YData: Introduction to Data Science

Lecture 31: least squares

Overview

Quick review of correlation

Linear regression

- Linear predictions
- Relationship to the correlation coefficient

If there is time: least squares

- Errors (residuals)
- Minimizing the root mean squared error

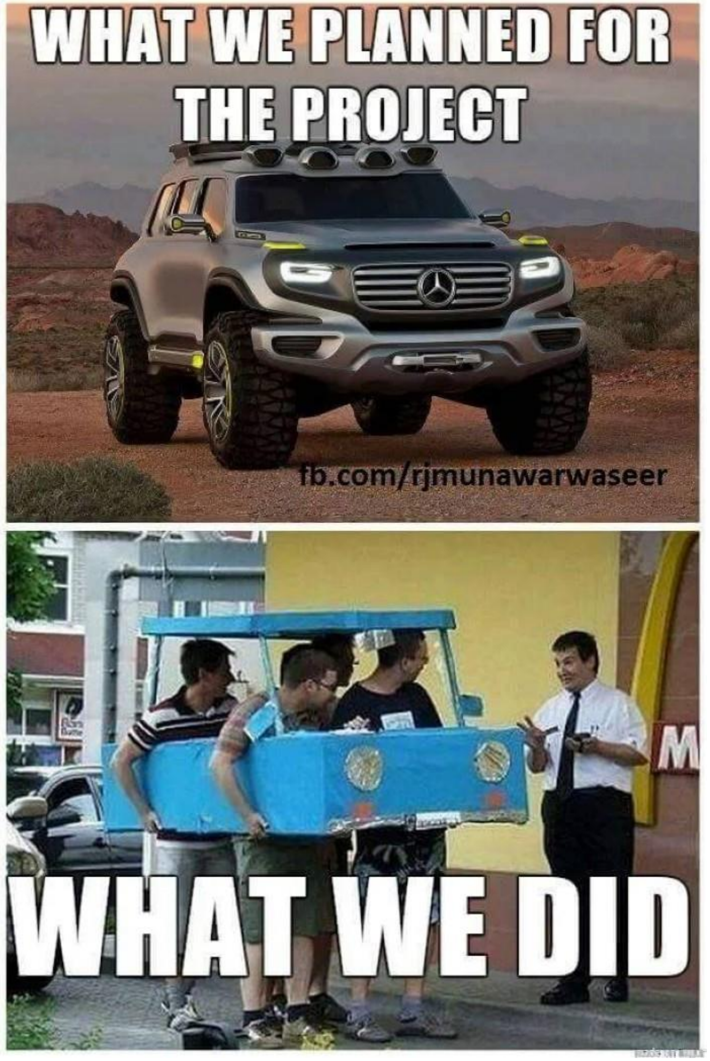
Announcements

Project 3 has been posted

- It is due Wednesday the 27th
  - (rather than on Friday the 22nd)

Homework 9 has been posted

- It is due on Sunday the 17th



Quick review of correlation

Review: the correlation coefficient

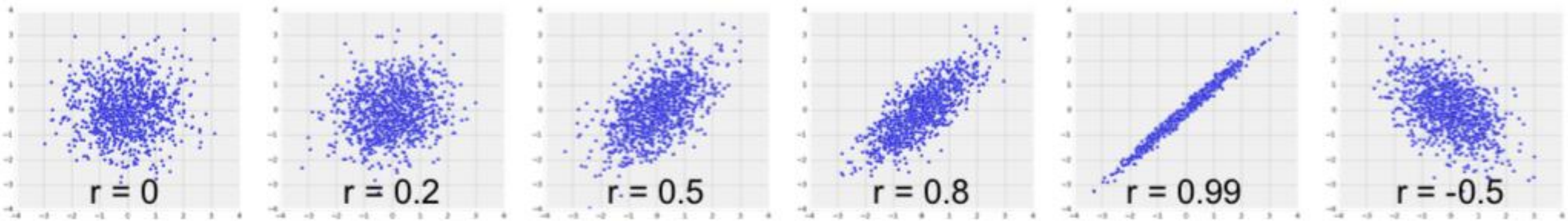
Measures linear association

Based on standard units: $$r = \frac{1}{(n-1)} \sum_{i=1}^{n} \left(\frac{x_i - \overline{x}}{s_x} \right) \left(\frac{y_i - \overline{y}}{s_y} \right)$$

-1 ≤ r ≤ 1

- r = 1: scatter is perfect straight line sloping up
- r = -1: scatter is perfect straight line sloping down
- r = 0: No linear association; uncorrelated

Let's review this in Jupyter!

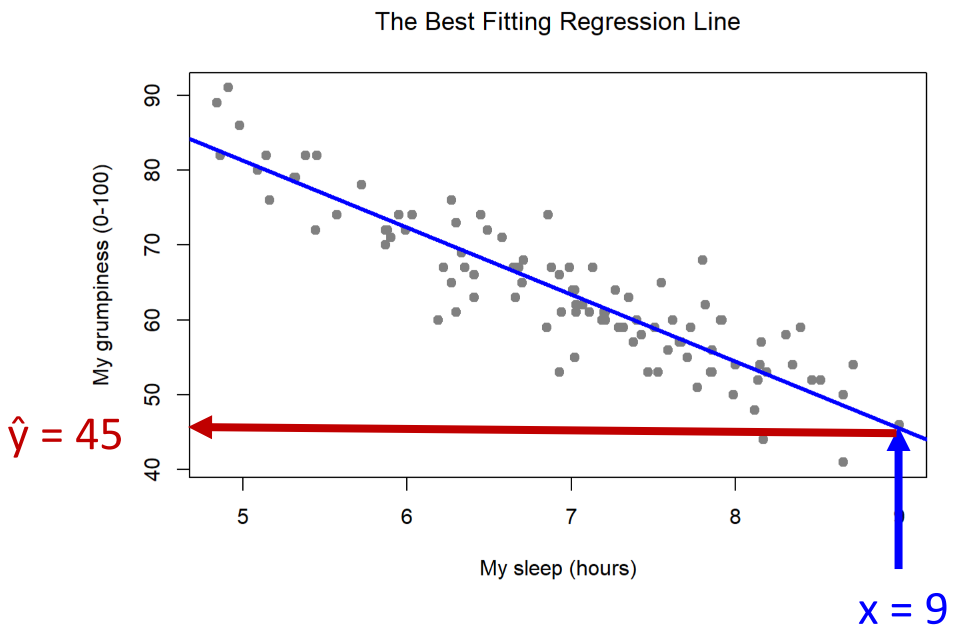


Linear regression

Regression

Regression is method of using one variable ***x*** <u>*to predict*</u> the value of a second variable ***y***

- i.e., $\hat{y} = f(x)$



In **linear regression** we fit a <u>line</u> to the data, called the **regression line**

Lines can be expressed by a slope and intercept:

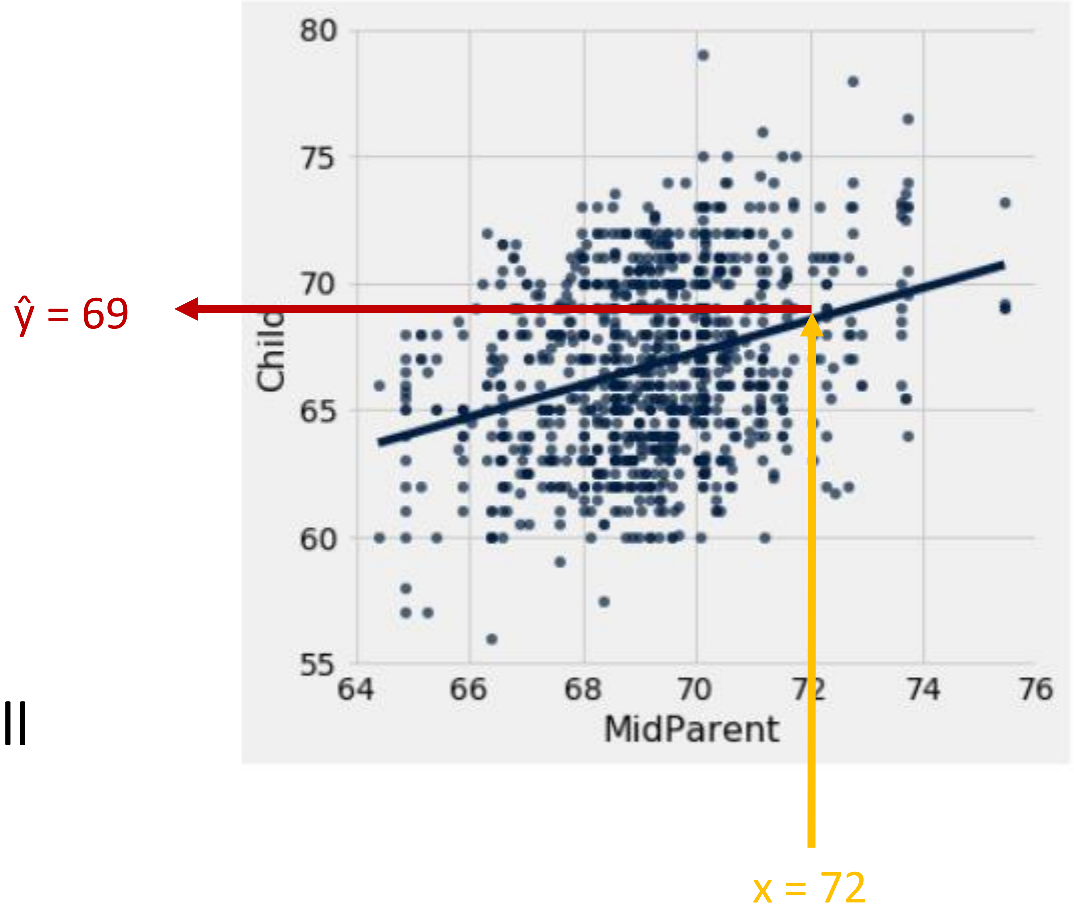
$$\hat{y} = \text{slope} \cdot x + \text{intercept}$$

Regression predictions

The regression line predicts an "average" value:

- For a given x value, the average y could be considered the "best" prediction

Example: Take all children whose midparent height is 72 standard unit. The average height of these children is somewhat less than 70 inches



It doesn't say that all of these children will be somewhat less than 70 inches in height. Some will be taller, and some will be shorter.

Let's explore this in Jupyter!

Slope and intercept

Regression with standardized units

Suppose we standardize our x and y variables through a z-score transformation:

- $y_{(SU)} = (y - \bar{y})/SD_y$ where $\bar{y}$ and SD_y are the mean and SD of y
- $x_{(SU)} = (x - \bar{x})/SD_x$ where $\bar{x}$ and SD_x are the mean and SD of x

The we can relate our predictions of these standardized x and y variables to the correlation coefficient r:

$$\underbrace{y_{(\mathrm{su})} = \underbrace{r}_{\text{Correlation}} \times x_{(\mathrm{su})}}_{\text{Regression Line}}$$

Regression line

Our equation for the regression line in standardized units is:

Regression Line

$$y_{(\mathrm{su})} = \underbrace{r}_{\text{Correlation}} \times x_{(\mathrm{su})}$$

Expanding the definition of standardized units we have:

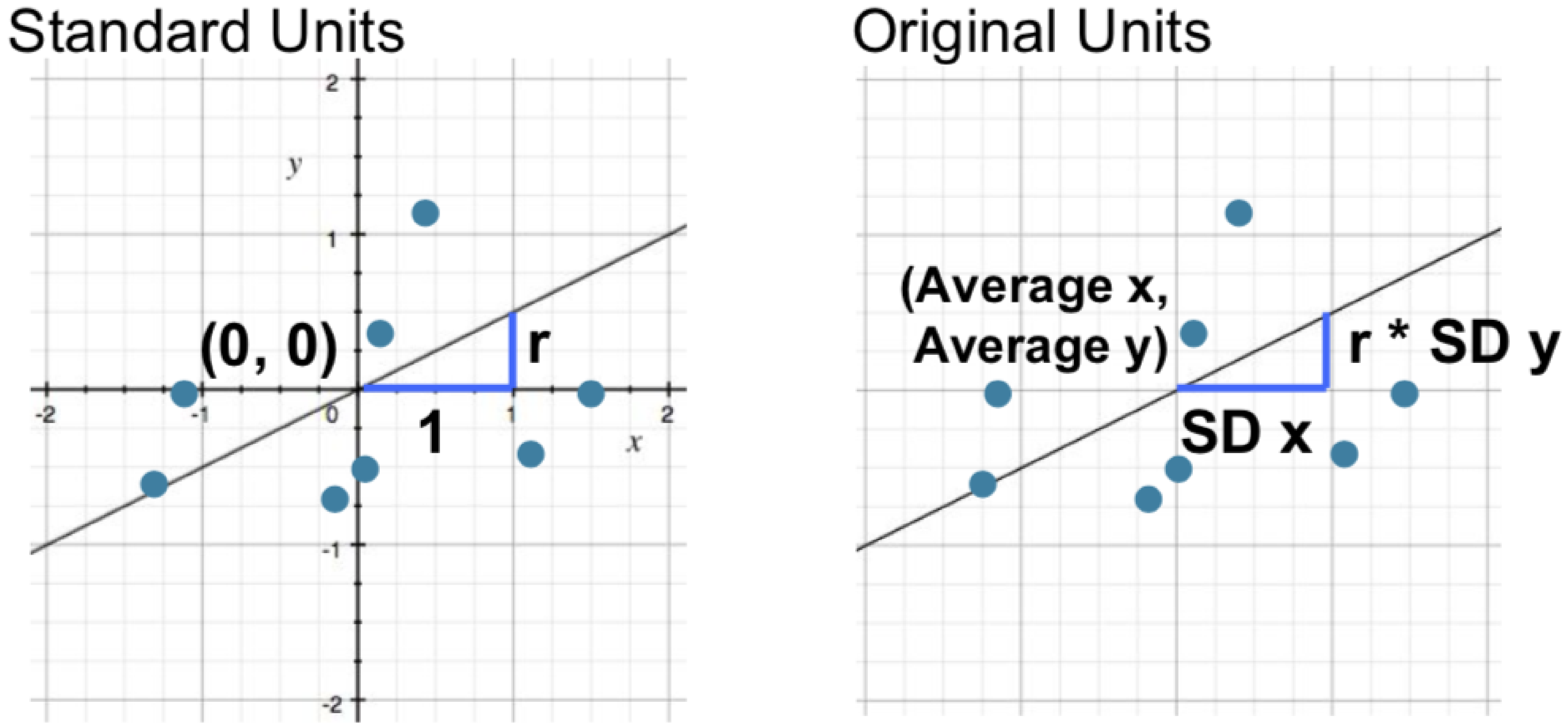
$$(\hat{y} - \bar{y}) / SD_y = r \cdot (x - \bar{x}) / SD_x$$

Solving in our original units: $\hat{y} = \text{slope} \cdot x + \text{intercept}$

$$\text{Slope} = r \cdot SD_y / SD_x$$

$$\text{Intercept} = \bar{y} - \text{slope} \cdot \bar{x}$$

Regression line



Regression to the mean

Our equation for the regression line in standardized units is:

Regression Line

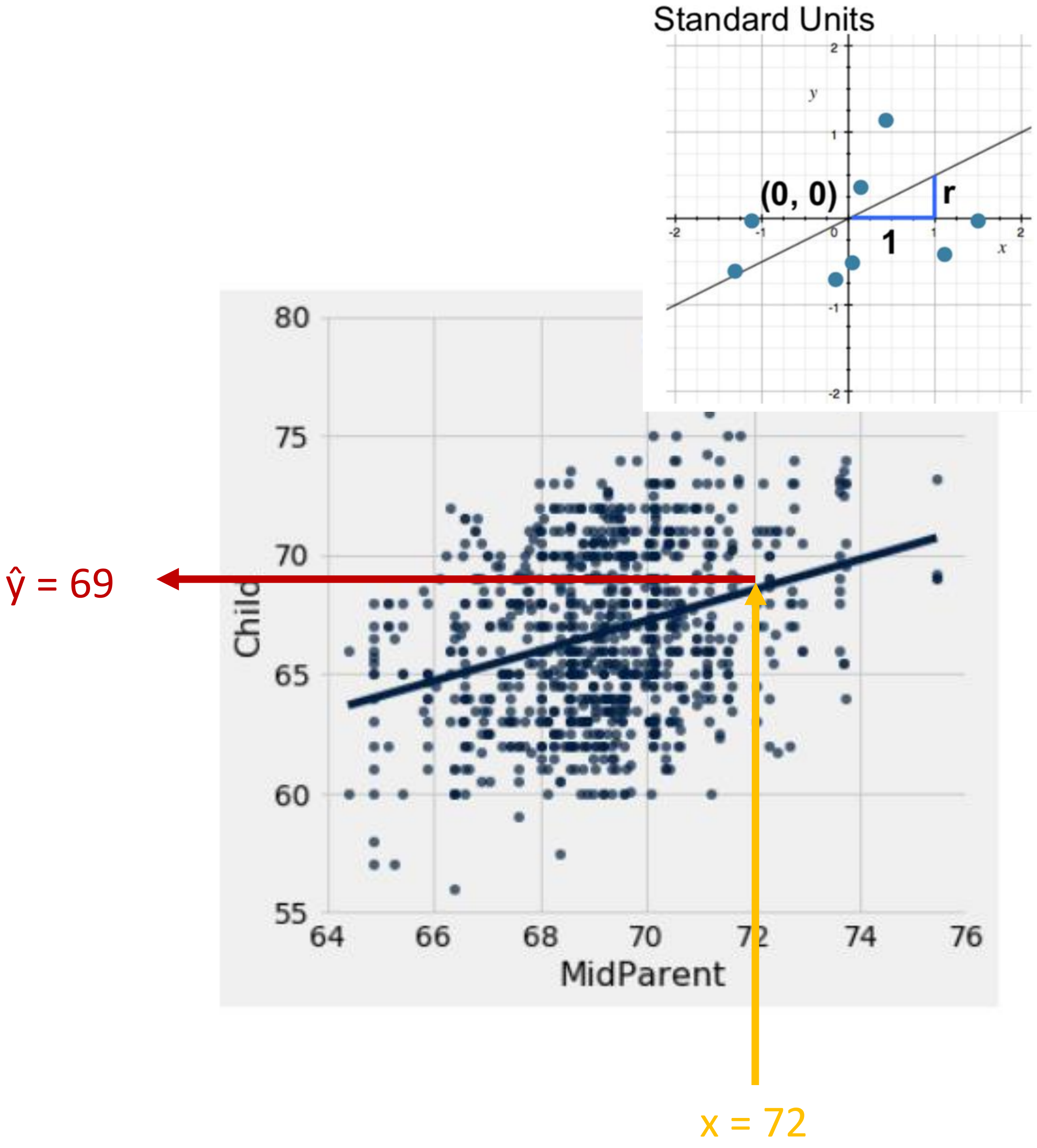
$$y_{(\text{su})} = r \times x_{(\text{su})}$$

Correlation

Because -1 ≤ r ≤ 1 this means that standardized predicted y values will be closer to their mean the than standardized x values used for the prediction

This phenomenon is called "regression to the mean"

- Galton called it "regression to mediocrity"



Let's explore this in Jupyter!

Least Squares

Errors in estimation

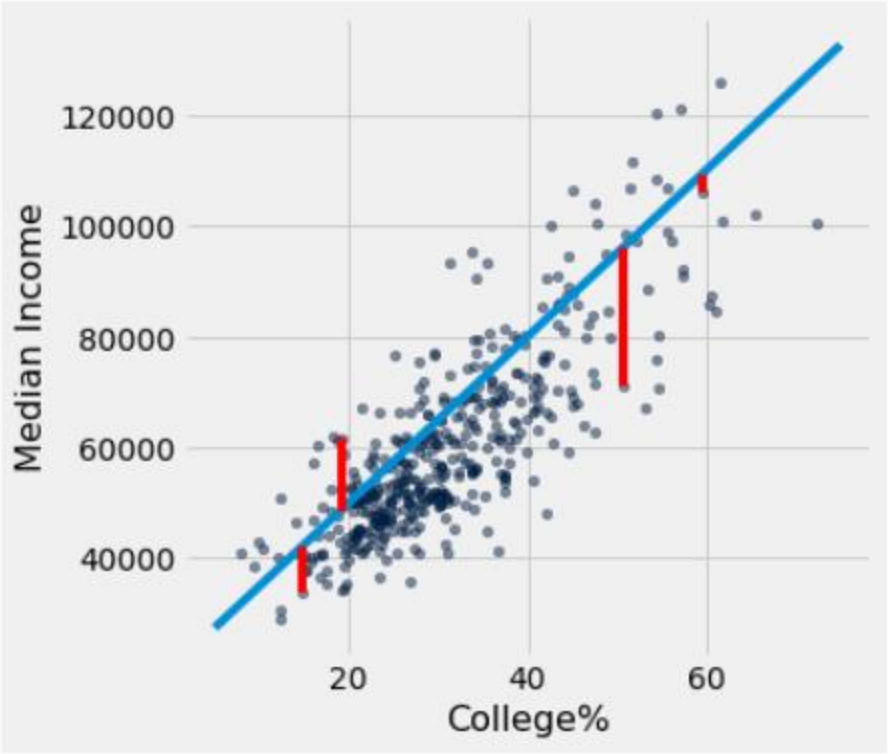
error = actual value - estimate

$e_i = y_i - \hat{y}_i$

Typically, some errors are positive and some negative

To measure the rough size of the errors we calculate the root mean square error (RMSE):

- Square the errors to eliminate cancellation
- Take the mean of the squared errors
- Take the square root to fix the units



$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Let's explore this in Jupyter!

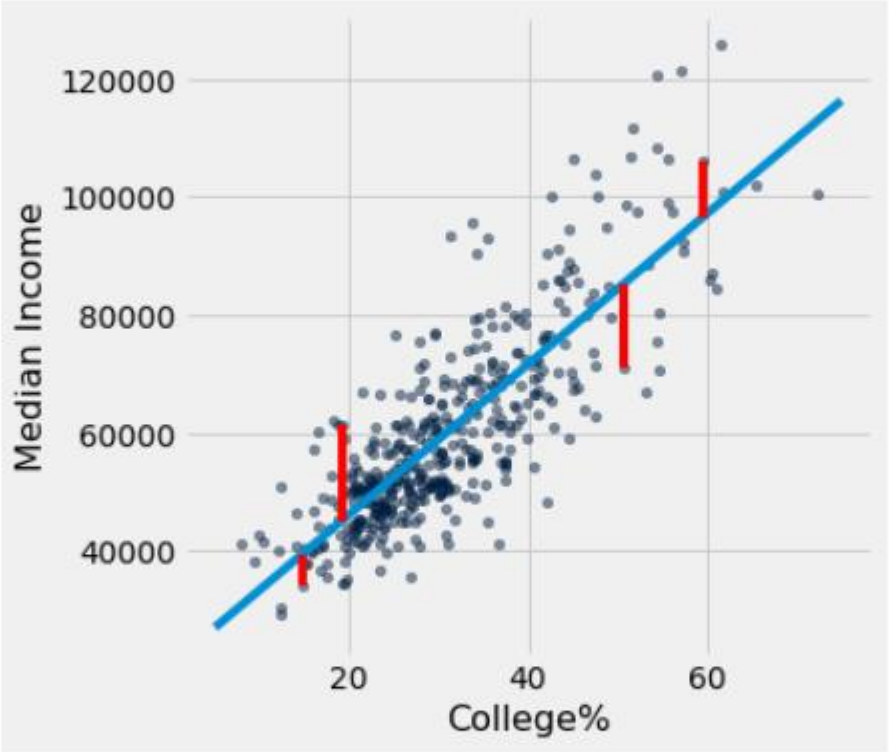
Least Squares Line

Minimizes the root mean squared error (RMSE) among all lines

- Equivalently, minimizes the mean squared error (MSE) among all lines

Names:

- “Best fit" line
- Least squares line
- Regression line



Numerical optimization

Numerical minimization is approximate but effective

Much of machine learning is based on numerical minimization

If the function mse(a, b) returns the MSE of estimation using the line “estimate = ax + b"

- then minimize(mse) returns array $[a_0, b_0]$
- a_0 is the slope and b_0 the intercept of the line that minimizes the MSE among lines with arbitrary slope a and arbitrary intercept b (that is, among all lines)

Let's explore this in Jupyter!