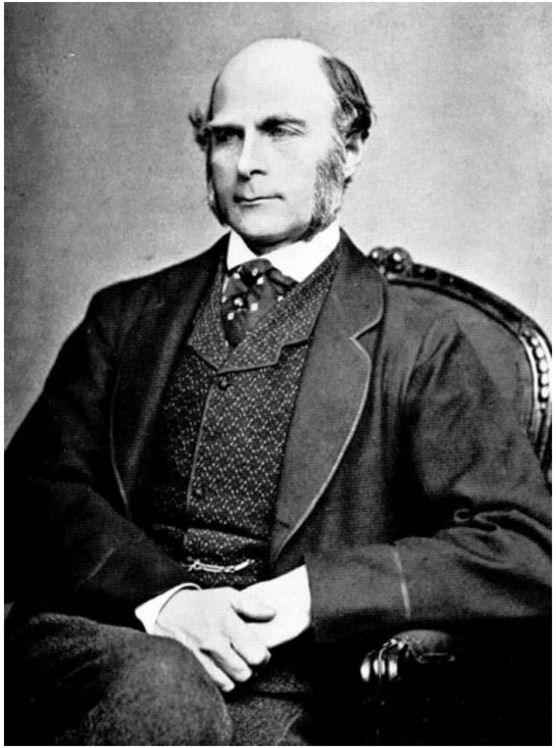
Linear Regression

Let’s learn something!

History

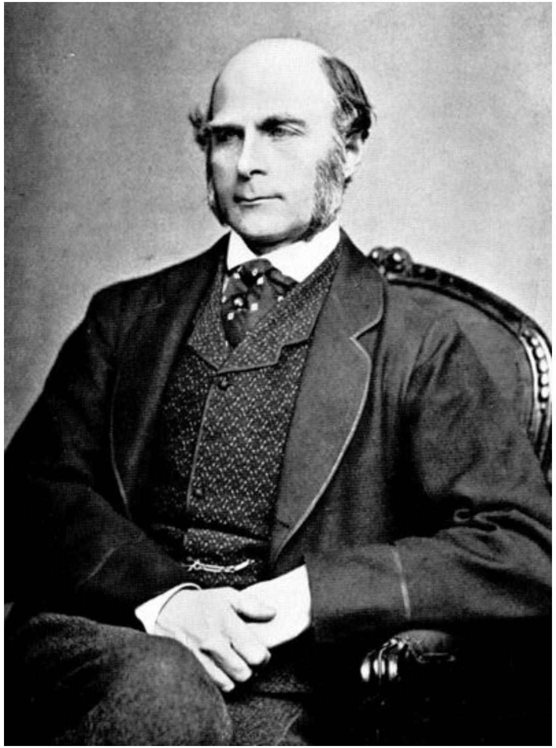
This all started in the 1800s with a guy named Francis Galton. Galton was studying the relationship between parents and their children. In particular, he investigated the relationship between the heights of fathers and their sons.



History

What he discovered was that a man's son tended to be roughly as tall as his father.

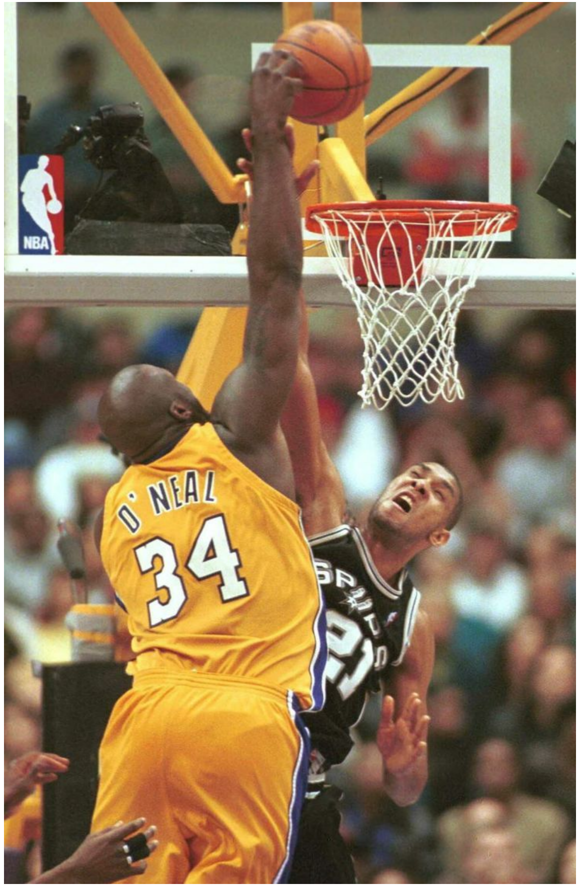
However Galton's breakthrough was that the son's height **tended to be closer to the overall average** height of all people.



Example

Let's take Shaquille O'Neal as an example. Shaq is really tall:7ft 1in (2.2 meters).

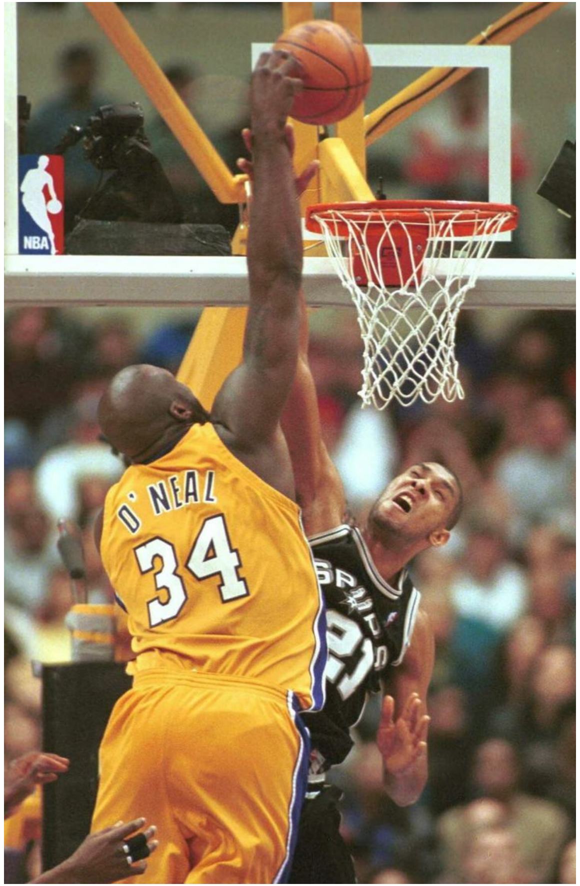
If Shaq has a son, chances are he'll be pretty tall too. However, Shaq is such an anomaly that there is also a very good chance that his son will be **not be as tall as Shaq**.



Example

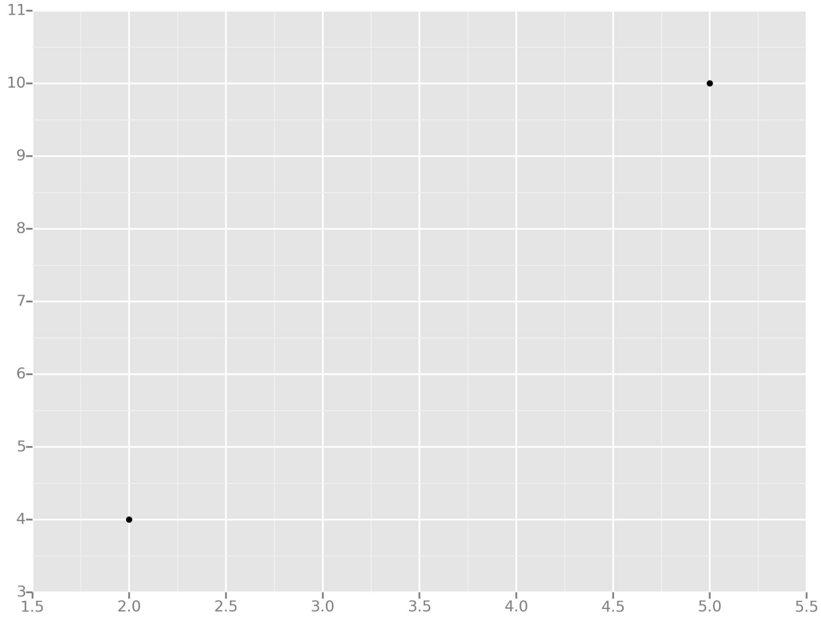
Turns out this is the case: Shaq's son is pretty tall (6 ft 7 in), but not nearly as tall as his dad.

Galton called this phenomenon **regression**, as in "A father's son's height tends to regress (or drift towards) the mean (average) height."



Example

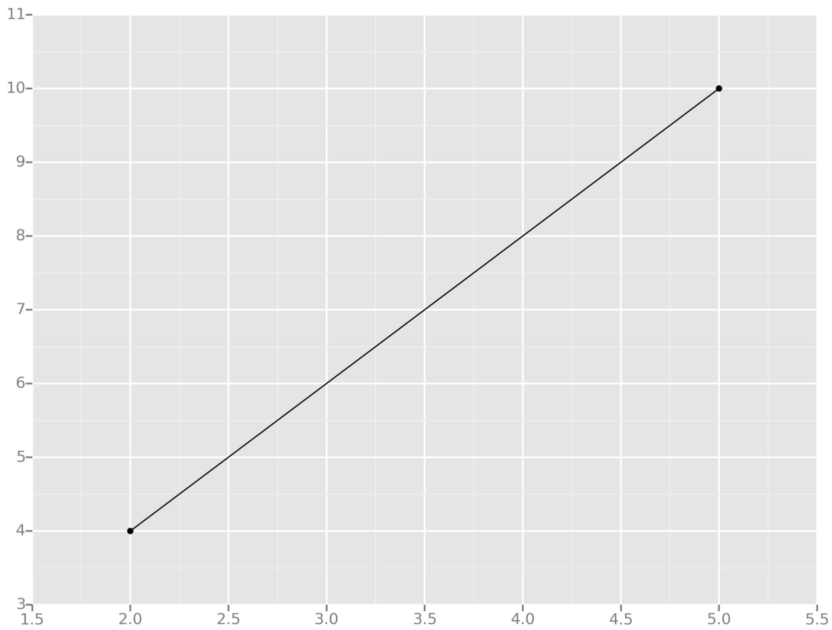
Let's take the simplest possible example: calculating a regression with only 2 data points.



Example

All we're trying to do when we calculate our regression line is draw a line that's as close to every dot as possible.

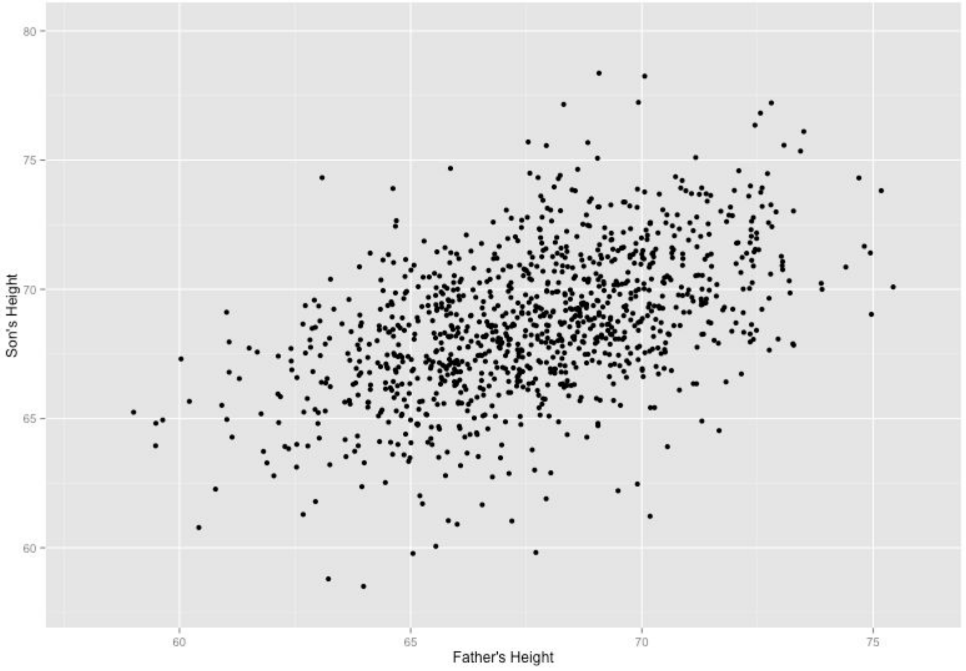
For classic linear regression, or "Least Squares Method", you only measure the closeness in the "up and down" direction



Example

Now wouldn't it be great if we could apply this same concept to a graph with more than just two data points?

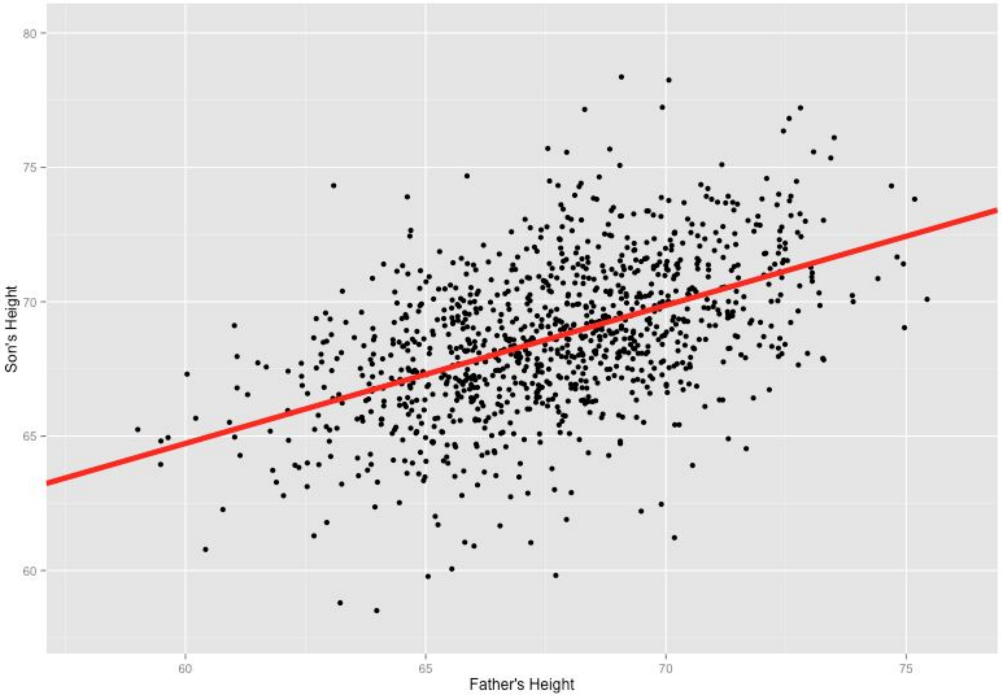
By doing this, we could take multiple men and their son's heights and do things like tell a man how tall we expect his son to be...before he even has a son!



Example

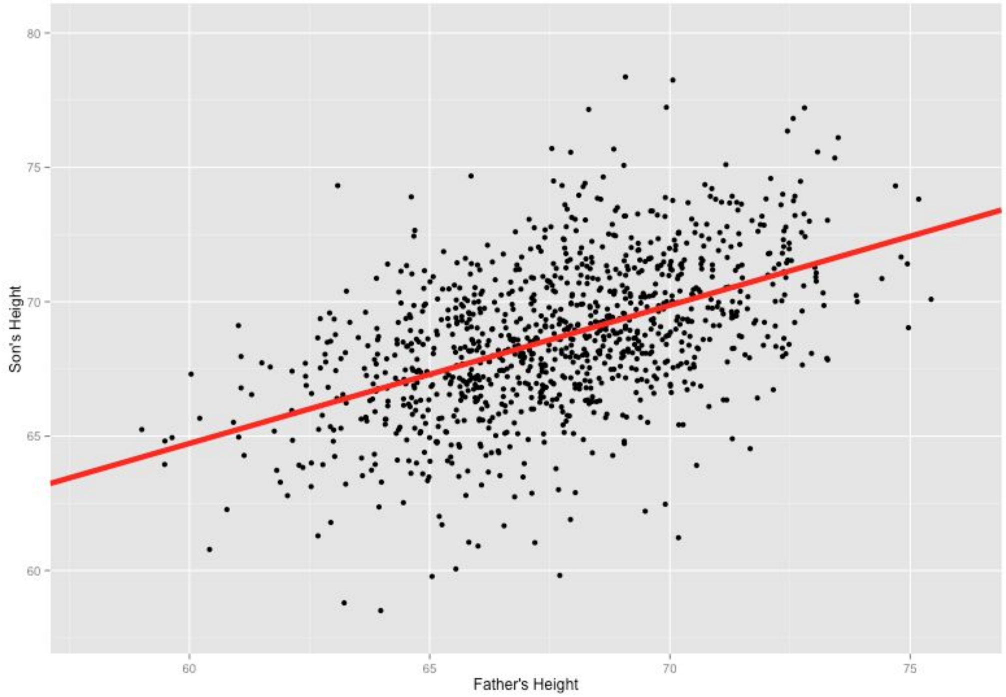
Our goal with linear regression is to **minimize the vertical distance** between all the data points and our line.

So in determining the **best line**, we are attempting to minimize the distance between **all** the points and their distance to our line.



Example

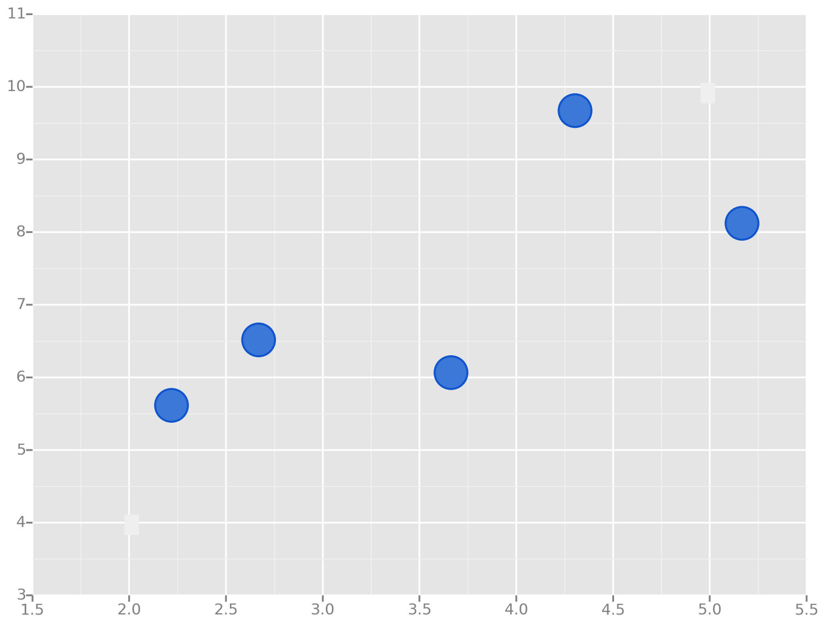
There are lots of different ways to minimize this, (sum of squared errors, sum of absolute errors, etc), but all these methods have a general goal of minimizing this distance.



Example

For example, one of the most popular methods is the least squares method.

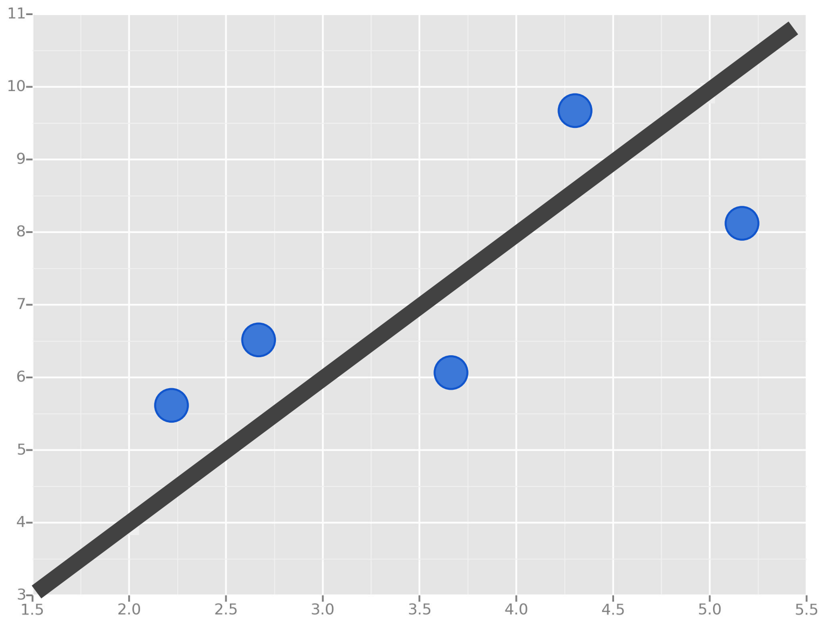
Here we have blue data points along an x and y axis.



Example

Now we want to fit a linear regression line.

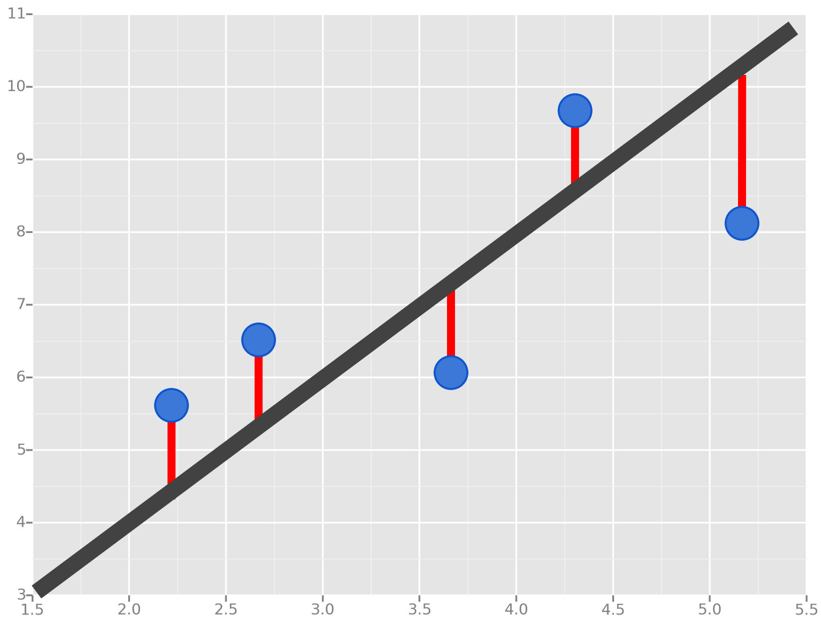
The question is, how do we decide which line is the best fitting one?



Example

We'll use the Least Squares Method, which is fitted by minimizing the ***sum of squares of the residuals.***

The residuals for an observation is the difference between the observation (the y-value) and the fitted line.



Python and Spark

- Now let’s get an idea of how to implement this idea with PySpark!
- Will ease into all of this by checking out the simpler documentation example first!

Evaluating Regression

Python and Spark

- Let’s take a quick break to discuss evaluating Regression Models
- Not just Linear Regression, but any model that attempts to predict continuous values (unlike categorical values, which is classification)

Python and Spark

- You may have heard of some evaluation metrics like accuracy or recall.
- These sort of metrics aren’t useful for regression problems, we need metrics designed for **continuous** values!

Python and Spark

- Let's discuss some of the most common evaluation metrics for regression:
    - Mean Absolute Error
    - Mean Squared Error
    - Root Mean Square Error
    - R Squared Values

Python and Spark

- Mean Absolute Error (MAE)
  - This is the mean of the absolute value of errors.
  - Easy to understand, just average er

$$\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Python and Spark

- Mean Squared Error (MSE)
  - This is the mean of the squared errors.
  - Larger errors are noted more than wi ... opular.

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

Python and Spark

- Root Mean Absolute Error (RMSE)
    - This is the root of the mean of the squared errors.
    - Most popular (has same units as y)

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

Python and Spark

- R Squared Values (also known as the coefficient of determination)
- Not quite an error metric, more of a statistical measure of your regression model.

Python and Spark

- By itself, R^2 , won’t tell you the “whole story”.
- In a basic sense it is a measure of how much variance your model accounts for.
- Between 0-1 (0% to 100%)
- There are also different ways of obtaining R^2 , such as adjusted R squared.

Python and Spark

- A full analysis and explanation of interpreting R Squared is outside the scope of this course, but check out the resource links for more information on this statistical topic.

Python and Spark

- Just keep in mind that R Squared can enhance your understanding of a model, or help you compare models, but it shouldn't be your sole source of evaluating a model!
- Let's continue on to our code example!