Linear Regression II

your name

2024-10-31

Math 2265 Chapter 8. Linear Regression

- Work as a group!
- You will need to replace `"ans"` or `your_answer` in the source code
- Update your name in L3
- Add your group members' name below; students may lose one point if Question 0 is unanswered
- Make sure you save and `knit` your work (to html or pdf) before submitting it to Canvas
- Please only submit your work if you attended the class and worked with other students; this is not an online course

---

Goal

1. Inference using a linear model
2. Apply linear regression to a real work data set

Question 0. Who are your group members? (List their first names)

Answer:

1. `<name_1>`
2. `<name_2>`

If you need more time to get used to `Markdown`, use the `Visual` mode.

The icon is located in the upper-left corner next to `source`.

`duke_forest`: Sale prices of houses in Duke Forest, Durham, NC

Data Set: `duke_forest`.

We will work with the `duke_forest` data set. Write a script to check the number of observations and the names of the variables of the data set.

```
# write your code
str(duke_forest)
```

```
## tibble [98 x 13] (S3: tbl_df/tbl/data.frame)
##  $ address   : chr [1:98] "1 Learned Pl, Durham, NC 27705" "1616 Pinecrest Rd, Durham, NC 27705" "24
##  $ price     : num [1:98] 1520000 1030000 420000 680000 428500 ...
##  $ bed       : num [1:98] 3 5 2 4 4 3 5 4 4 3 ...
##  $ bath      : num [1:98] 4 4 3 3 3 3 5 3 5 2 ...
##  $ area      : num [1:98] 6040 4475 1745 2091 1772 ...
##  $ type      : chr [1:98] "Single Family" "Single Family" "Single Family" "Single Family" ...
##  $ year_built: num [1:98] 1972 1969 1959 1961 2020 ...
##  $ heating   : chr [1:98] "Other, Gas" "Forced air, Gas" "Forced air, Gas" "Heat pump, Other, Electri
##  $ cooling   : Factor w/ 2 levels "other","central": 2 2 2 2 2 2 2 2 2 1 ...
##  $ parking   : chr [1:98] "0 spaces" "Carport, Covered" "Garage - Attached, Covered" "Carport, Covere
##  $ lot       : num [1:98] 0.97 1.38 0.51 0.84 0.16 0.45 0.94 0.79 0.53 0.73 ...
##  $ hoa       : chr [1:98] NA NA NA NA ...
##  $ url       : chr [1:98] "https://www.zillow.com/homedetails/1-Learned-Pl-Durham-NC-27705/49981897_z
```

Recall we use linear regression for a pair of numerical variables, explanatory (horizontal, often denoted by x) and response variable (vertical, often denoted by y).

Task

Save all the numerical variables as an R vector to the R variable `numerical_variables` below

- Just copy the names from the previous output and paste them as parameters of the vector
- Strings (e.g., variable names) need to be enclosed with quotation marks “variable_name”

```r
# numerical_variable <- c("variable_name1","variable_name2",...,...)
numerical_variable <- c("price","bed","bath","area","year_built","lot")
numerical_variable
```

```
## [1] "price"      "bed"        "bath"       "area"       "year_built"
## [6] "lot"
```

Among the variables we will use `price` as the response variable. The goal is to find the **most suitable explanatory variable** for linear regression. The word *most* requires justification, and we will use two crietria:

1. The correlation coefficient R.
2. The sum of squared residuals with respect to the least squares line.

Best with respect to the correlation coefficient

To do this, we will need to compute the correlation coefficients of `price` over all the rest of the numerical variables you found above. Use the following block to compute the correlation coefficients.

```r
###
#cor_1 <-
#cor_2 <-
#cor_3 <-
#cor_4 <-
#cor_5 <-
#cor_6 <-
#cor_7 <-
#cor_8 <-
# use as many as you need and delete unused ones

cor_1 <- cor(x=duke_forest$price, y=duke_forest$bed)
cor_2 <- cor(x=duke_forest$price, y=duke_forest$bath)
cor_3 <- cor(x=duke_forest$price, y=duke_forest$area)
cor_4 <- cor(x=duke_forest$price, y=duke_forest$year_built)
cor_5 <- cor(x=duke_forest$price, y=duke_forest$lot)

# cor_vec <- c(cor_1, cor_2, cor_3, your_answer, your_answer, ... )
```

```
cor_vec <- c(cor_1, cor_2, cor_3, cor_4, cor_5)
cor_vec
```

```
## [1] 0.4105010 0.5886360 0.6672290 0.2018445        NA
```

One of the results may be “NA” since that variable has missing value(s).

Use `sort`, `max`, or `min` to determine the best one. Recall when determining the best, we need to consider their absolute values.

```
sort(abs(cor_vec))
```

```
## [1] 0.2018445 0.4105010 0.5886360 0.6672290
```

Question: Which variables are linearly associated `price`?

Choose top two:

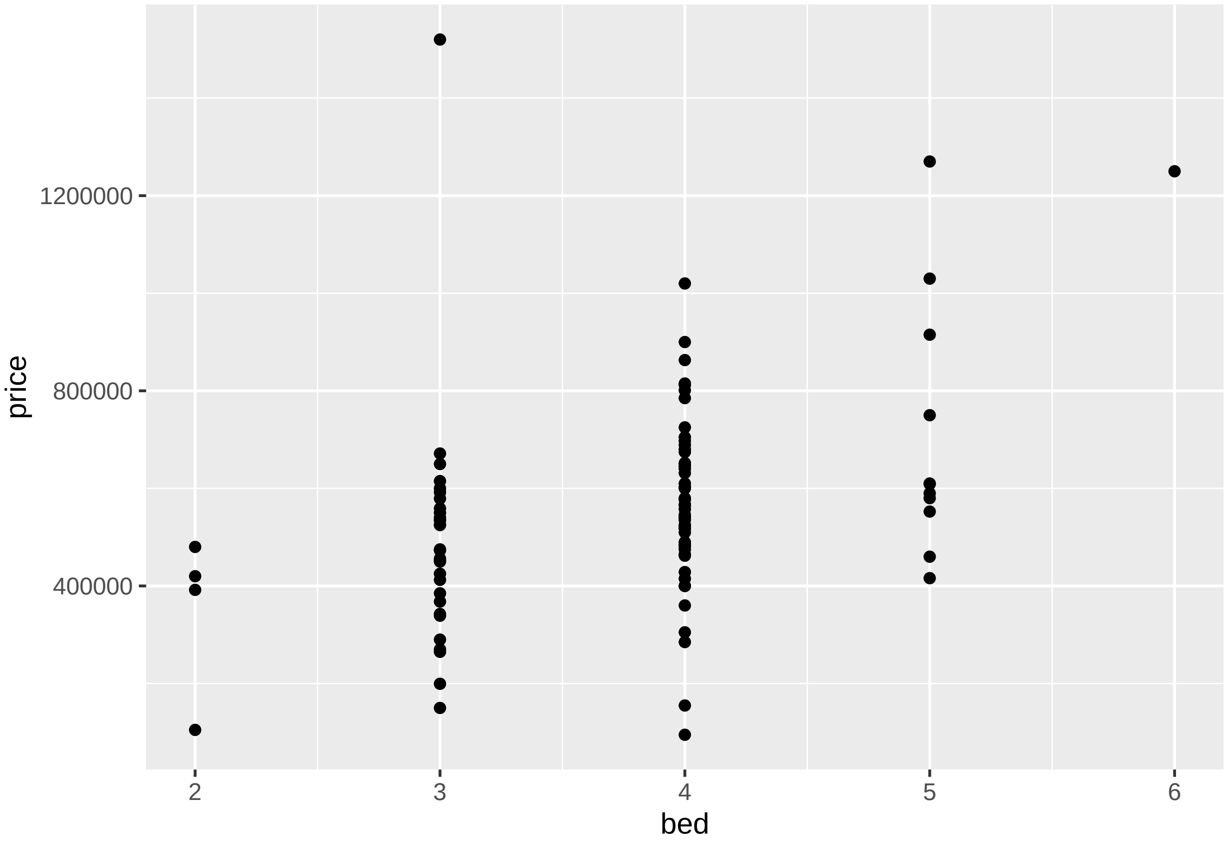
1. your_answer
2. your_answer

Best with respect to the least squares line

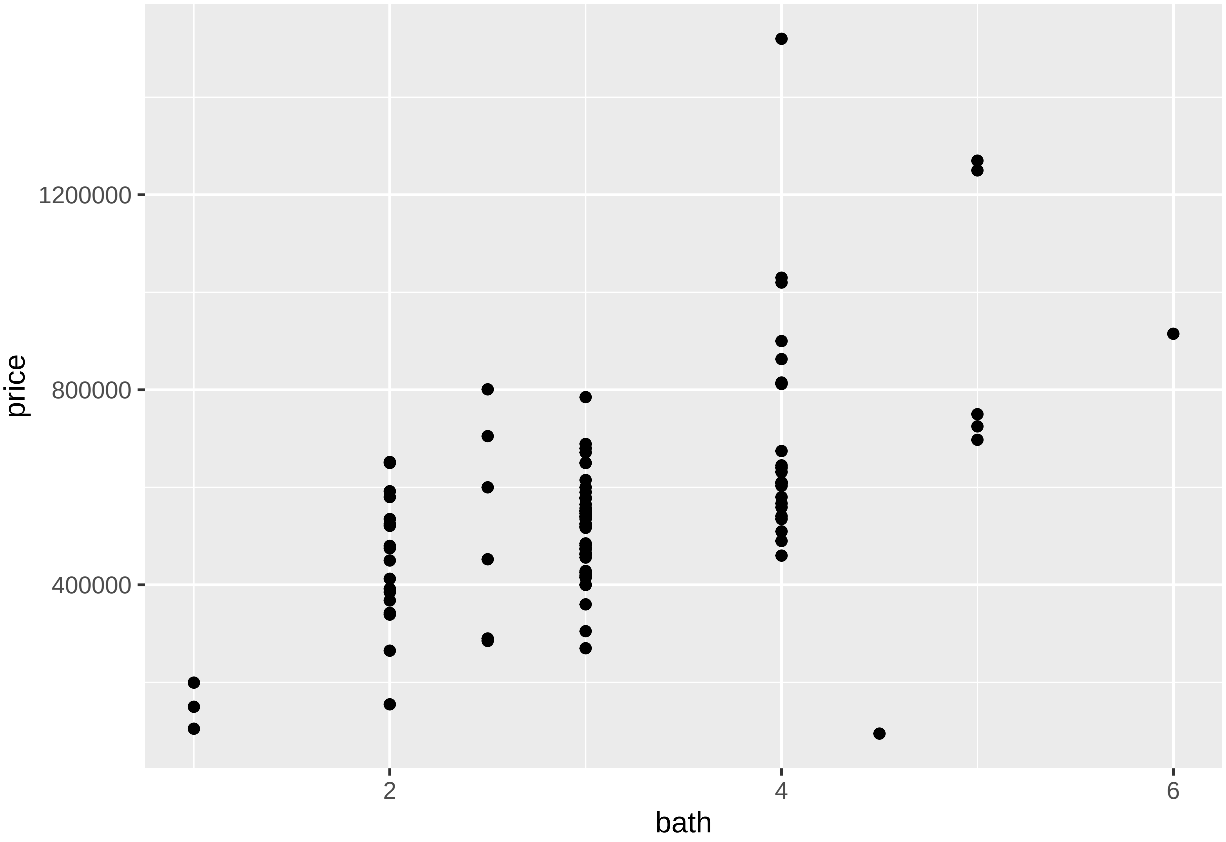
To do this, we will use `lm`. However, to use linear regression, we need to check that the data follows the linear trend. That is, the association is linear. We can check this using their scatter plots.

In the next cell, make scatter plots for each of the explanatory variables.

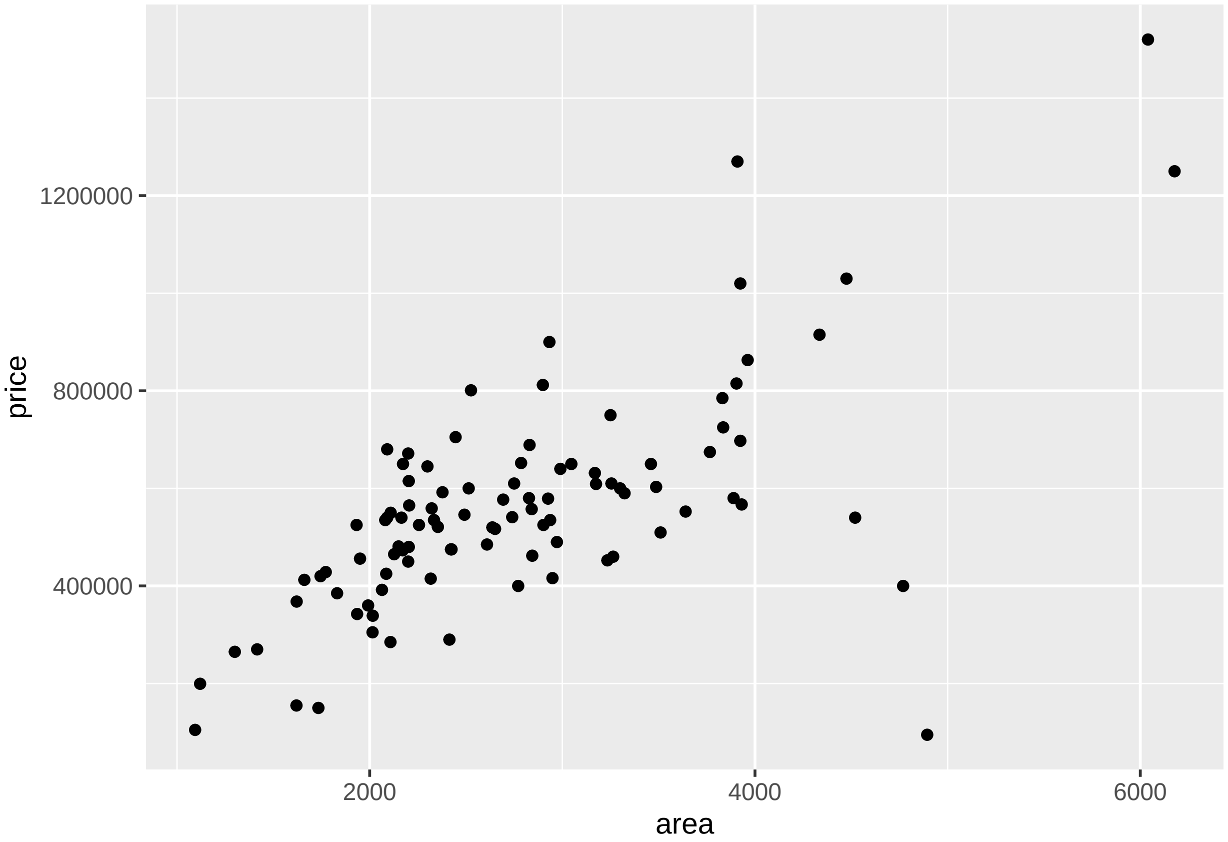
```
# ggplot2(data=duke_forest, aes(x=your_naswer,y=price))+
#   geom_point()
#
# ggplot2(data=duke_forest, aes(x=your_naswer,y=price))+
#   geom_point()
#
# ggplot2(data=duke_forest, aes(x=your_naswer,y=price))+
#   geom_point()
#
# # Use as many as you need and delete the rest
# ggplot2(data=duke_forest, aes(x=your_naswer,y=price))+
#   geom_point()
#
# ggplot2(data=duke_forest, aes(x=your_naswer,y=price))+
#   geom_point()
#
# ggplot2(data=duke_forest, aes(x=your_naswer,y=price))+
#   geom_point()

ggplot(data=duke_forest, aes(x=bed,y=price))+
  geom_point()
```



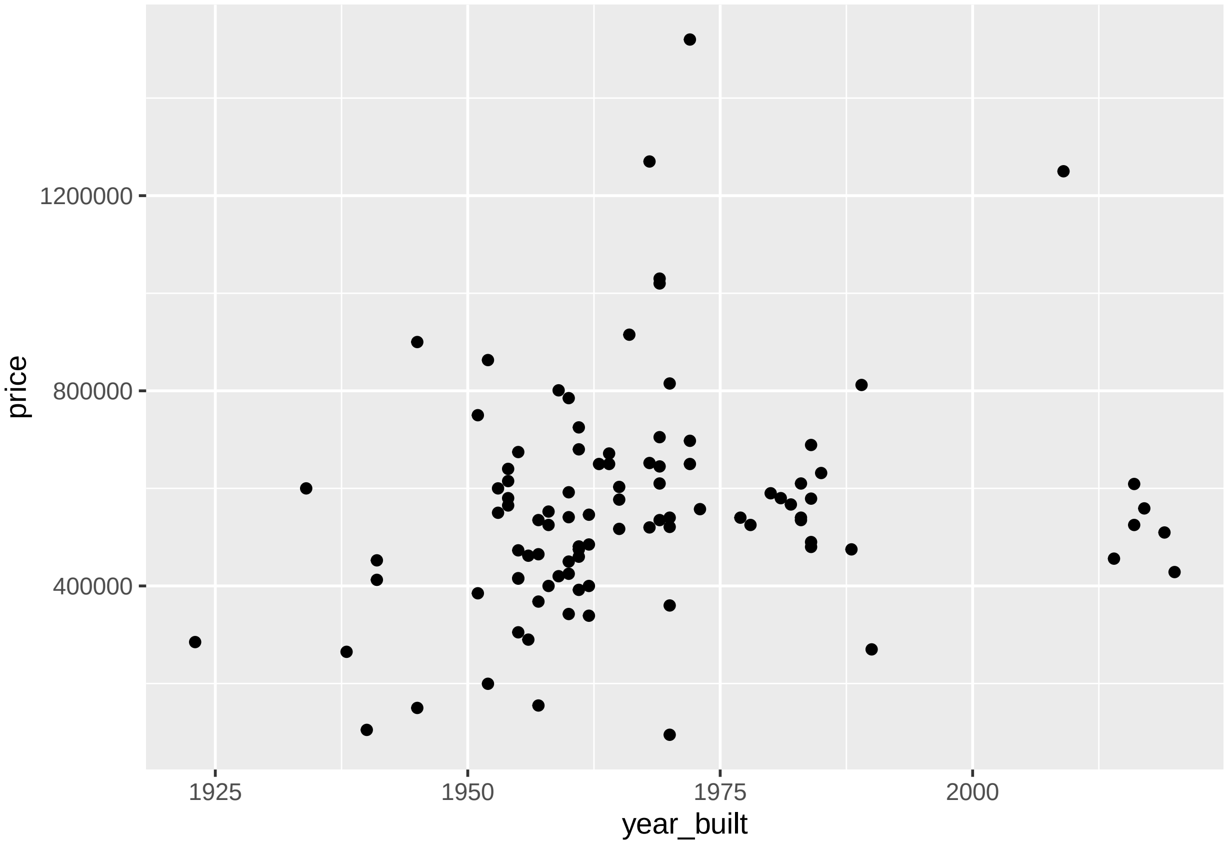
```r
ggplot(data=duke_forest, aes(x=bath,y=price))+
  geom_point()
```



```r
ggplot(data=duke_forest, aes(x=area,y=price))+
  geom_point()
```



```r
ggplot(data=duke_forest, aes(x=year_built,y=price))+
  geom_point()
```



Use the scatter plots to choose two best variables that have linear association with `price`. Now, we will use `lm` to find the least suqres line and the sum of squared residuals for each case.

Save the linear model for your 1st choice

```
# model1 <- lm( price ~ your_answer, data=duke_forest)
model1 <- lm( price ~ area, data=duke_forest)
summary(model1)
```

```
##
## Call:
## lm(formula = price ~ area, data = duke_forest)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -802163  -70824   -3786   85449  529928
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 116652.33   53302.46   2.188   0.0311 *
## area           159.48      18.17   8.777 6.29e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 168800 on 96 degrees of freedom
## Multiple R-squared:  0.4452, Adjusted R-squared:  0.4394
## F-statistic: 77.03 on 1 and 96 DF,  p-value: 6.292e-14
```

Compute the sum of the squared residuals and save it to `ssr1`.

```
# ssr1 <- sum( your_answer)
ssr1 <- sum( residuals(model1)^2 )
ssr1
```

```
## [1] 2.735303e+12
```

Save the linear model for your 2nd choice

```
# model2 <- lm( price ~ your_answer, data=duke_forest)
model2 <- lm( price ~ year_built, data=duke_forest)
summary(model2)
```

```
##
## Call:
## lm(formula = price ~ year_built, data = duke_forest)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -472194 -121367  -29801   87334  947858
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4306458    2410117  -1.787   0.0771 .
## year_built      2474       1225   2.019   0.0463 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 222000 on 96 degrees of freedom
## Multiple R-squared:  0.04074,    Adjusted R-squared:  0.03075
## F-statistic: 4.077 on 1 and 96 DF,  p-value: 0.04625
```

Compute the sum of the squared residuals and save it to `ssr2`.

```
# ssr2 <- sum( your_answer)
ssr2 <- sum( residuals(model2)^2 )
ssr2
```

```
## [1] 4.72934e+12
```

Execute the next line to compare the results.

```
if(ssr1 < ssr2) {
  print("Your 1st choice has lower error. Use this to make a prediction in the next part.")
} else {
  print("Your 2nd choice has lower error. Use this to make a prediction in the next part.")
}
```

```
## [1] "Your 1st choice has lower error. Use this to make a prediction in the next part."
```

Question:

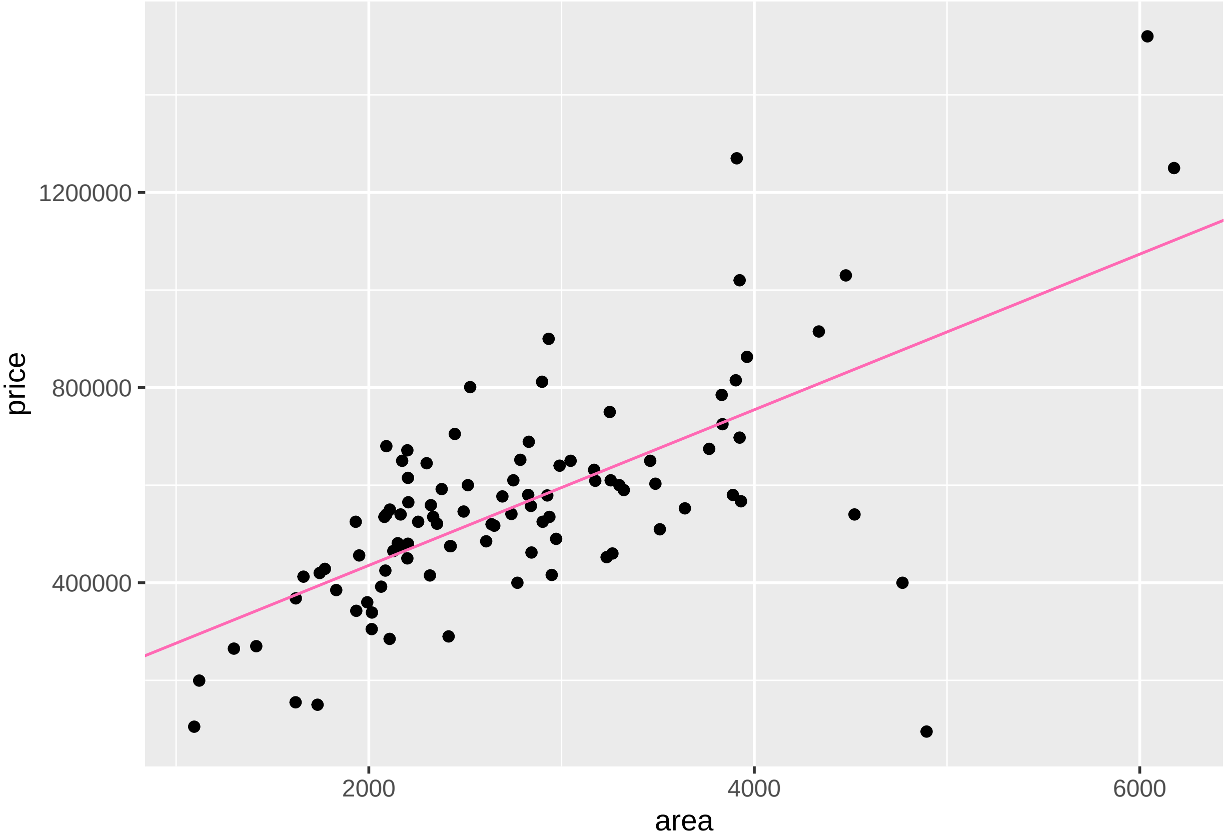
Which variable explains the responds variable `price` better?

Answer:

Make prediction based on your model

First plot the least squares line. Use your answer above as your explanatory variable.

```
# ggplot(data=duke_forest, aes(x = your_answer, y = price)) +
#   geom_point() +
#   geom_abline(intercept = your_answer, slope = your_answer, color='hotpink')

ggplot(data=duke_forest, aes(x = area, y = price)) +
  geom_point() +
  geom_abline(intercept = 116652.33 , slope = 159.48, color='hotpink')
```



Make a prediction where your x value is the median and mean of your variable. That is if your variable is called `my_expanatory_variable`, $x = median(my_expanatory_variable)$ or $x = mean(my_expanatory_variable)$.

```
# x <- median(duke_forest$my_expanatory_variable)
# or use mean (or do both)
# x <- mean(duke_forest$my_expanatory_variable)
x <- median(duke_forest$area)

y_hat <- function(x) {
  # intercept + slope * x
  116652.33 + 159.48 * x
}

# You do not need to modify the following two lines. It will print the outcome.
print(paste("For", x, "y_hat(x) = ", y_hat(x)))
```

```
## [1] "For 2623 y_hat(x) =  534968.37"
```

```
print(paste("That is, for a house of your variable ", x, "the estimated price is $", y_hat(x)))
```

```
## [1] "That is, for a house of your variable  2623 the estimated price is $ 534968.37"
```

Please only submit your work if you attended the class and worked with other students; this is not an online course

Complete the handout as a group, and turn it in to the instructor