và ta thu được kết quả như sau:



Khoảng tin cậy cho giá trị price dự đoán được ở x2 lớn hơn x1.