Regularization

BCSE0105 MACHINE LEARNING

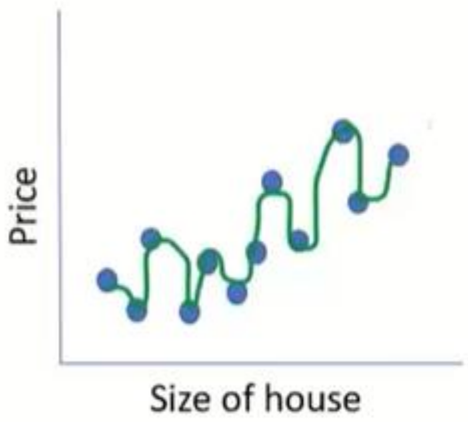
What is Regularization?

House Price Prediction

Underfitting

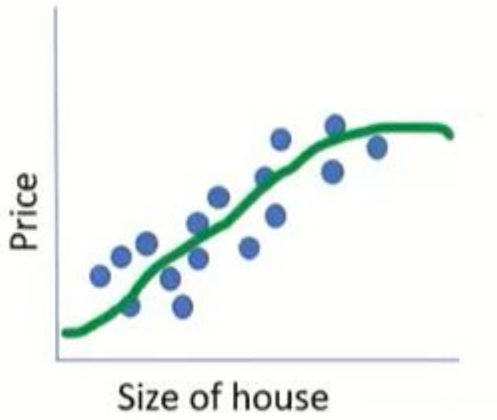
$$\text{Price} = \beta_0 + \beta_1 * \text{size}$$

Overfitting



$$\text{Price} = \beta_0 + \beta_1 * \text{size} + \beta_2 * \text{size}^2 + \beta_3 * \text{size } 3 + \beta_4 * \text{size } 4$$

Good Fit

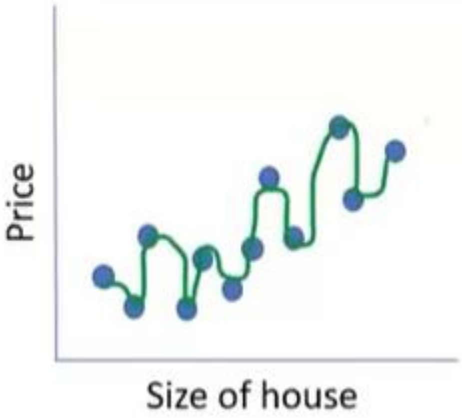


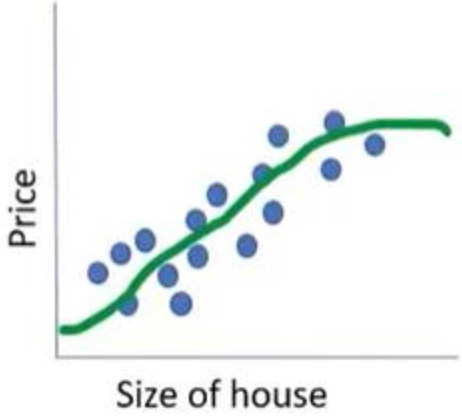
$$\text{Price} = \beta_0 + \beta_1 * \text{size} + \beta_2 * \text{size}^2$$

What is Regularization?

- Constraining a model to make it simpler and reduce the risk of overfitting is called **regularization.**
- A training constraint whose objective is to reduce overfitting and thus improve the model’s ability to generalize
- The amount of regularization to apply during learning can be controlled by a hyperparameter.

Working of Regularization





Price = $\beta_0 + \beta_1 * \text{size} + \beta_2 * \text{size}^2 + \beta_3 * \text{size } 3 + \beta_4 * \text{size } 4$

↓

reduce β_3 and β_4 close to zero

↓

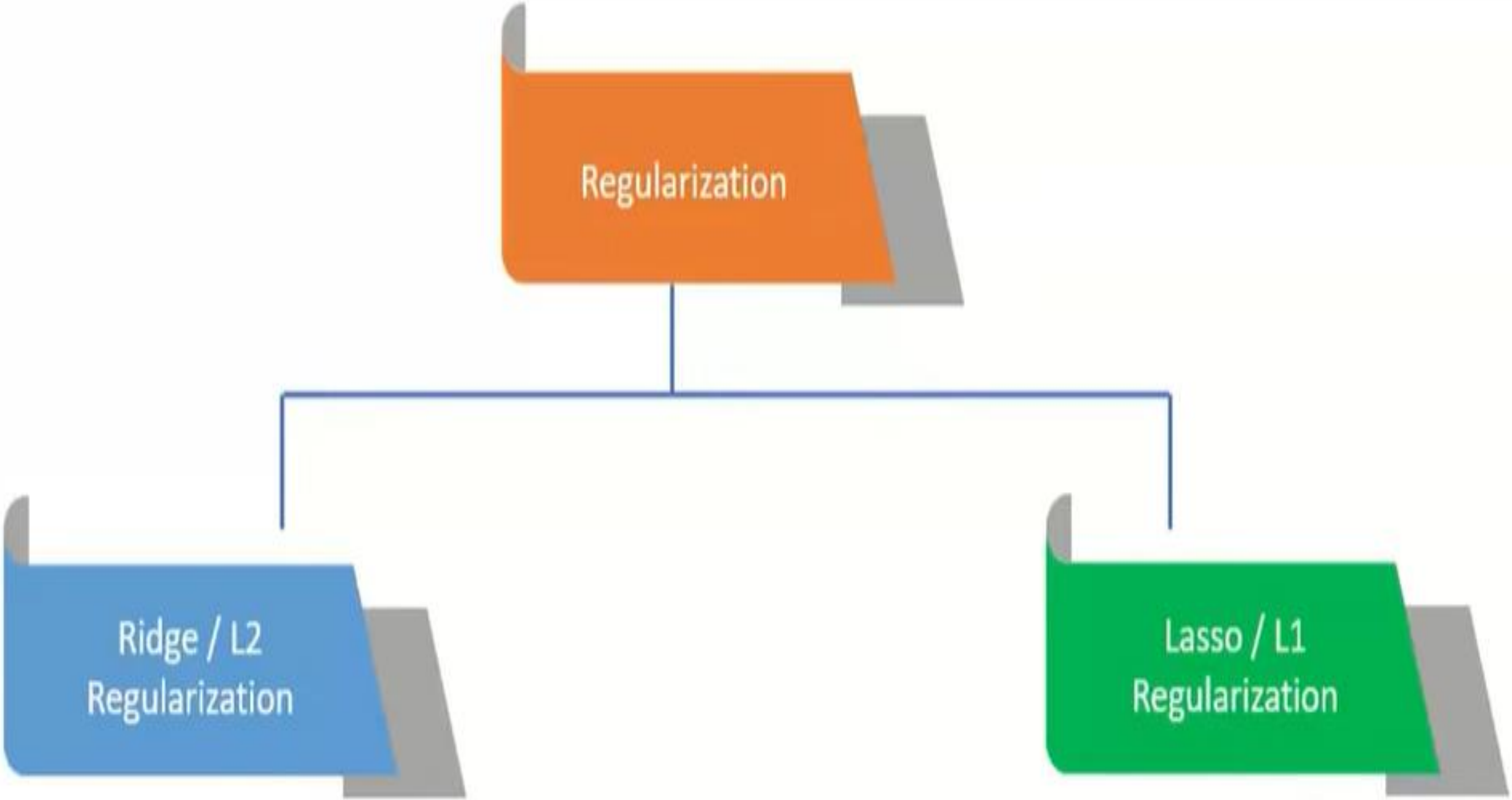
Price = $\beta_0 + \beta_1 * \text{size} + \beta_2 * \text{size}^2$

How to reduce /shrink coefficients

- Use cost function

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_{i\,(actual} - y_{i\,(predicted)})^2$$

This is higher order polynomial equation and our aim is to reduce MSE.
For this we have different **regularization techniques.**


Regularization
Ridge / L2 Regularization
Lasso / L1 Regularization

Regularization Techniques

- Used to reduce overfitting
- Used to generalize the model well

Ridge Regularization

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_{i\,(actual)} - y_{i\,(predicted)}\right)^2$$

$$Loss = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{P} \beta_j^{\ 2}$$

Penalty term regularizes the coefficients

λ = Tuning parameter

Lasso Regularization

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_{i\,(actual)} - y_{i\,(predicted)})^2$$

$$Loss = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda\sum_{j=1}^{P}|\beta_j|$$

Penalty term regularizes the coefficients

λ = Tuning parameter

Which Technique To Use?

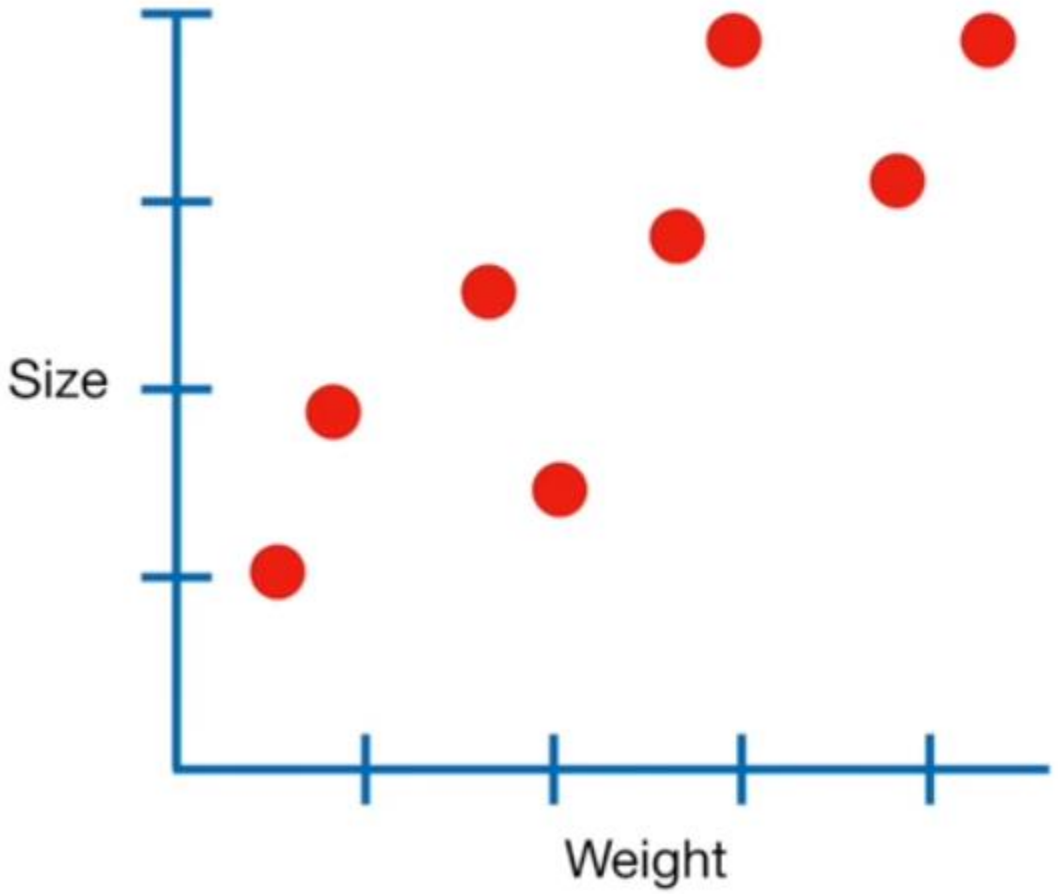
Ridge

Lot of features
In the dataset and
all features have
small coefficients.

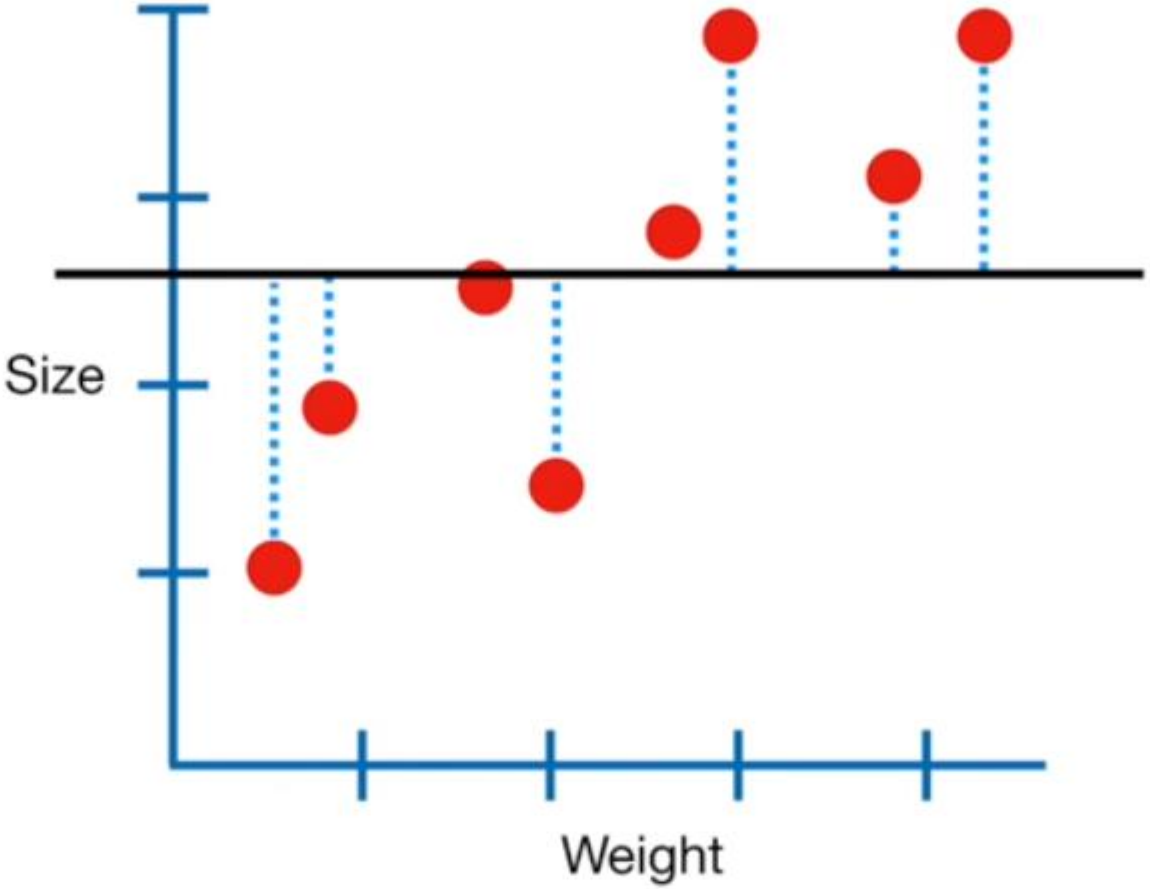
Lasso

Small number of
features and few
features have high
coefficient value.

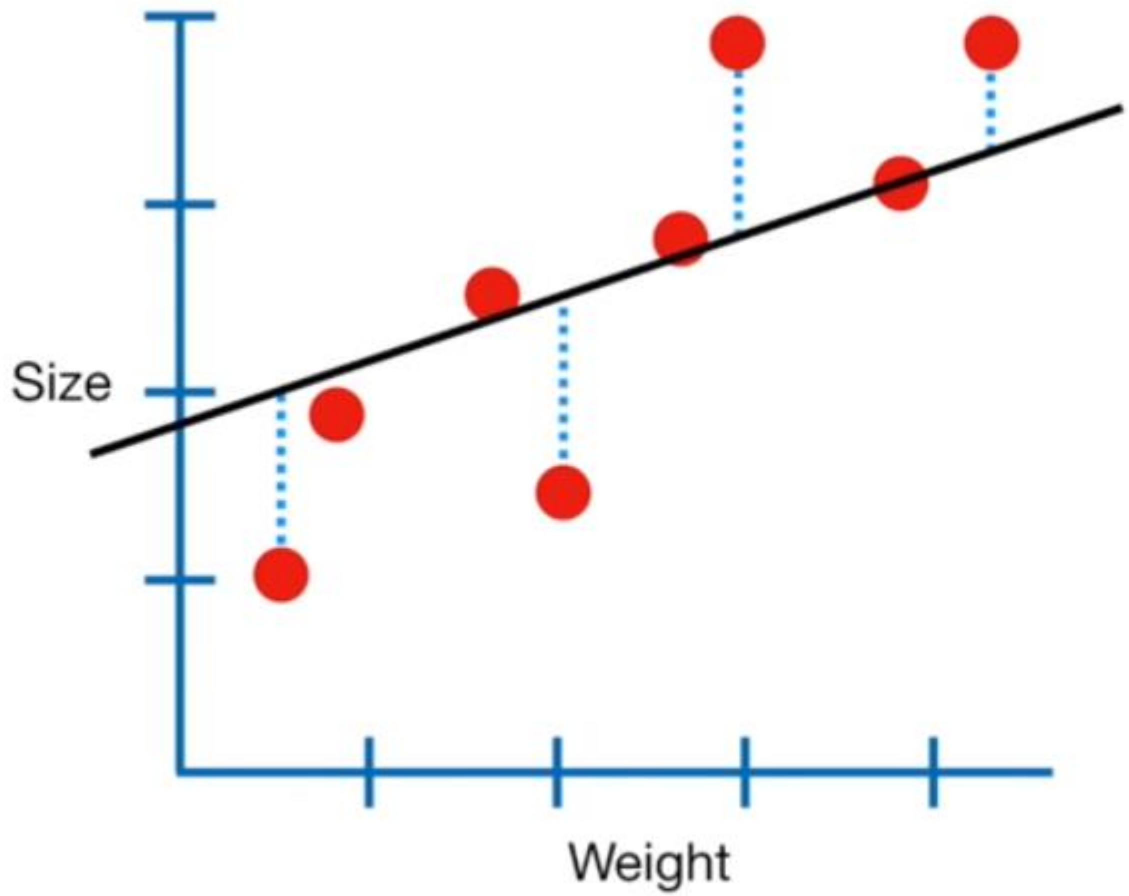
Understanding the Idea behind regularization



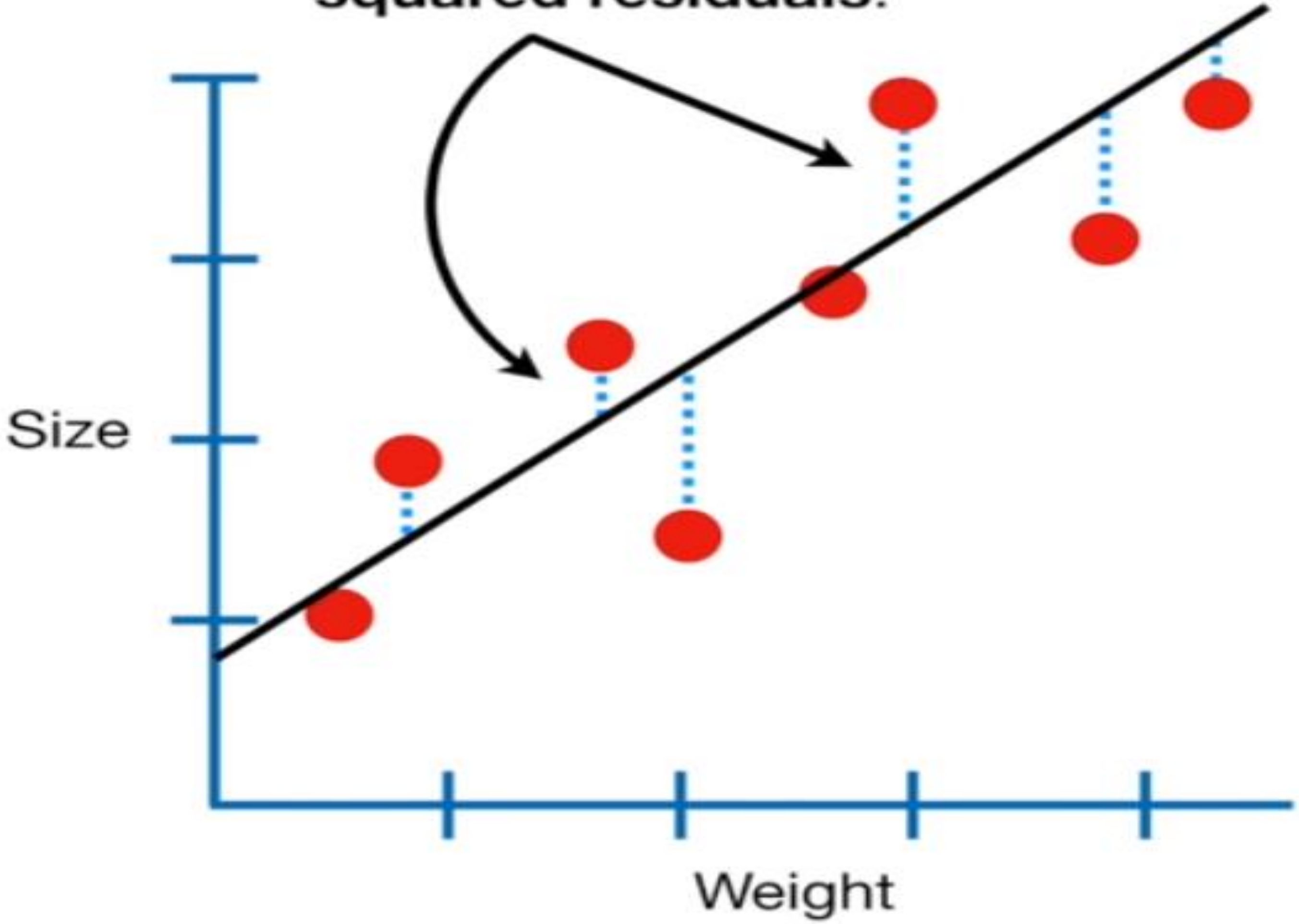
So we'll fit a line to the data using **Least Squares**.



So we'll fit a line to the data using
Least Squares.

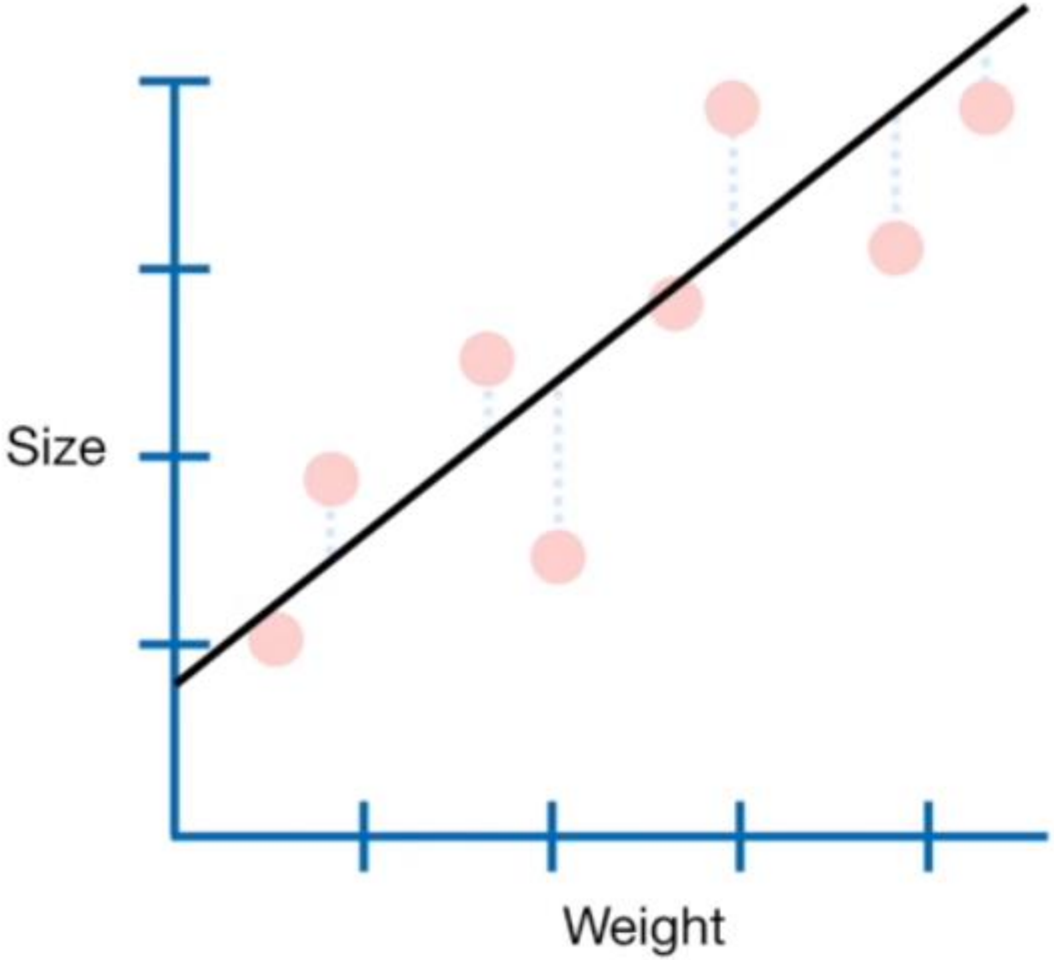


In other words, we find the line that results in the **minimum sum of squared residuals**.

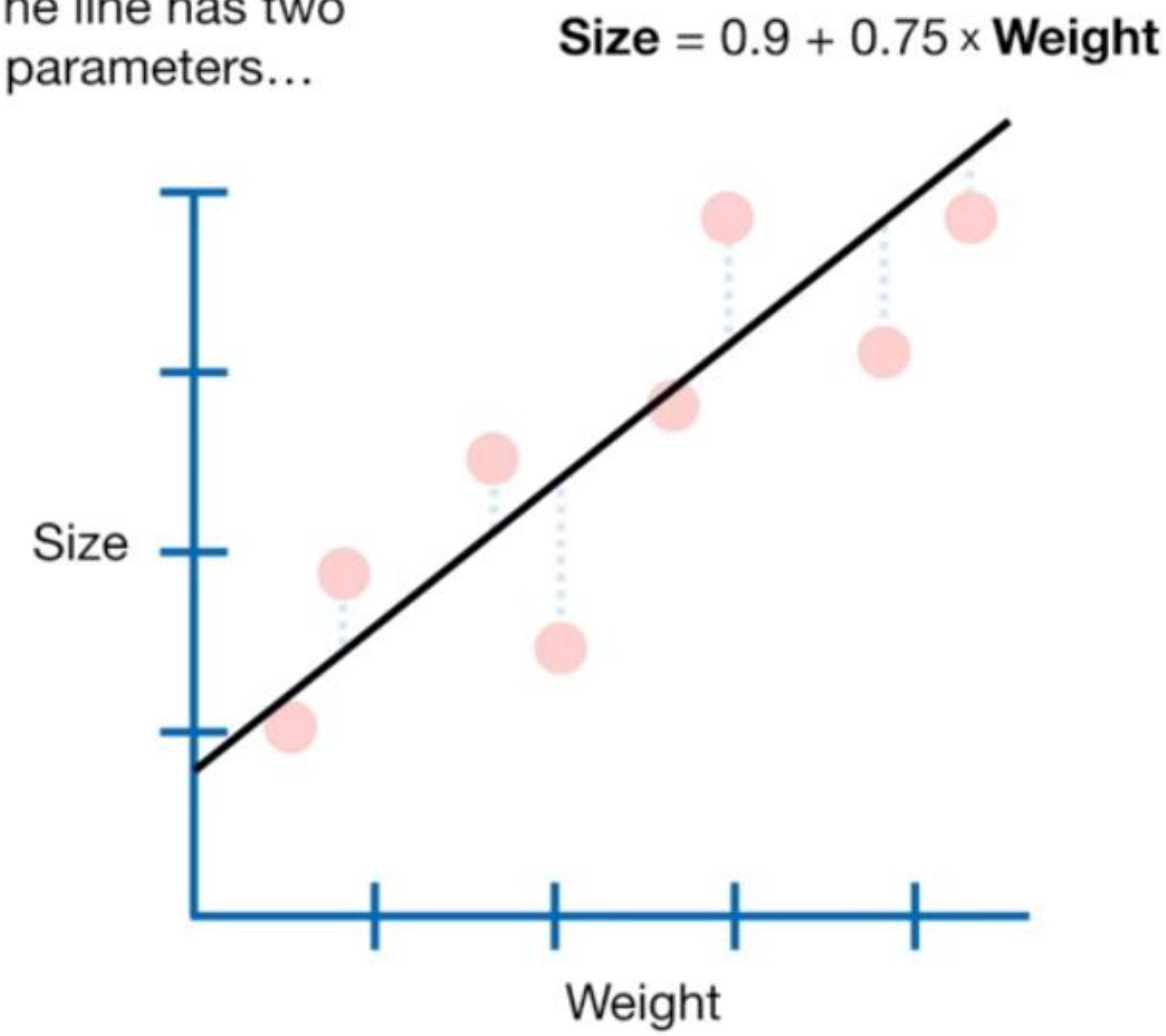


Ultimately, we end up with this equation for the line:

Size = 0.9 + 0.75 x **Weight**

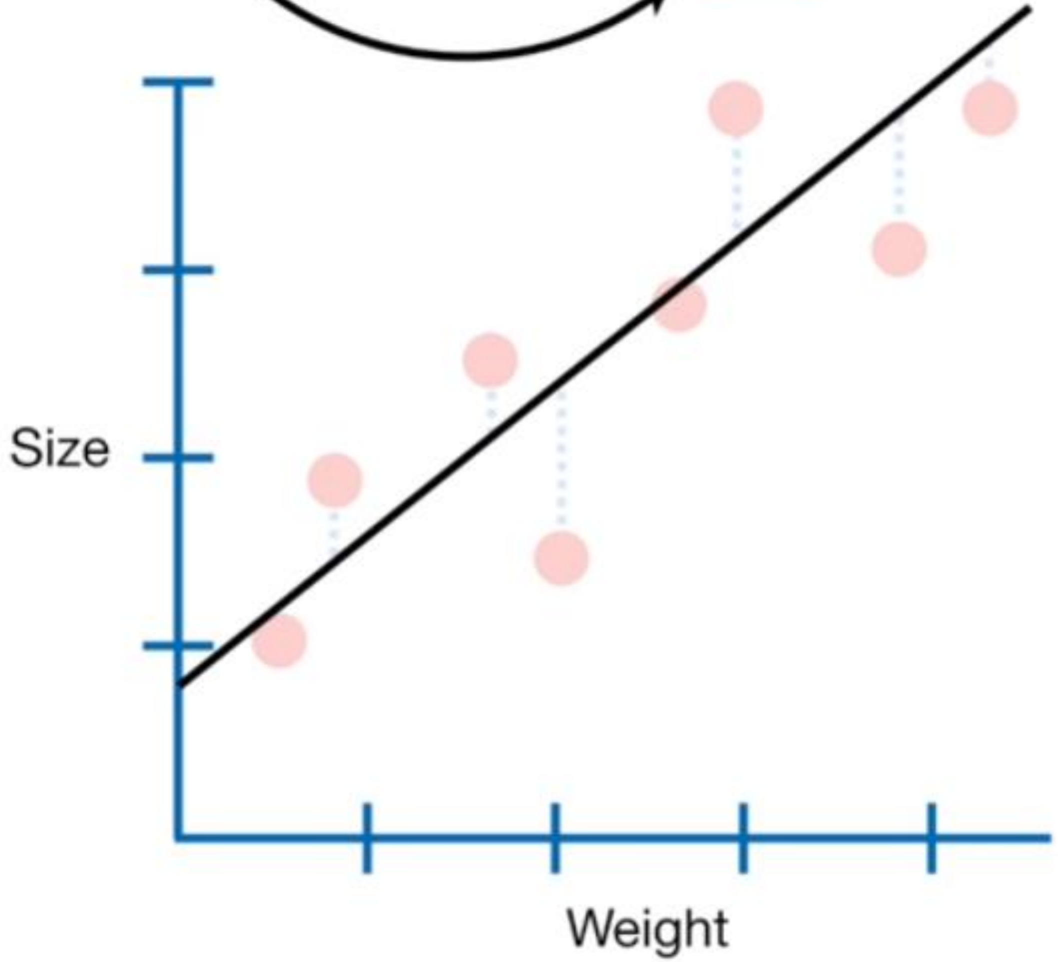


The line has two parameters...



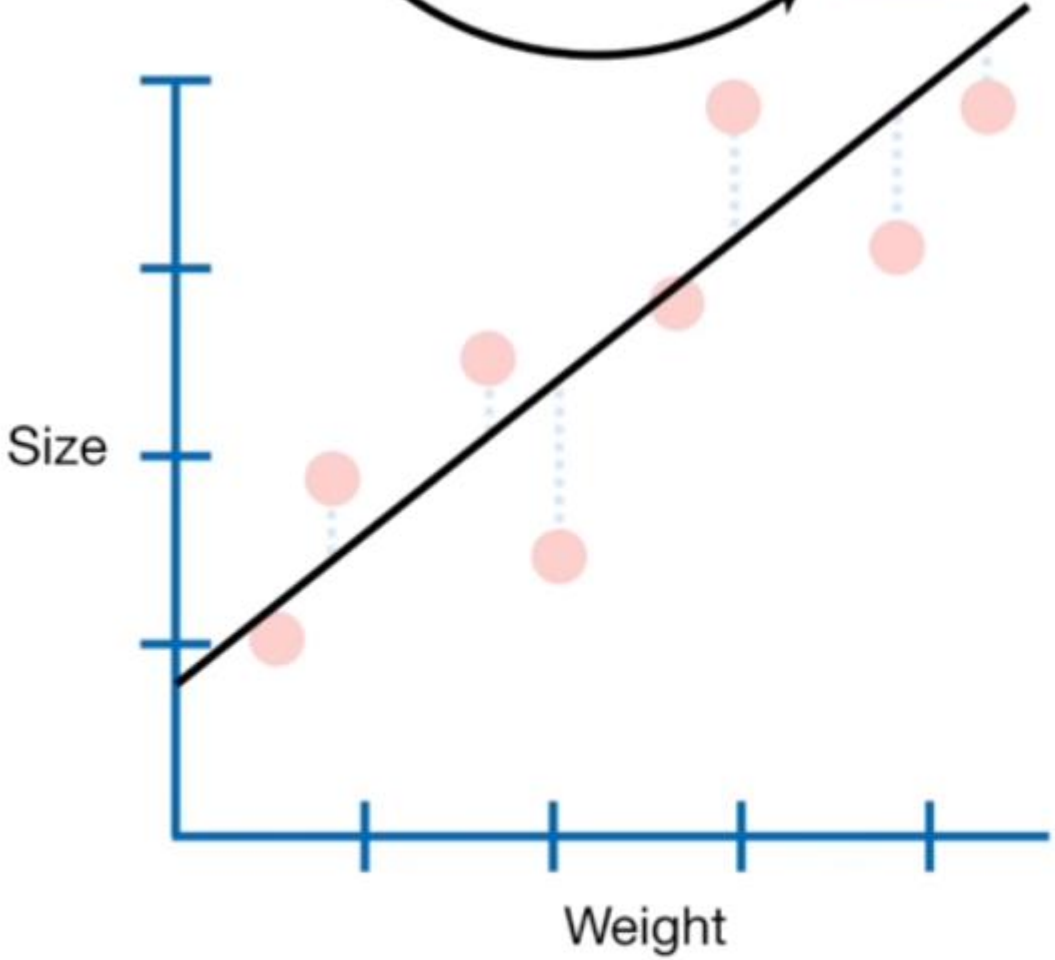
…a **y-axis intercept**…

$$\textbf{Size} = \boxed{0.9} + 0.75 \times \textbf{Weight}$$



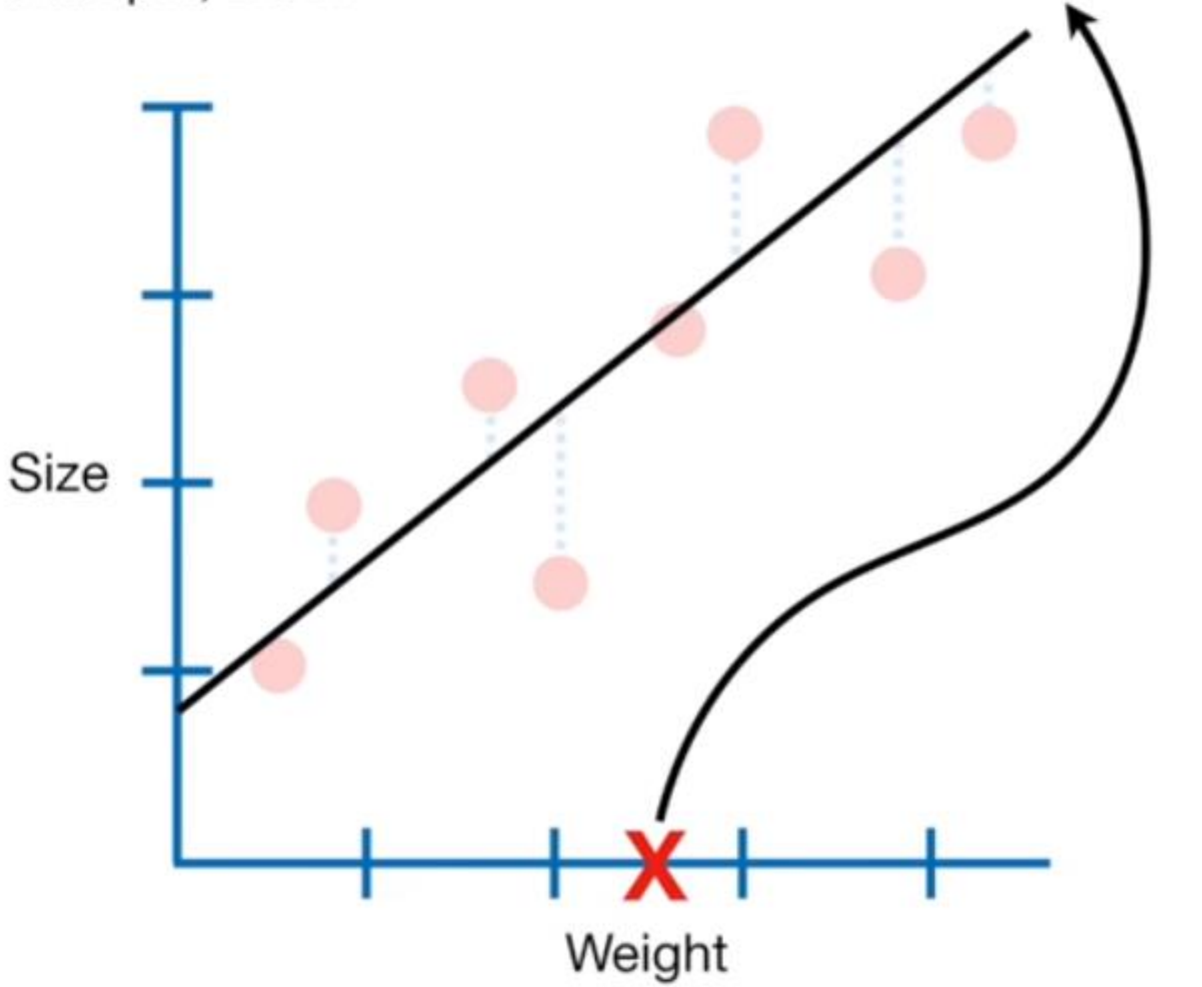
…and a **slope**.

$$\textbf{Size} = 0.9 + \boxed{0.75} \times \textbf{Weight}$$



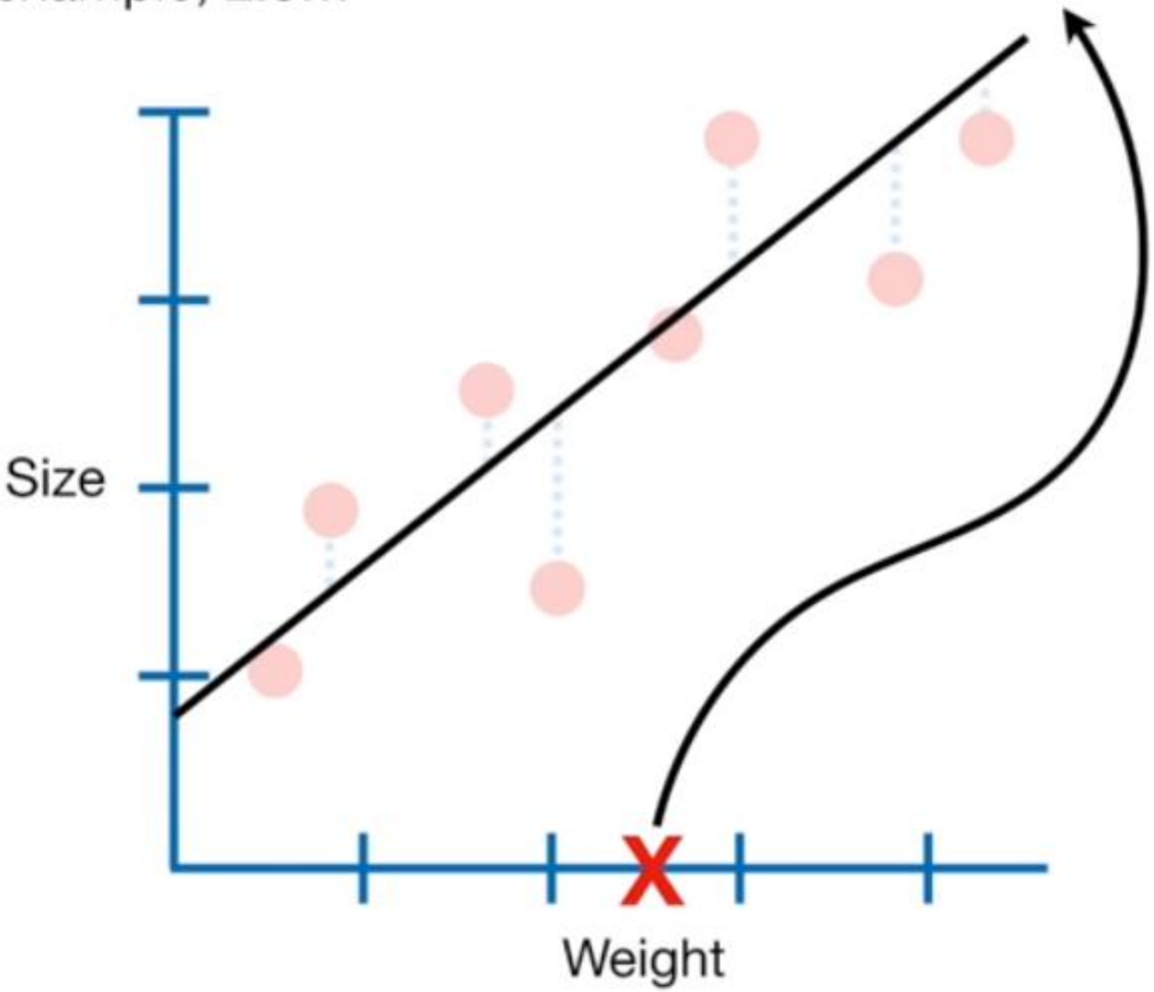
We can plug in a value for **Weight**, for example, **2.5**…

Size = 0.9 + 0.75 x **Weight**

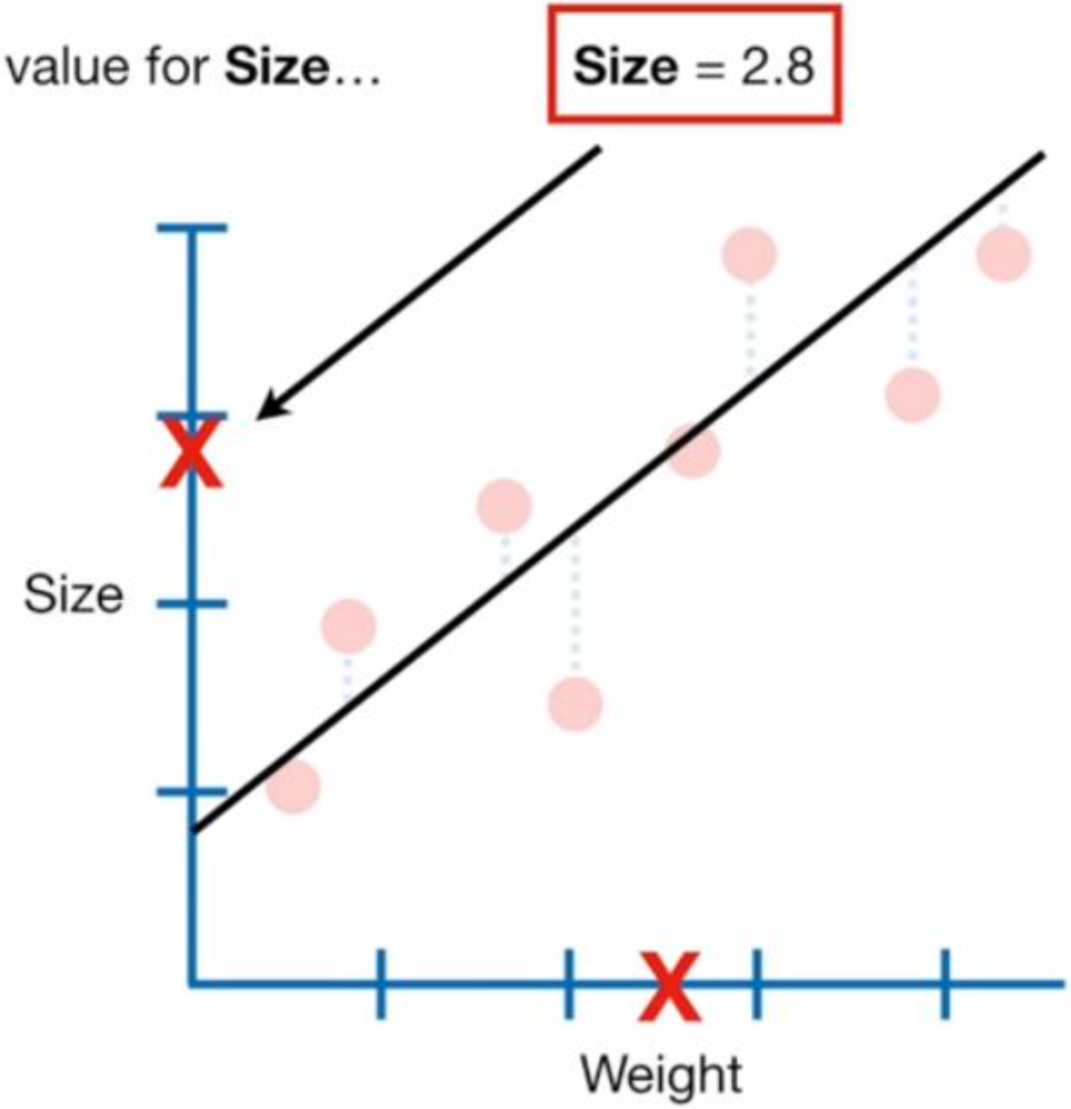


We can plug in a value for **Weight**, for example, **2.5**…

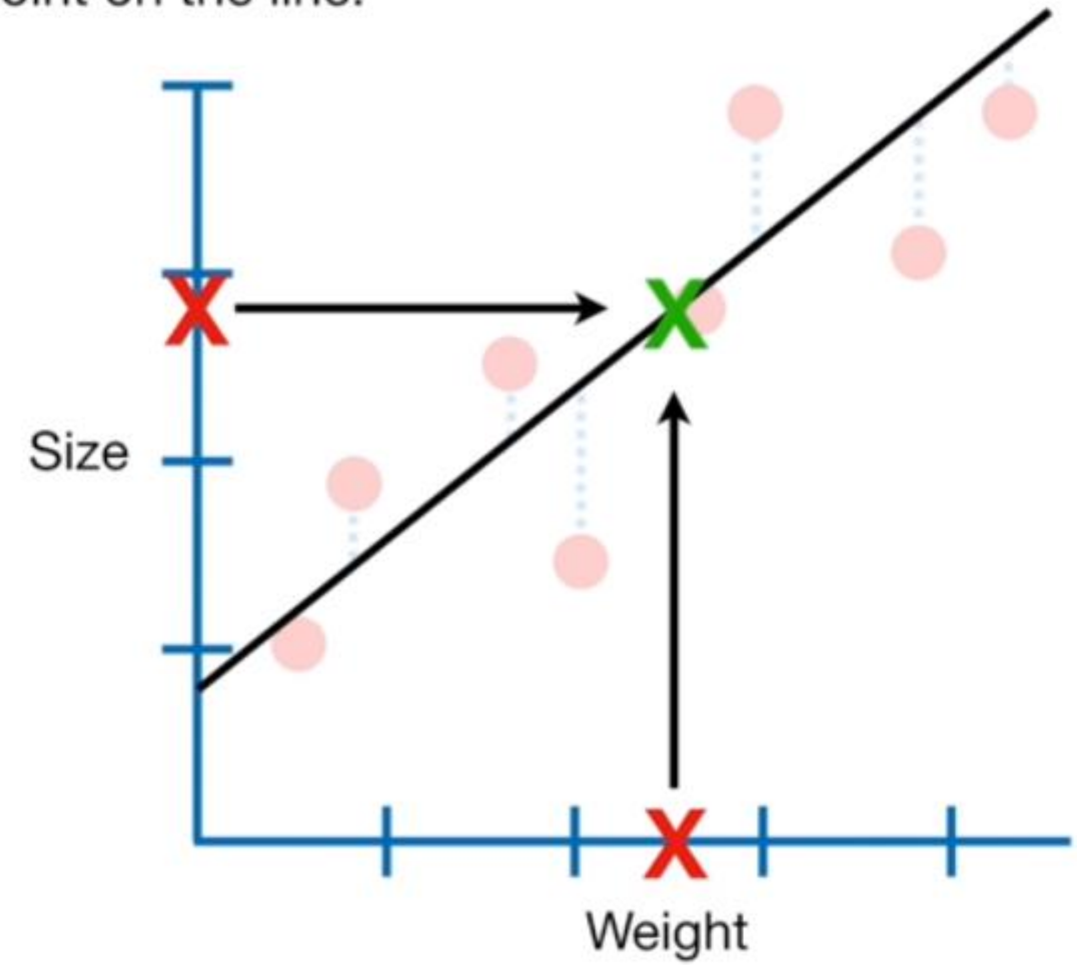
$$\textbf{Size} = 0.9 + 0.75 \times \textbf{2.5}$$



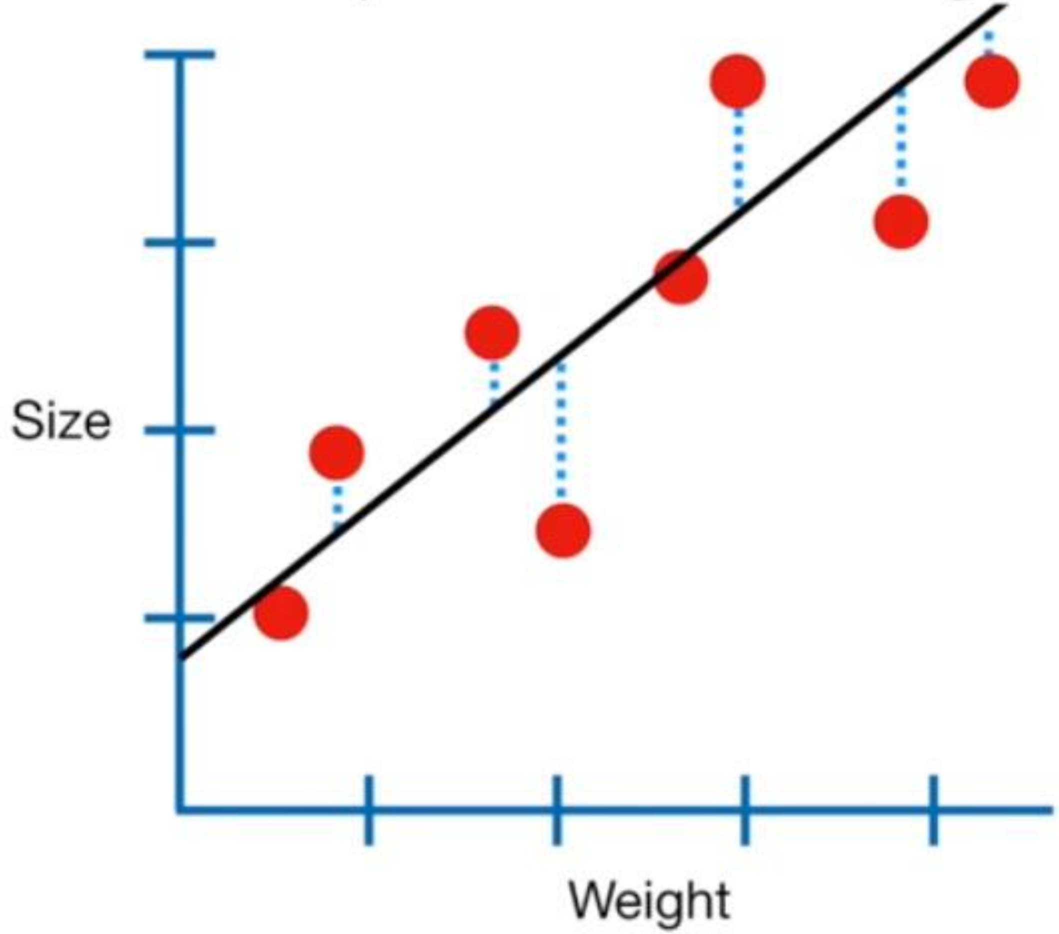
…and get a value for **Size**…



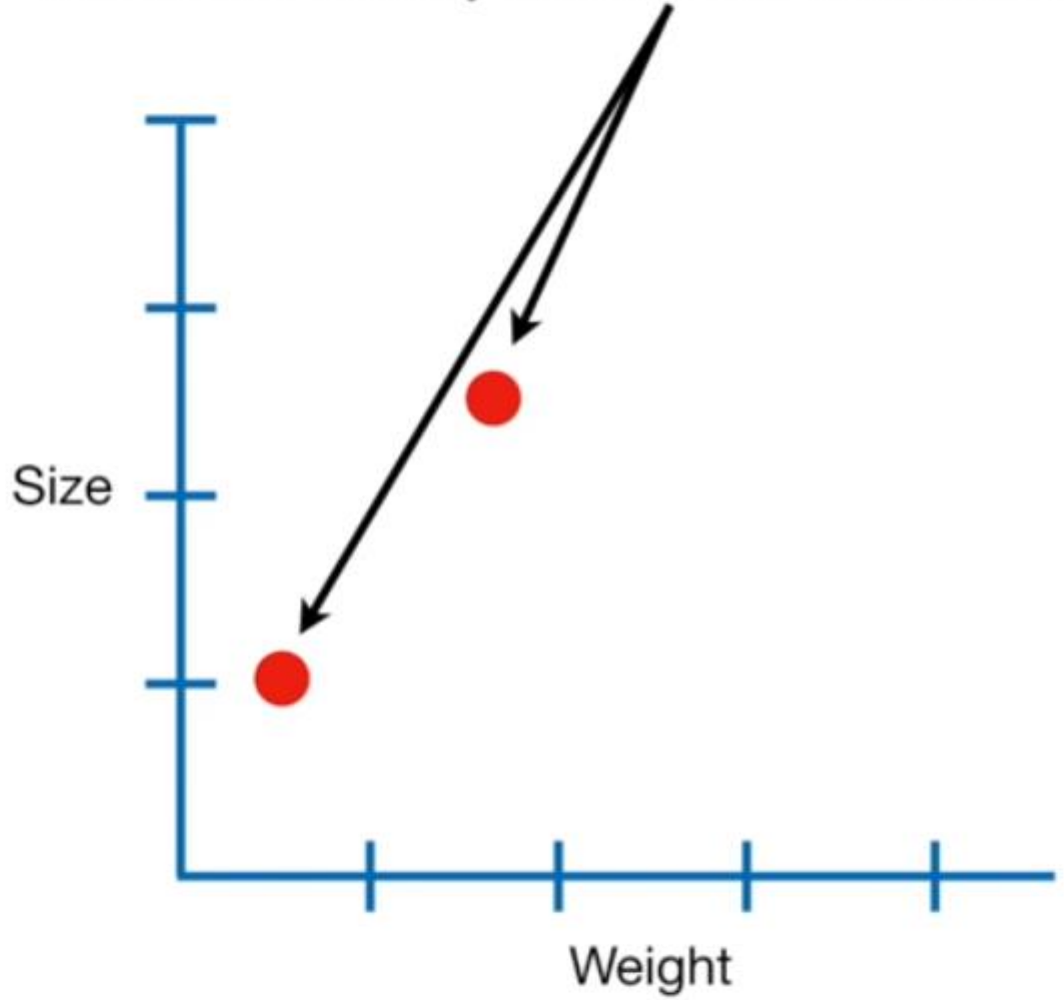
Together, the value for **Weight**, **2.5**, and the value for **Size**, **2.8**, give us a point on the line.



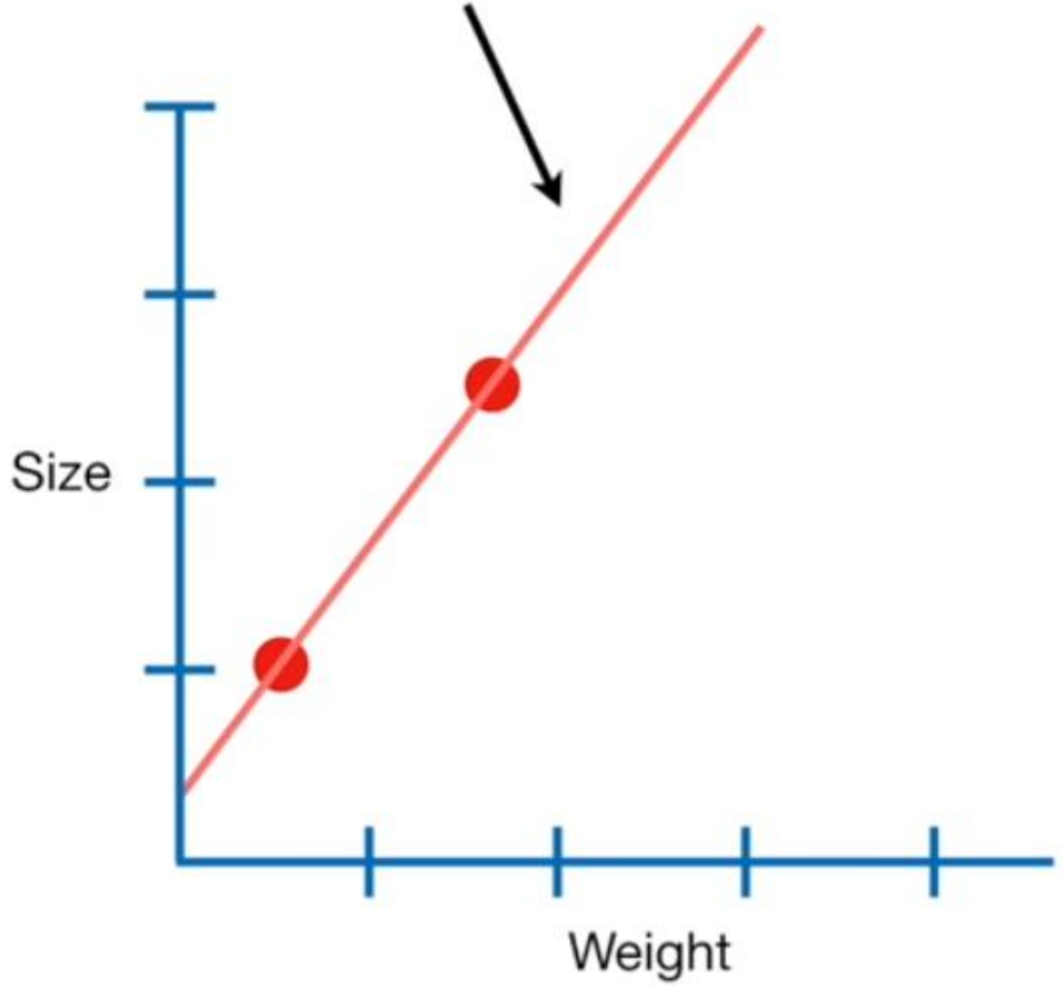
When we have a lot of measurements, we can be fairly confident that the **Least Squares** line accurately reflects the relationship between **Size** and **Weight**.

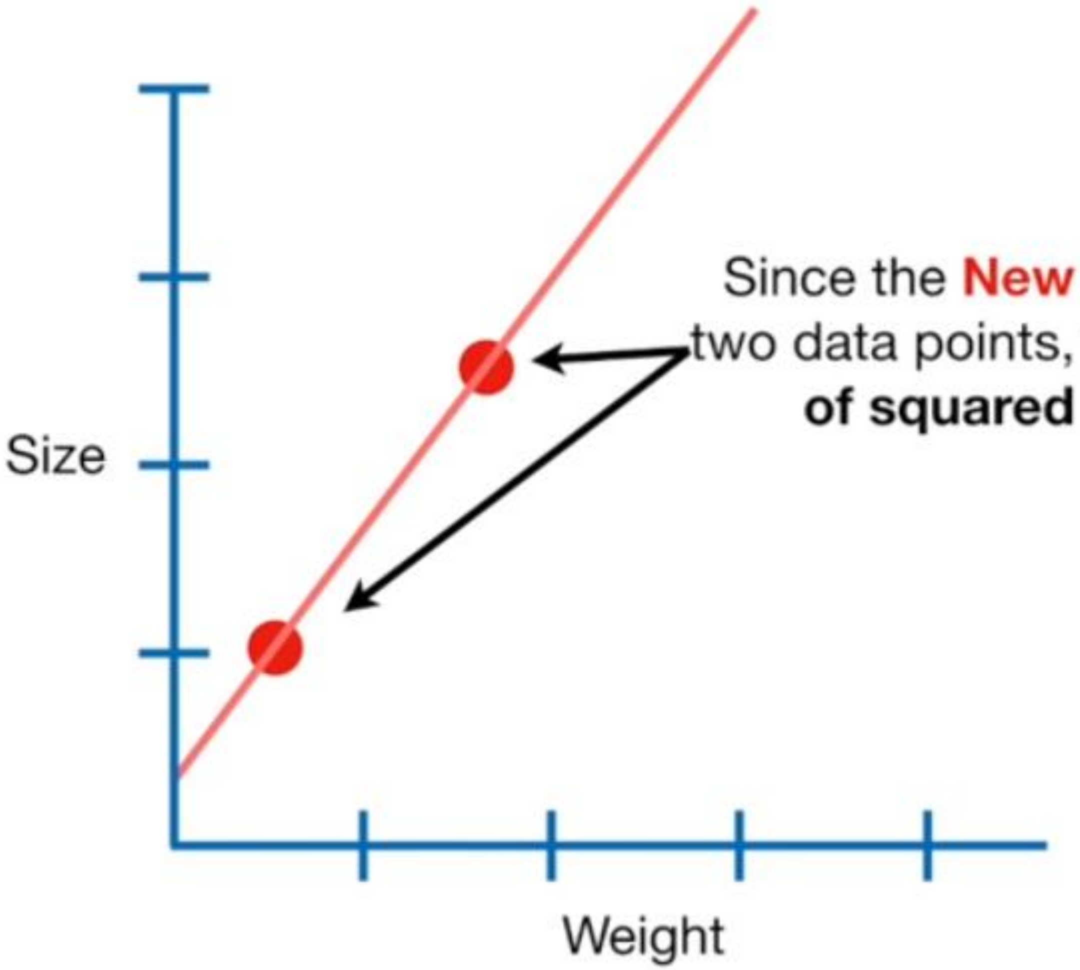


But what if we only have two measurements?

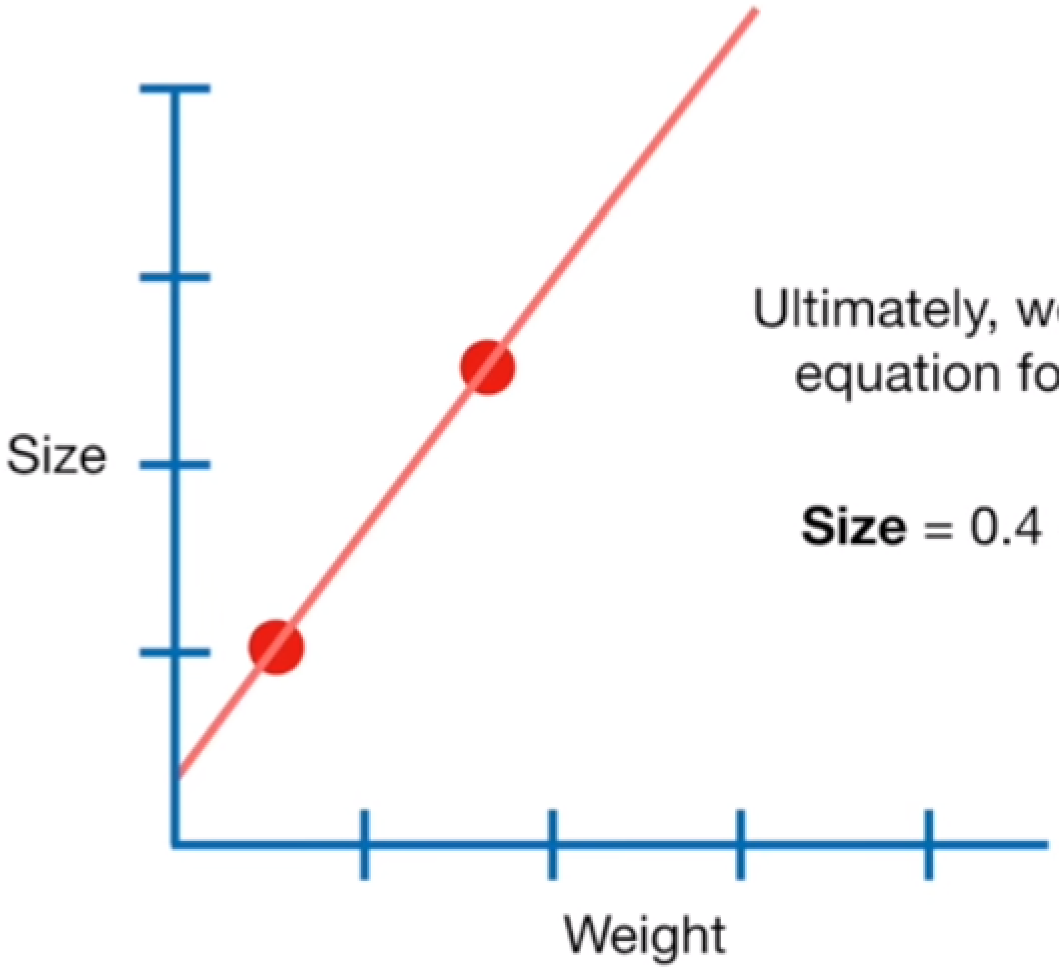


We fit a **New Line** with **Least Squares**...



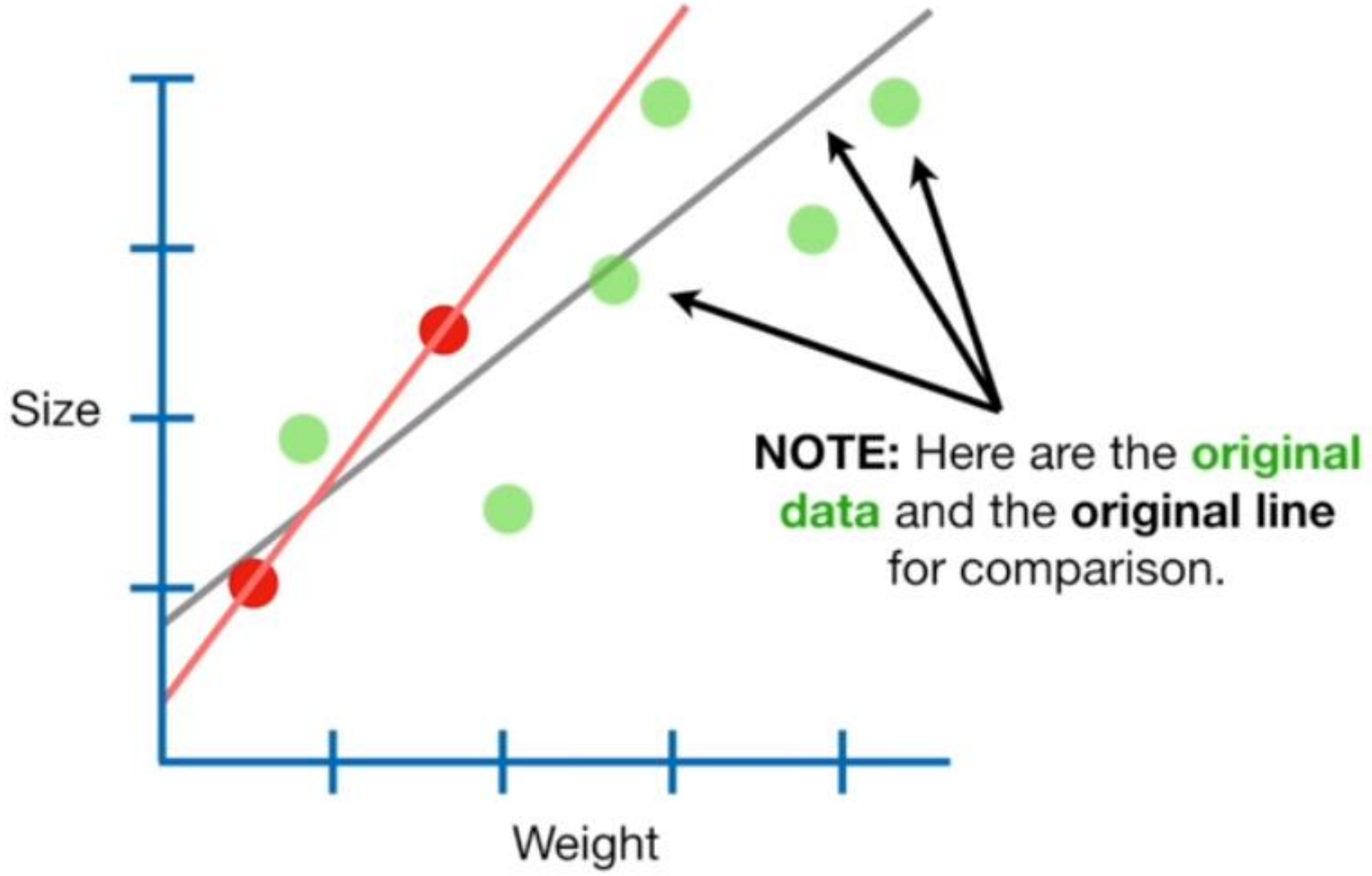


Since the New Line overlaps the two data points, the **minimum sum of squared residuals = 0**.

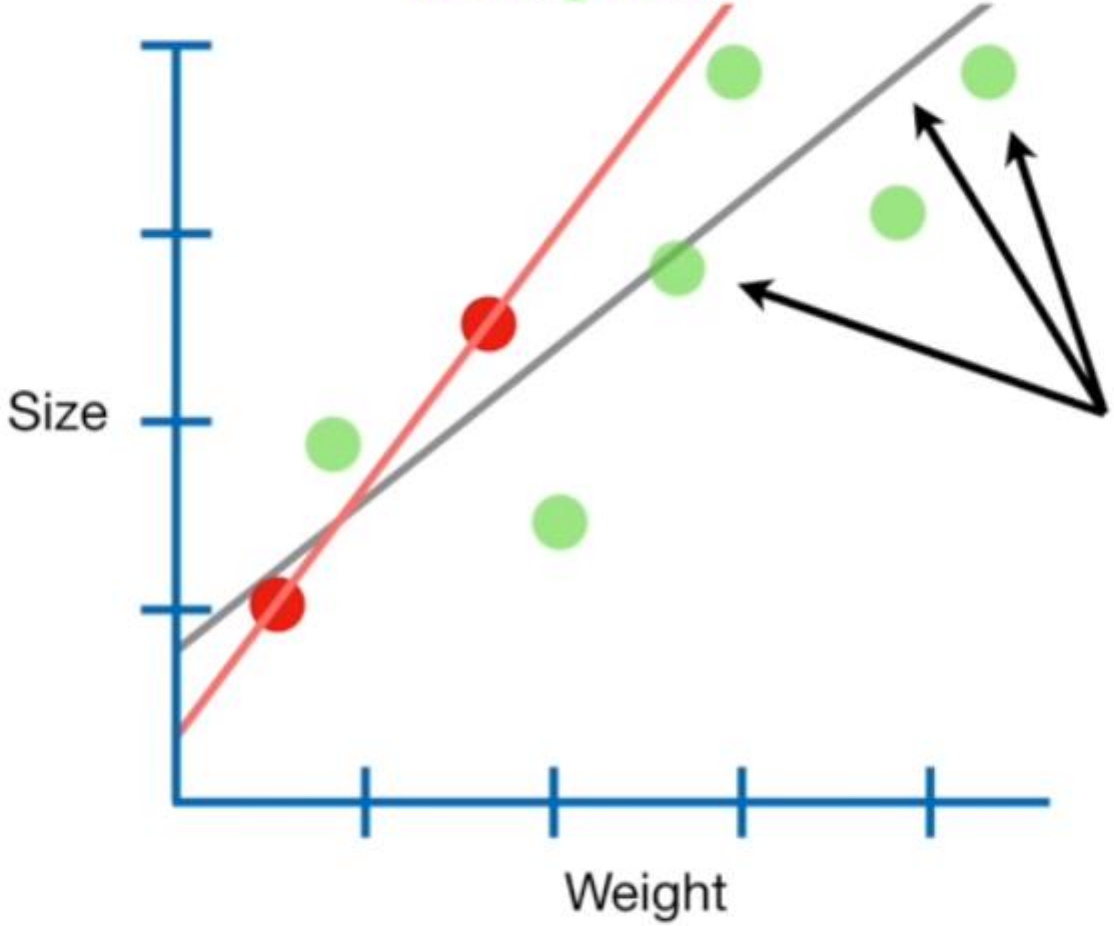


Ultimately, we end up with this equation for the **New Line**:

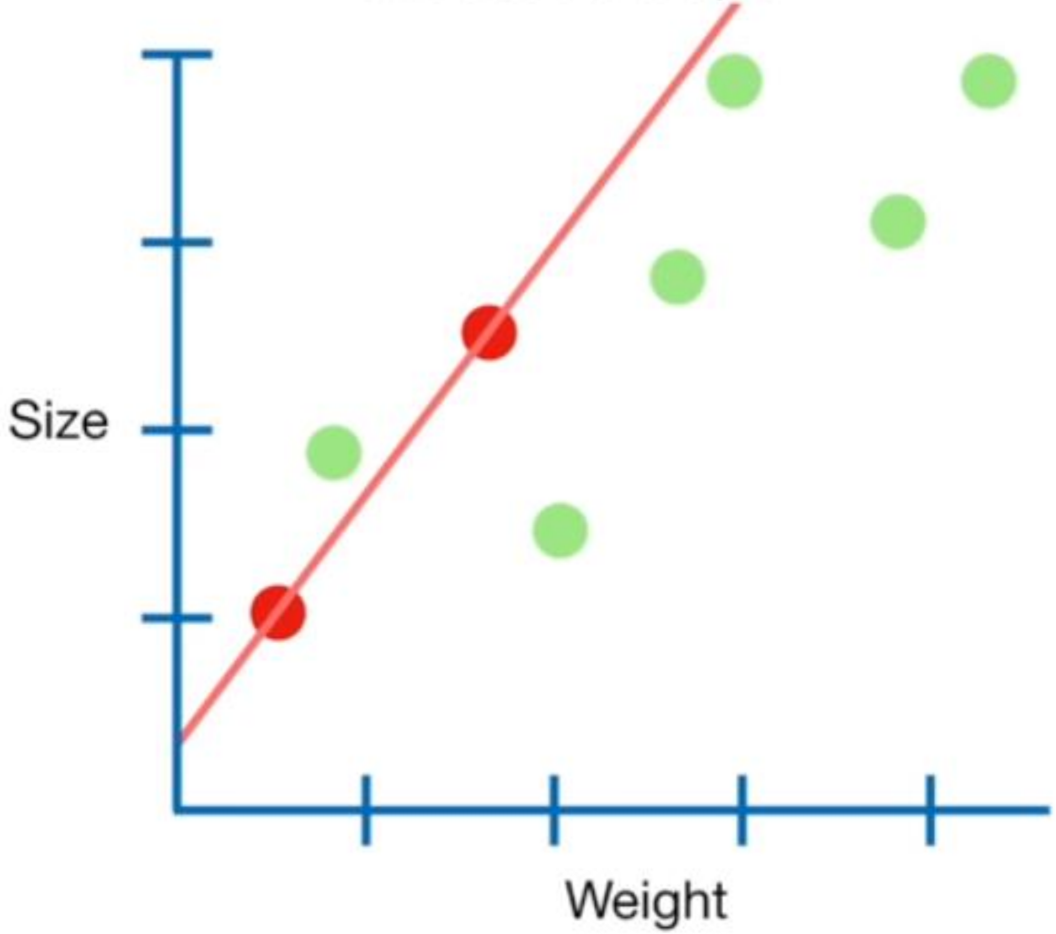
Size = 0.4 + 1.3 × **Weight**


Size
Weight
NOTE: Here are the **original data** and the **original line** for comparison.

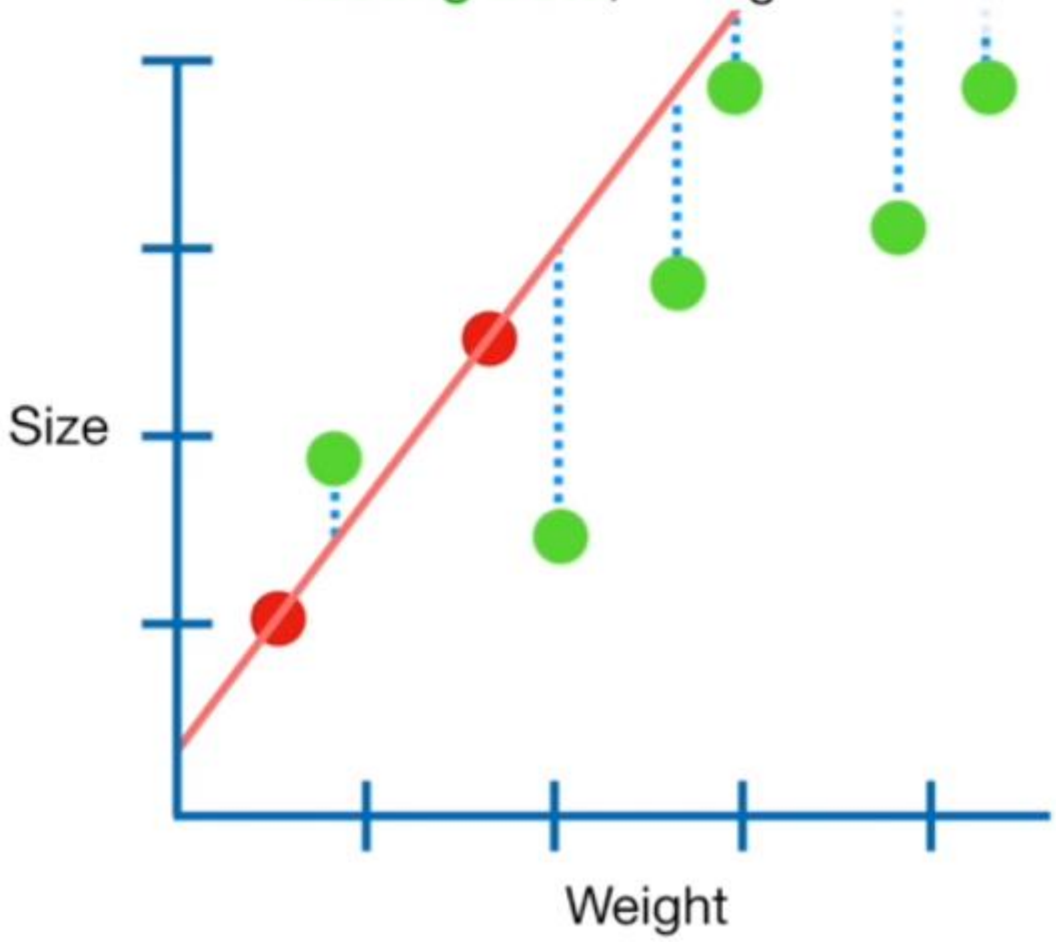
Let's call the **Two Red Dots** the **Training Data**, and the remaining **Green Dots** the **Testing Data**.



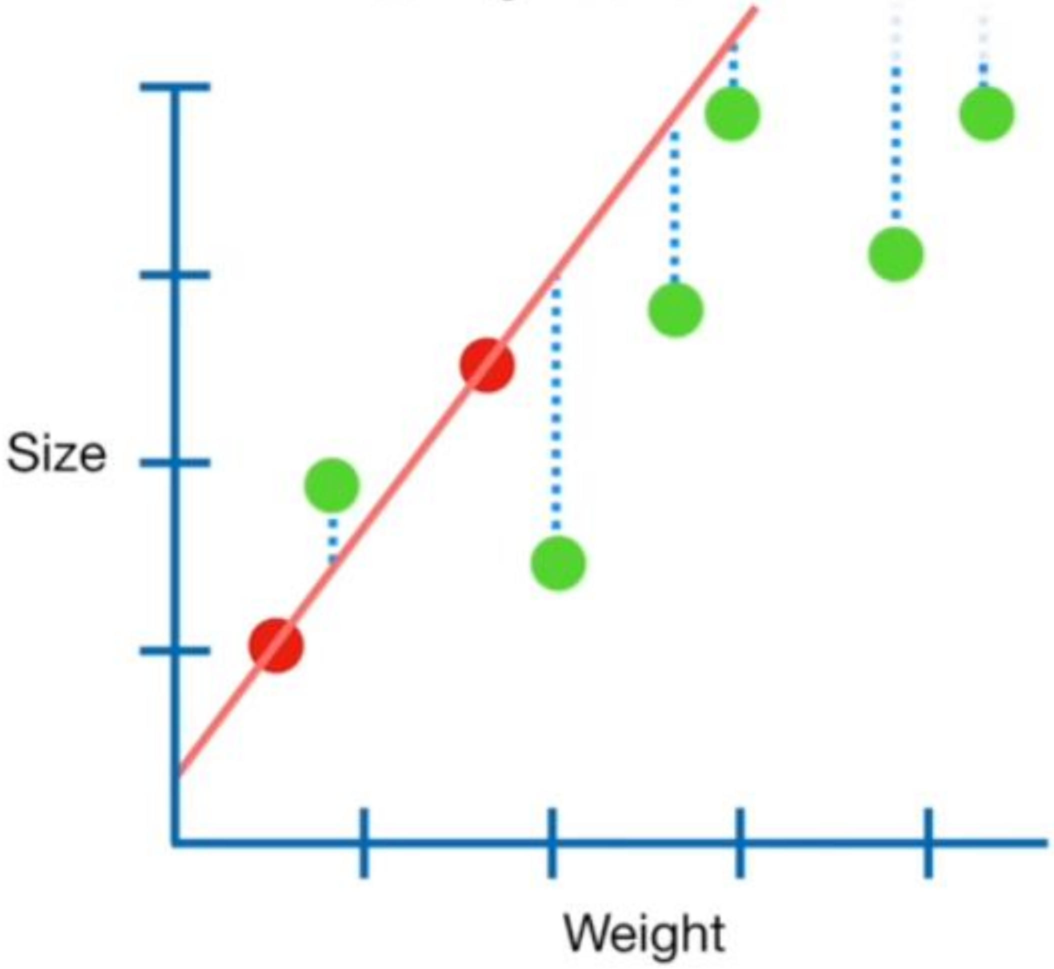
The sum of the squared residuals for just the **Two Red Points**, the **Training Data**, is small (in this case it is 0)…



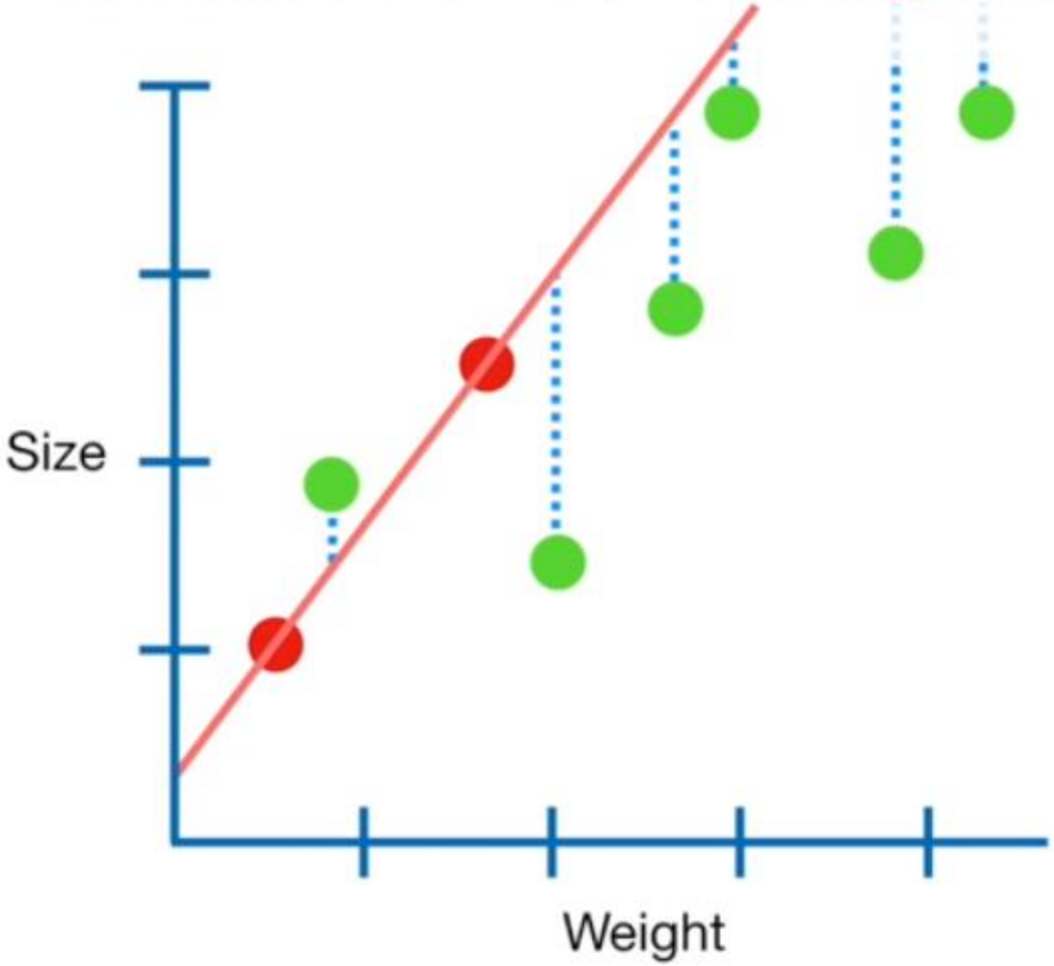
...but the sum of the squared residuals for the **Green Points**, the **Testing Data**, is large...



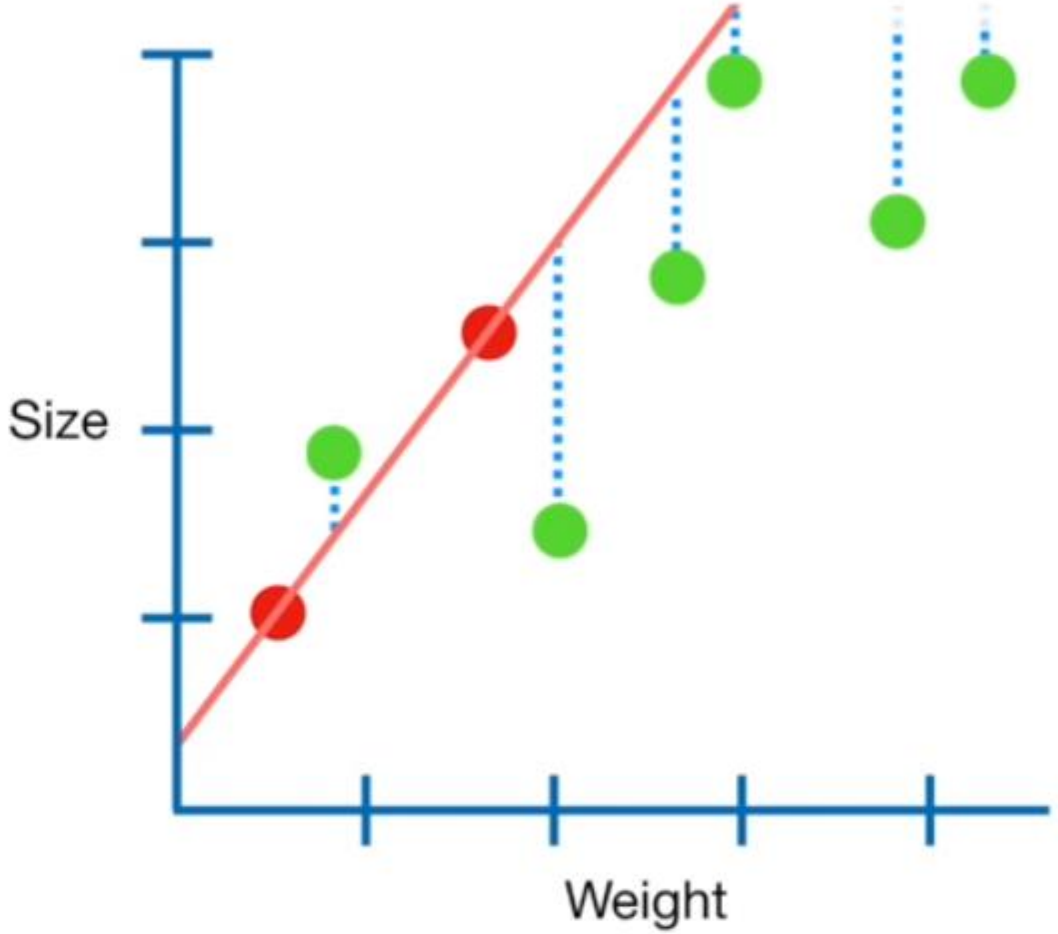
…and that means that the **New Line** has **High Variance**.

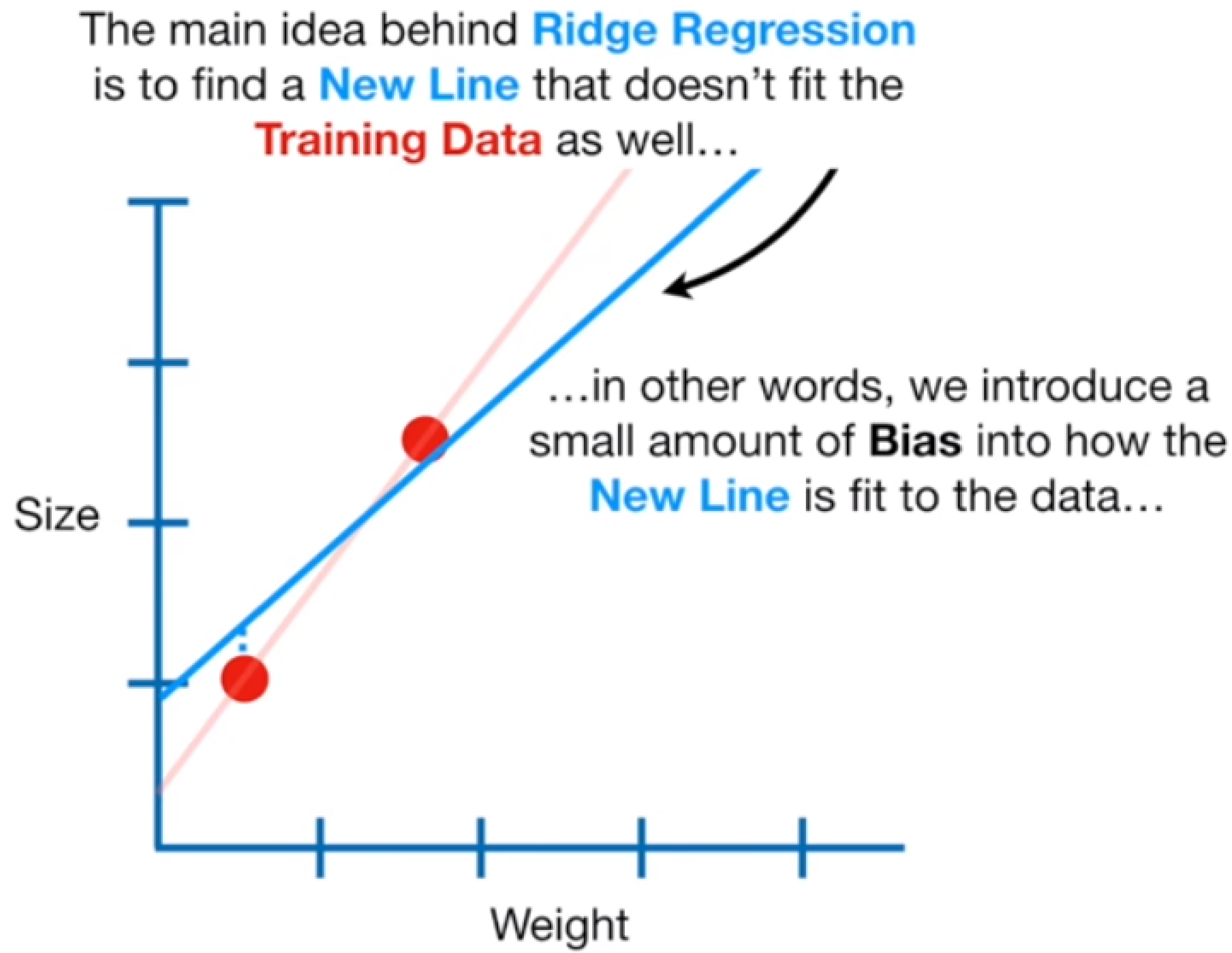


In machine learning lingo, we'd say that the **New Line** is **Over Fit** to the **Training Data**.

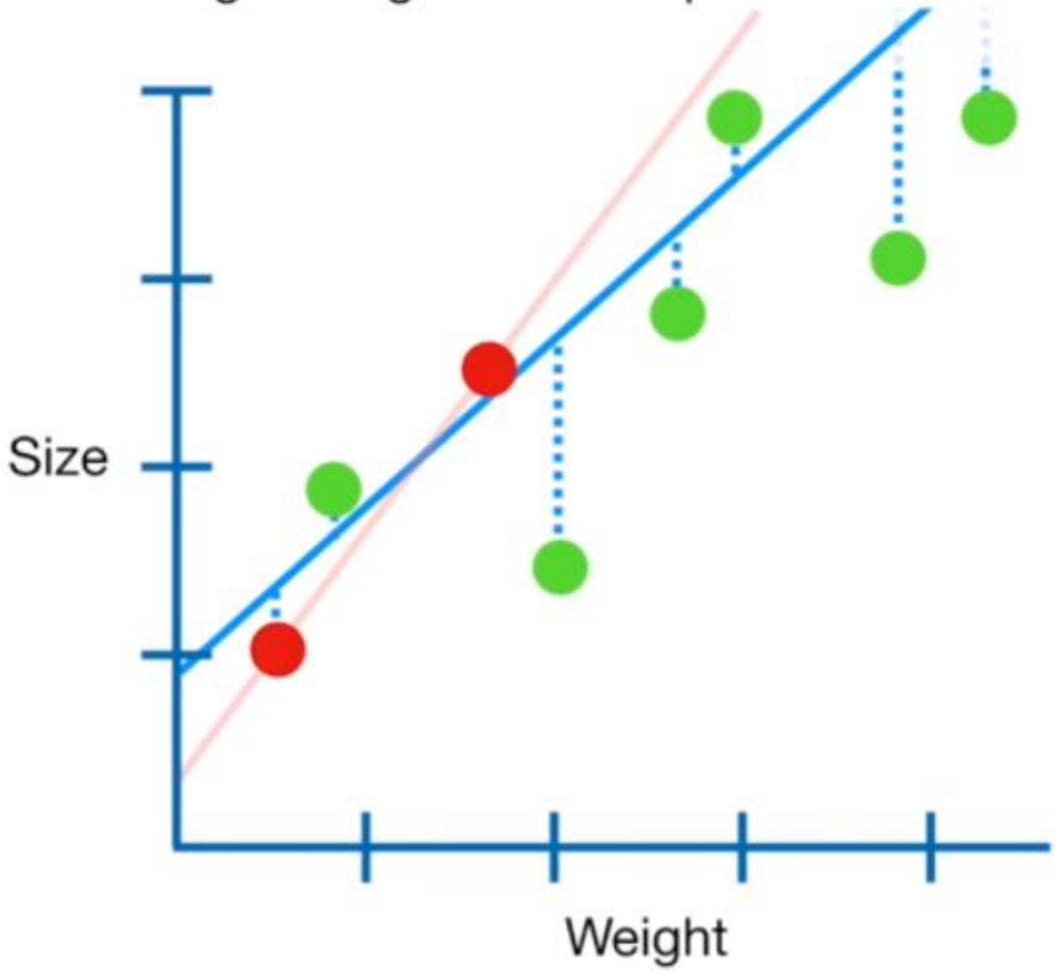


We just saw that **Least Squares** results in a **Line** that is **Over Fit** and has **High Variance**…

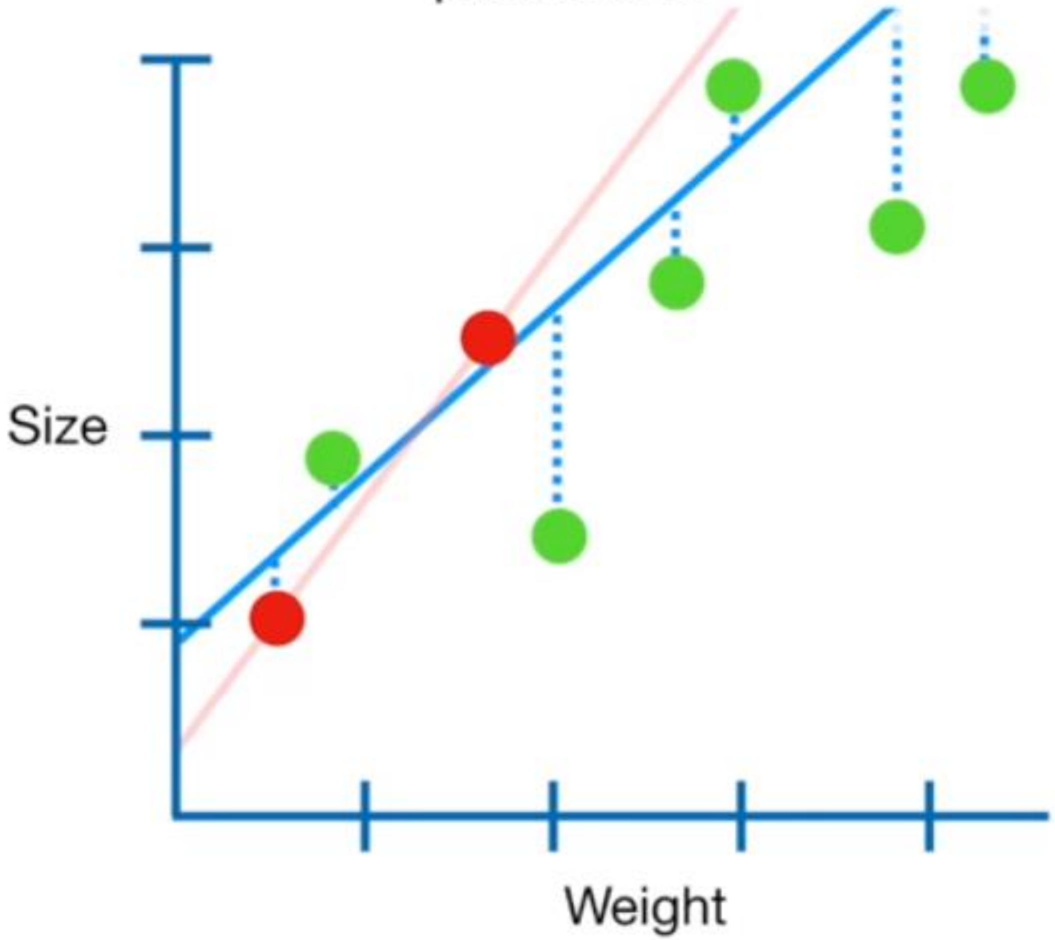


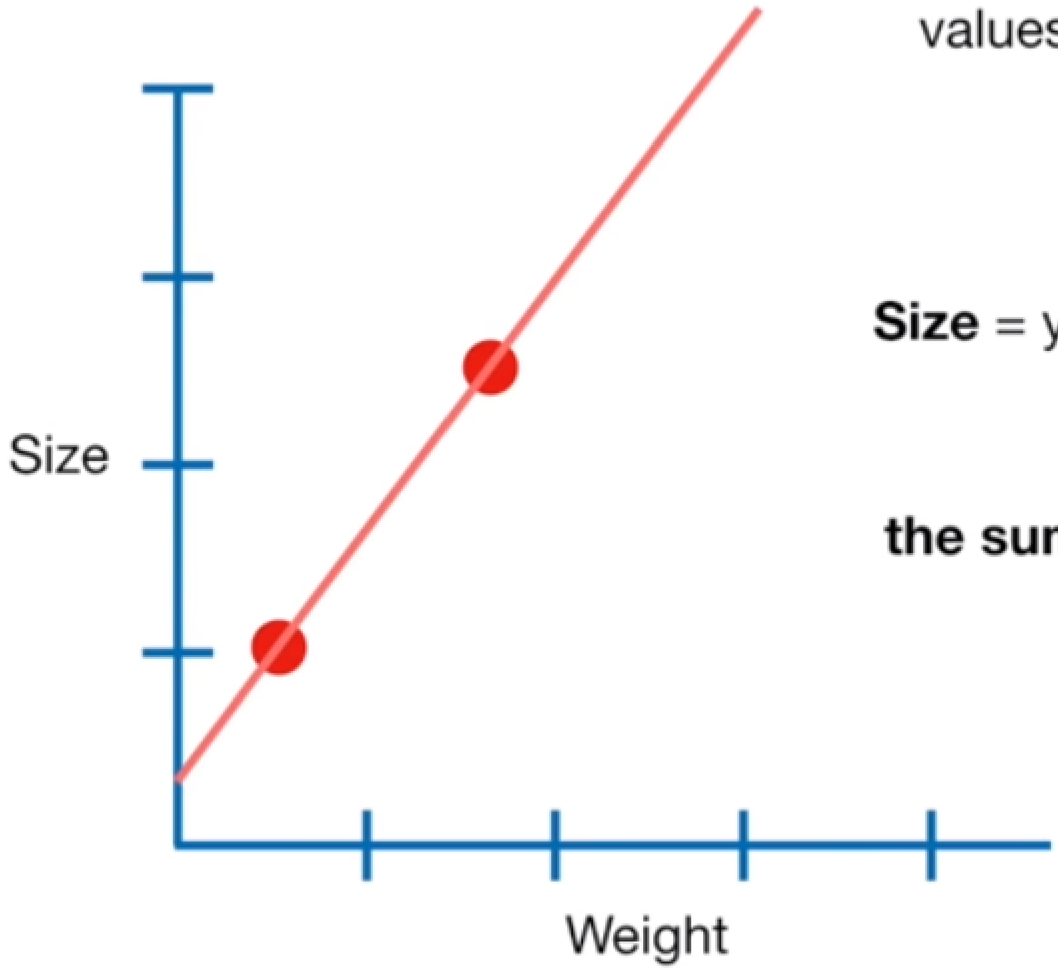

The main idea behind Ridge Regression is to find a New Line that doesn't fit the Training Data as well…
…in other words, we introduce a small amount of Bias into how the New Line is fit to the data…
Size
Weight

…but in return for that small amount of **Bias**, we get a significant drop in **Variance**.



In other words, by starting with a slightly worse fit, **Ridge Regression** can provide better long term predictions.





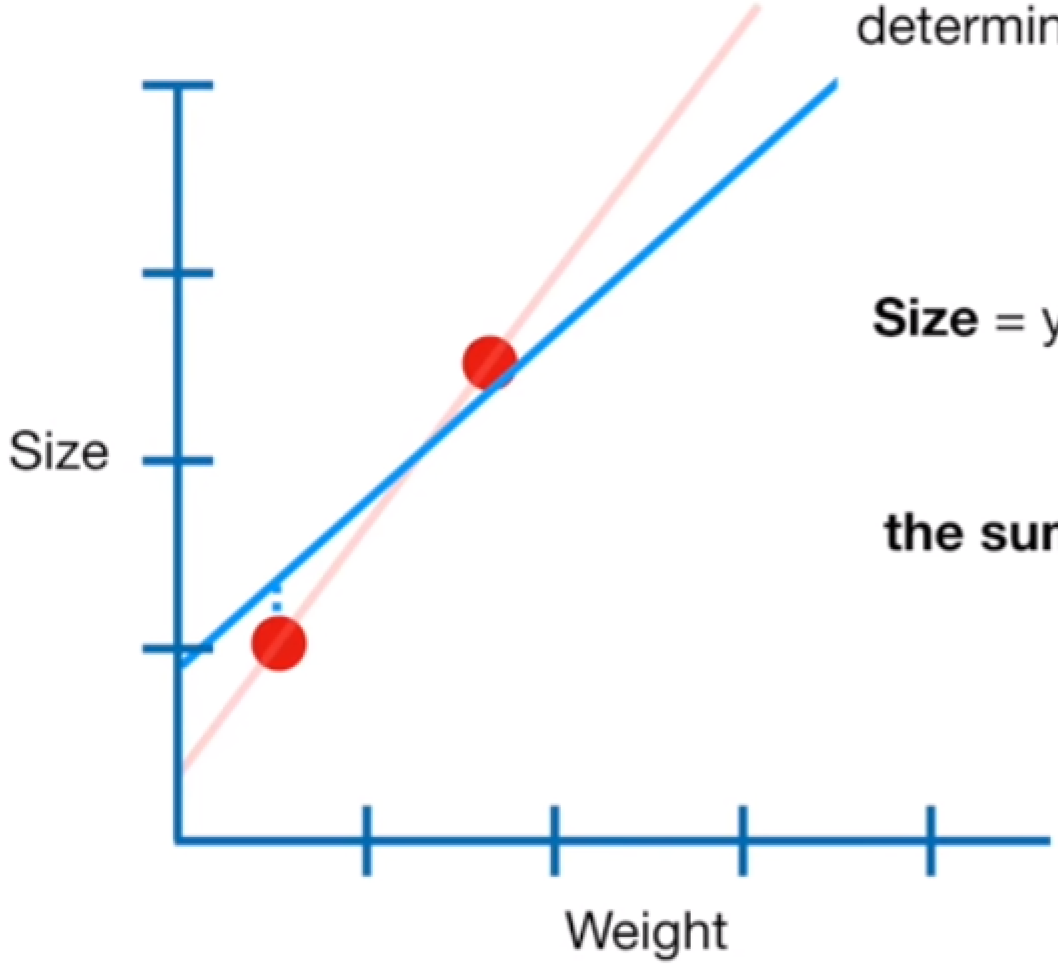
When **Least Squares** determines values for the parameters in this equation…

↓

Size = y-axis intercept + slope × **Weight**

…it minimizes…

the sum of the squared residuals



In contrast, when **Ridge Regression** determines values for the parameters in this equation…

↓

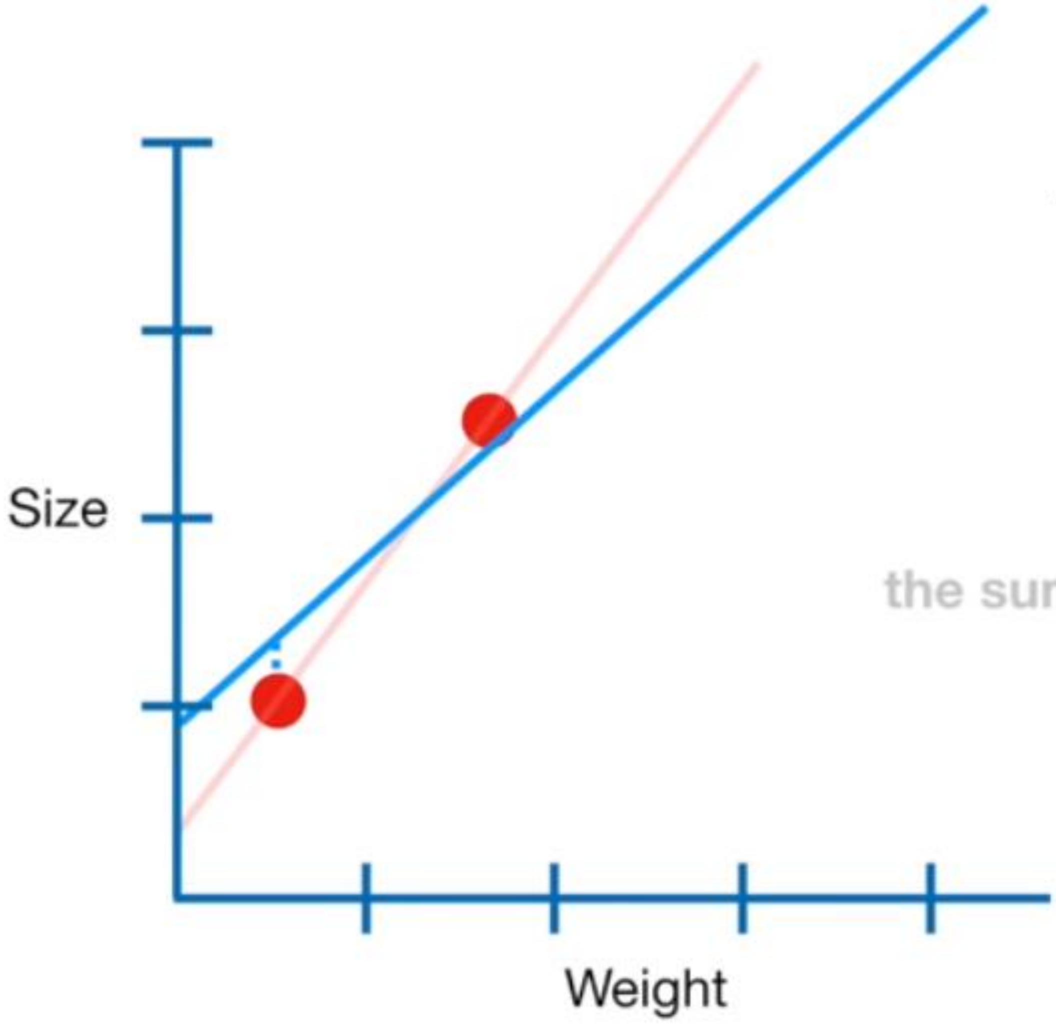
Size = y-axis intercept + slope × **Weight**

…it minimizes…

the sum of the squared residuals

+

λ × the slope²

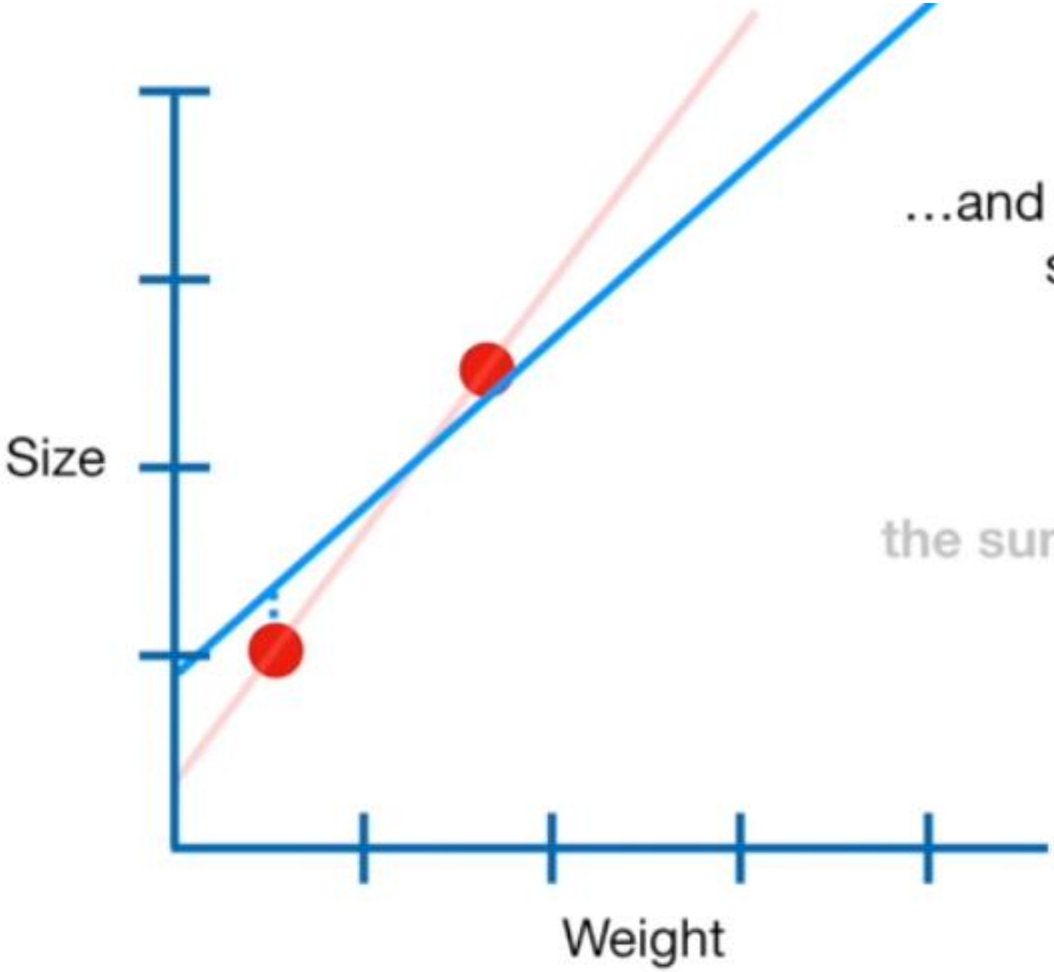


This part of the equation adds a penalty to the traditional **Least Squares** method…

the sum of the squared residuals

+

$\lambda \times \boxed{\textbf{the slope}^2}$

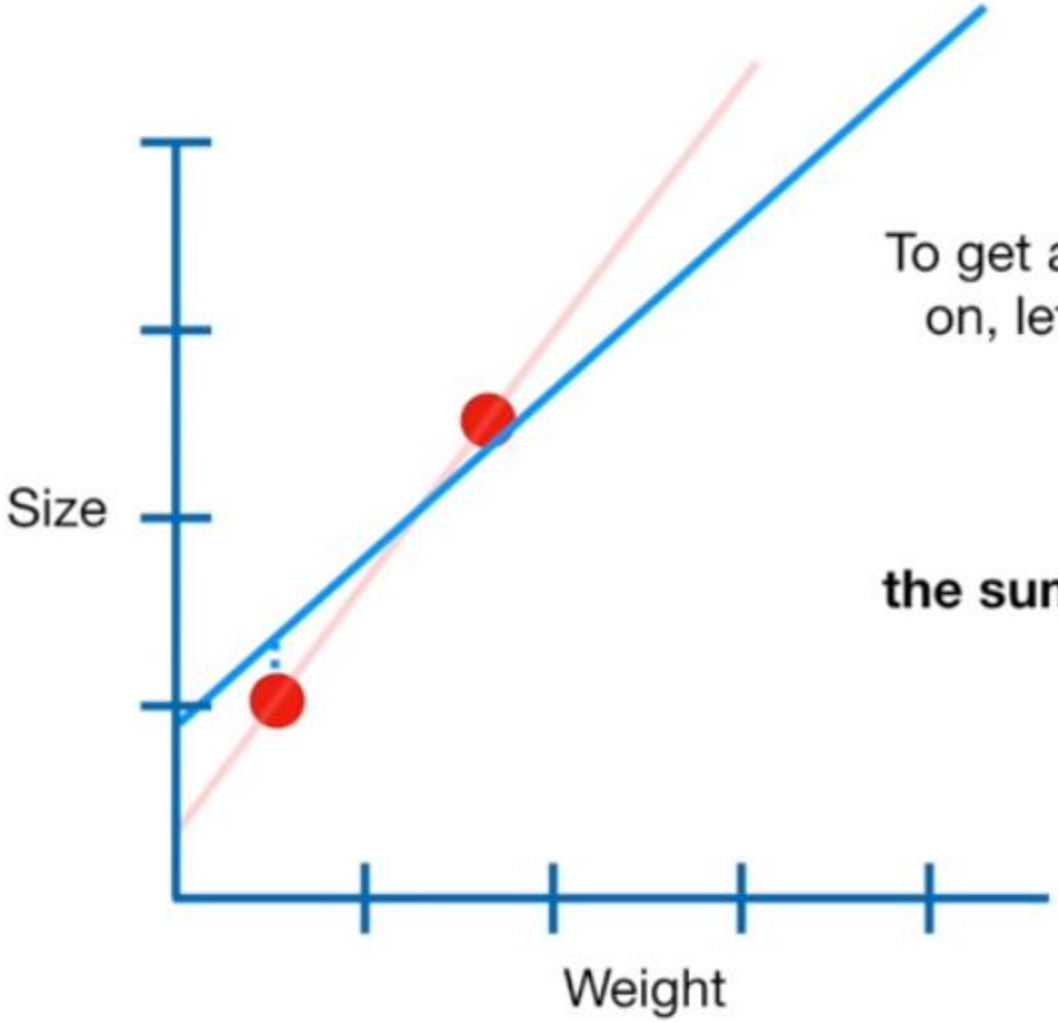


…and **λ** (lambda) determines how severe that penalty is.

the sum of the squared residuals

+

$\boxed{\lambda} \times$ **the slope^2**

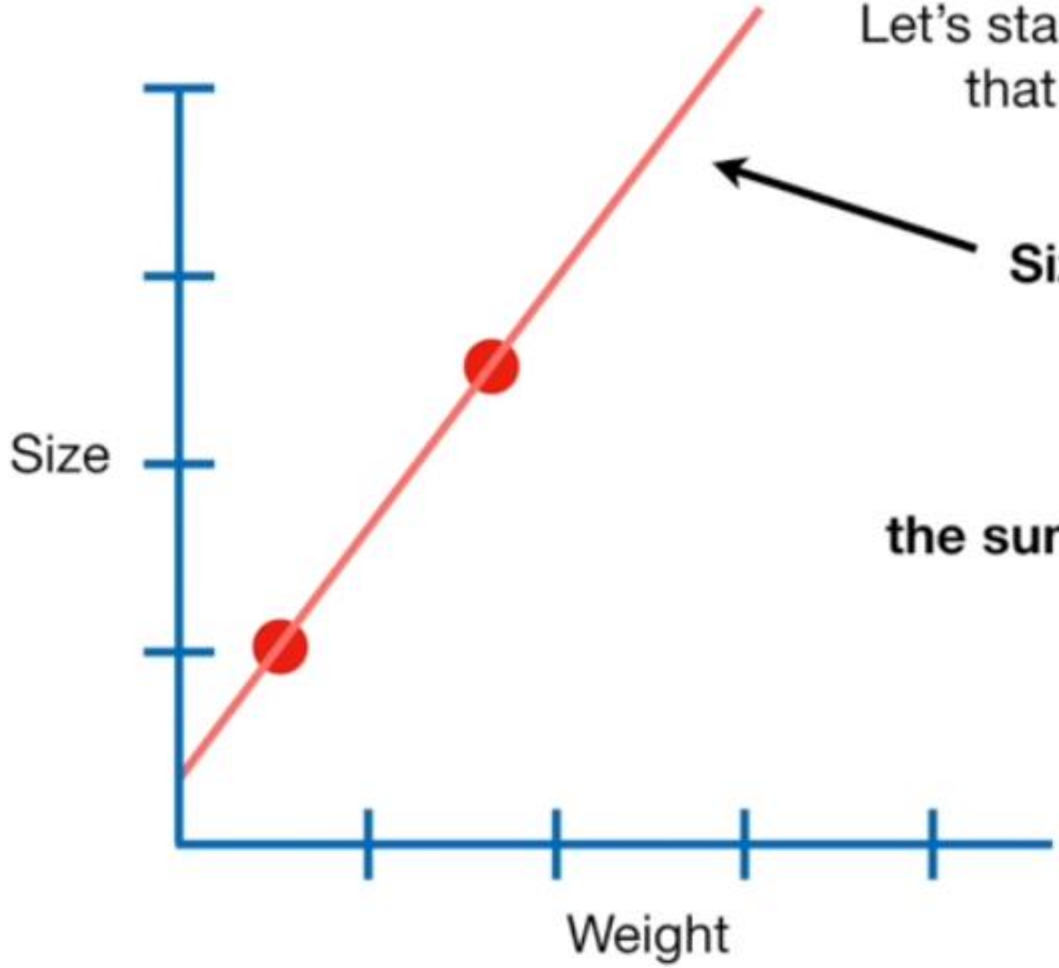


To get a better idea of what's going on, lets plug in some numbers!!!

the sum of the squared residuals

+

λ × the slope^2



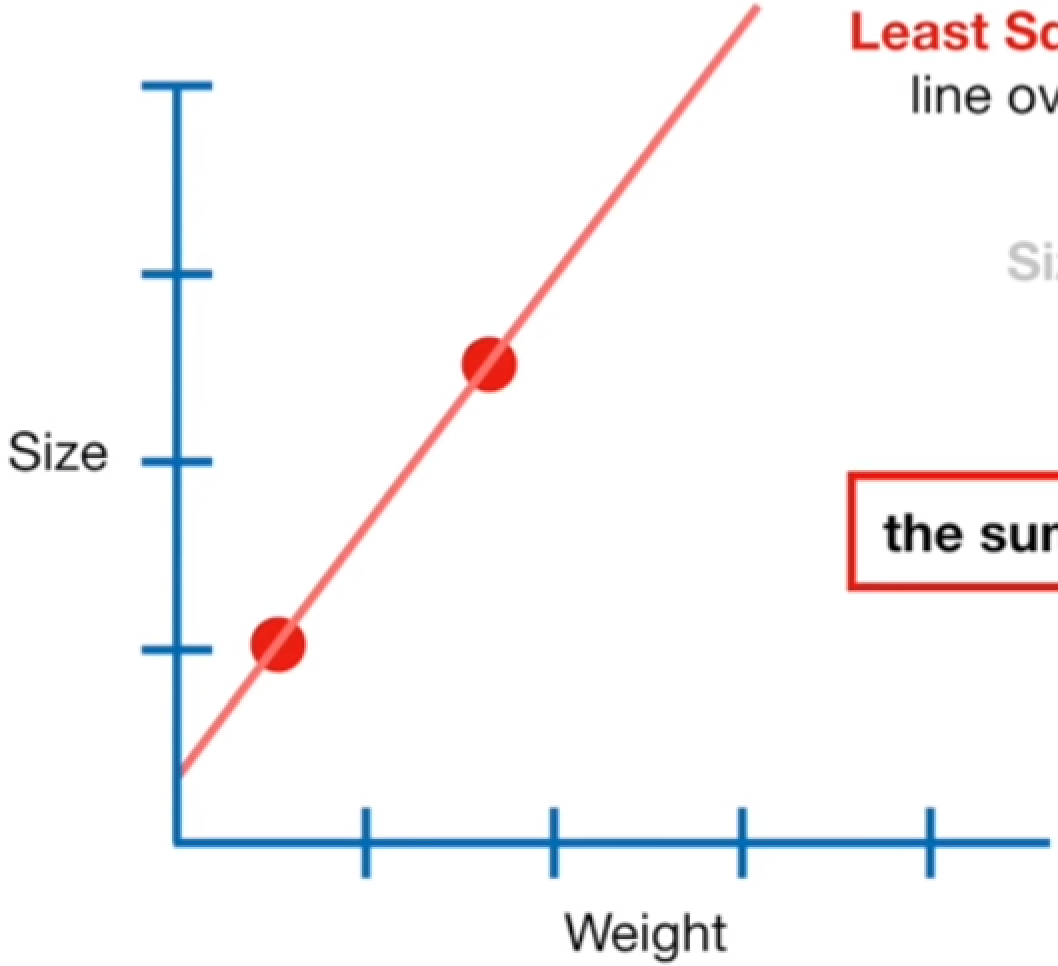
Let's start by plugging in the numbers that correspond to the **Least Squares Fit**.

Size = 0.4 + 1.3 × **Weight**

the sum of the squared residuals

+

λ × the slope²



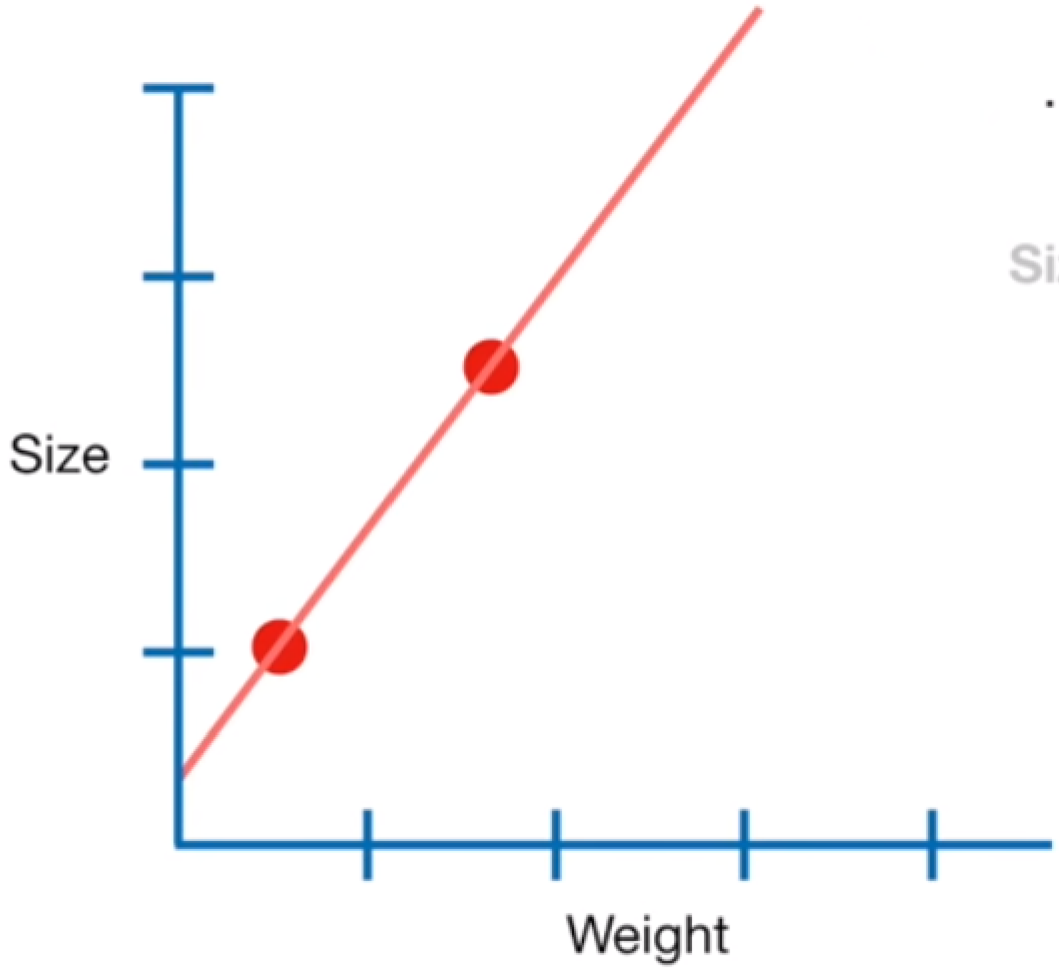
The sum of squared residuals for the **Least Squares Fit** is **0** (because the line overlaps the data points)…

Size = 0.4 + 1.3 × **Weight**

the sum of the squared residuals

+

λ × **the slope**^{2}

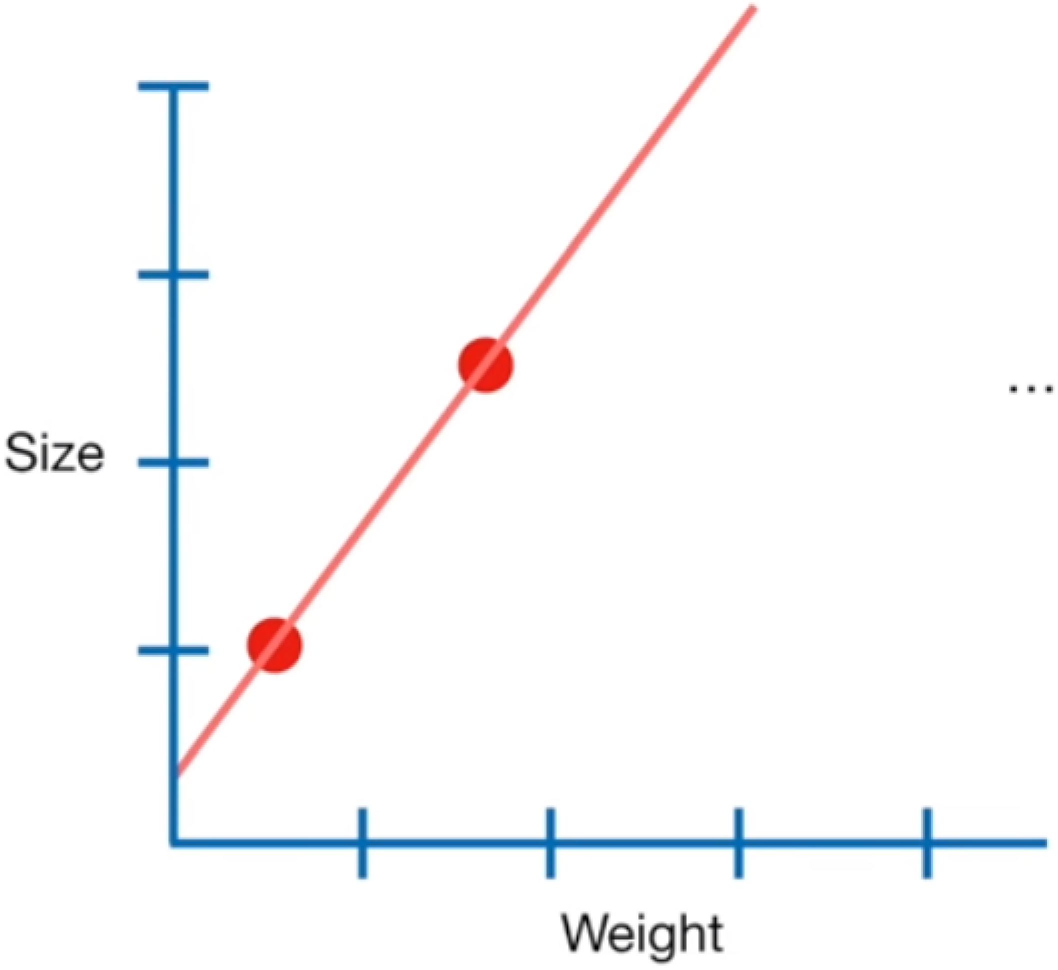


…and the slope is **1.3**.

$$\text{Size} = 0.4 + \boxed{1.3} \times \text{Weight}$$

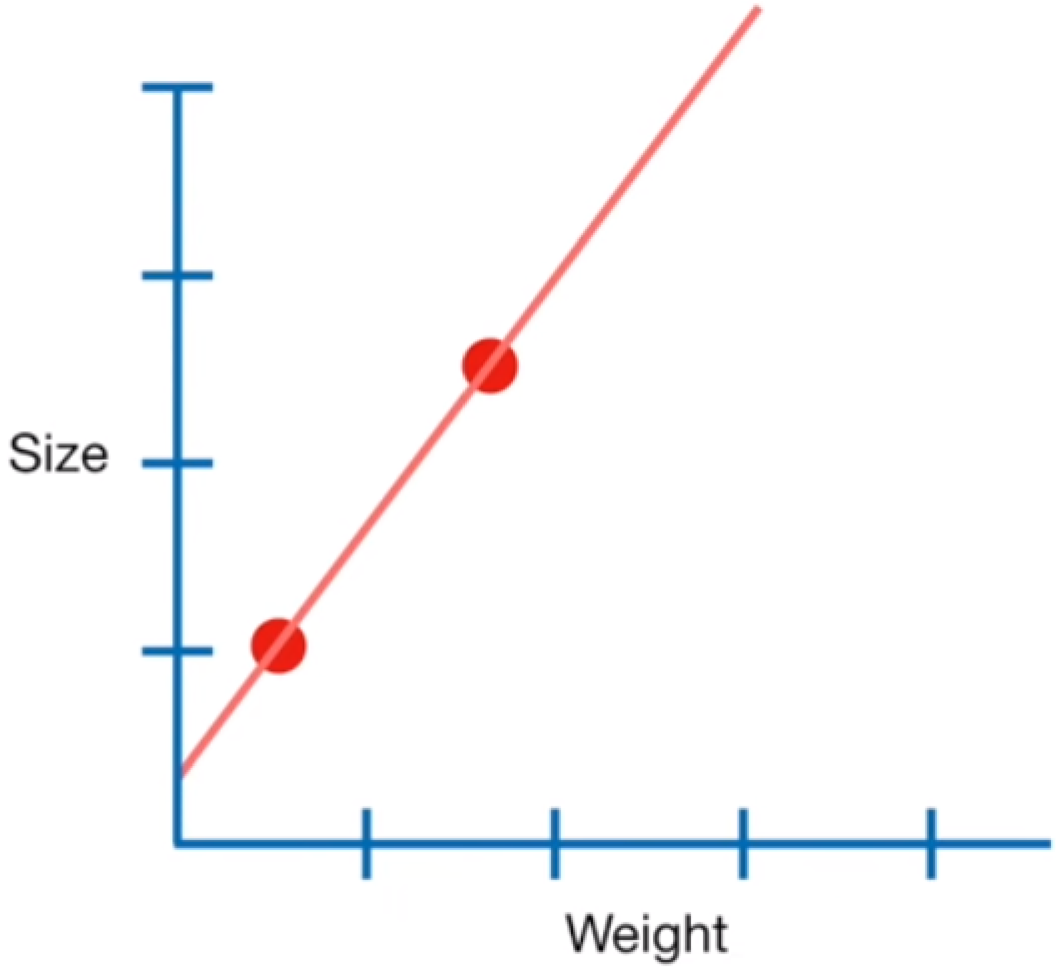
$$0 + \lambda \times \boxed{1.3^2}$$

Lambda =1



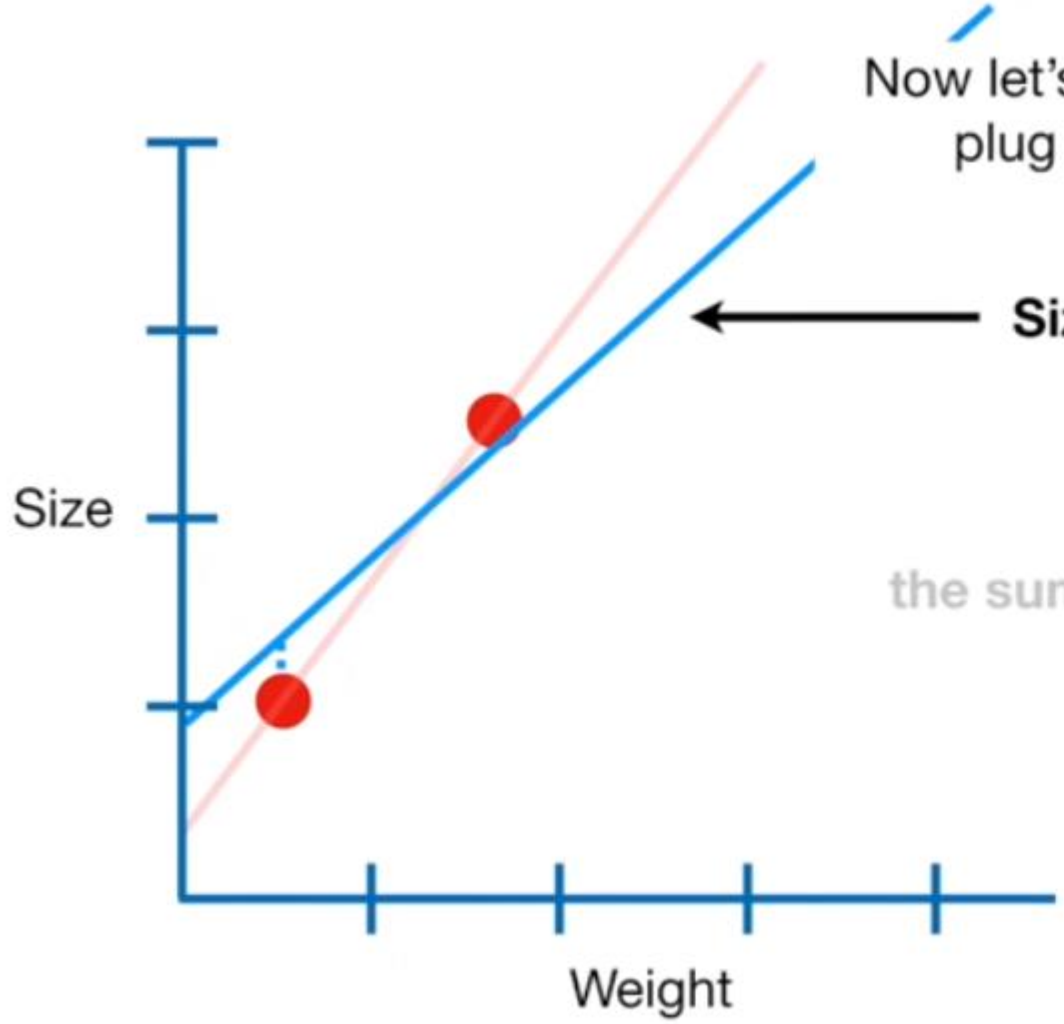
…all together, we have…

$$\boxed{\begin{array}{c} 0 \\ + \\ 1 \times 1.3^2 \end{array}}$$



…and when we do the math we get…

$$0 + 1 \times 1.3^2 = 0 + 1.69 \boxed{= 1.69}$$



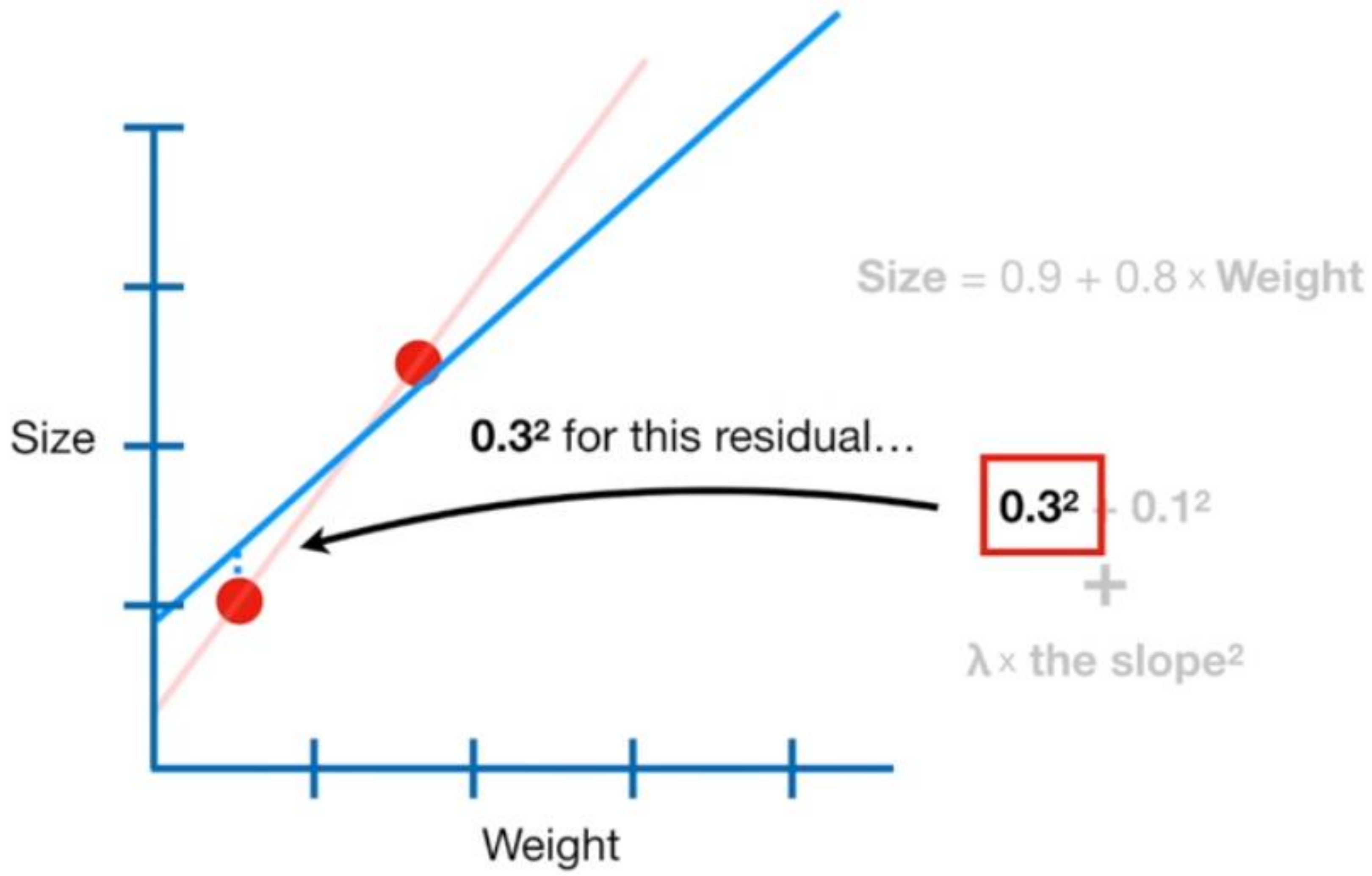
Now let's see what happens when we plug in numbers for the **Ridge Regression** line...

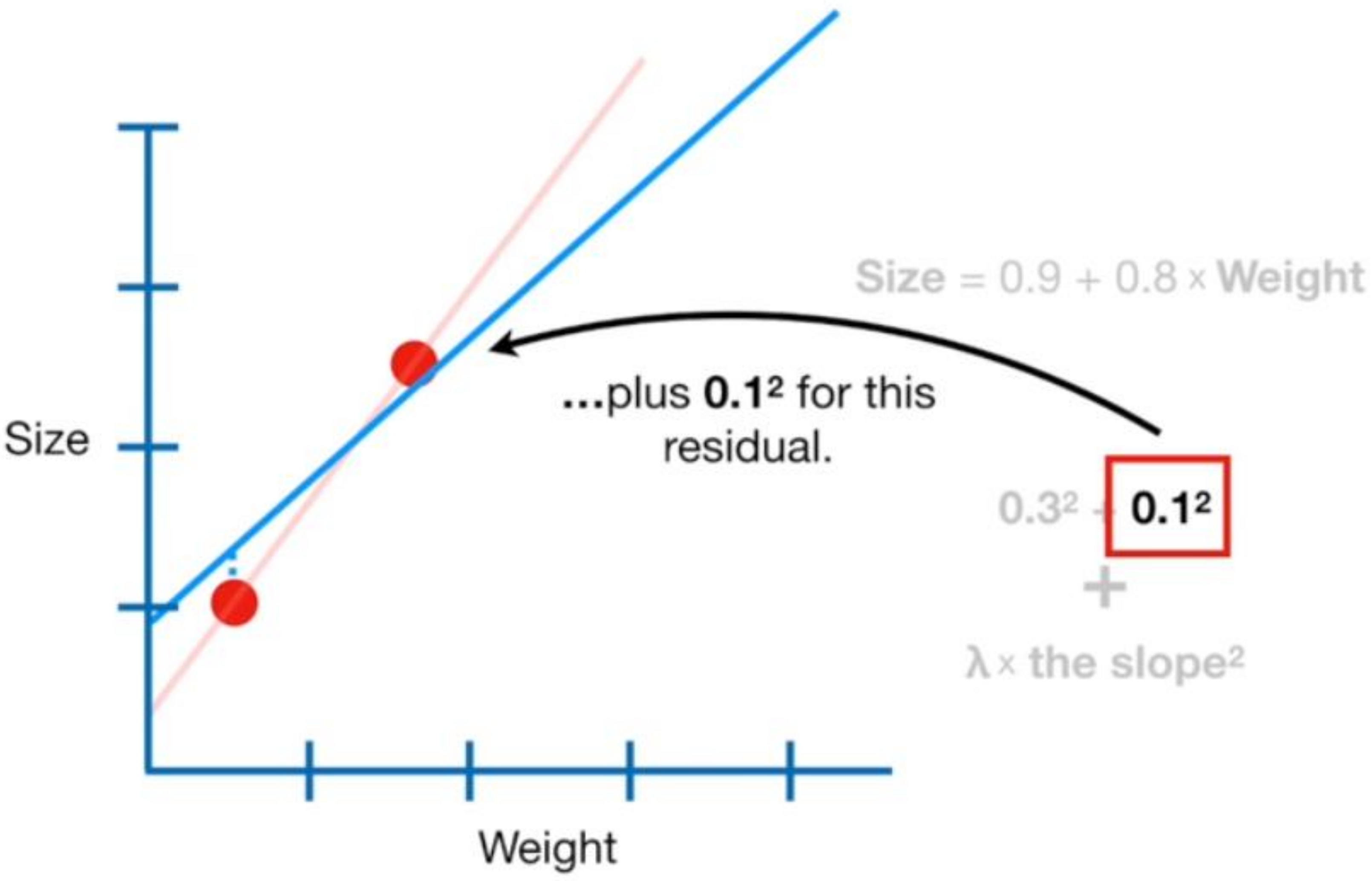
Size = 0.9 + 0.8 × **Weight**

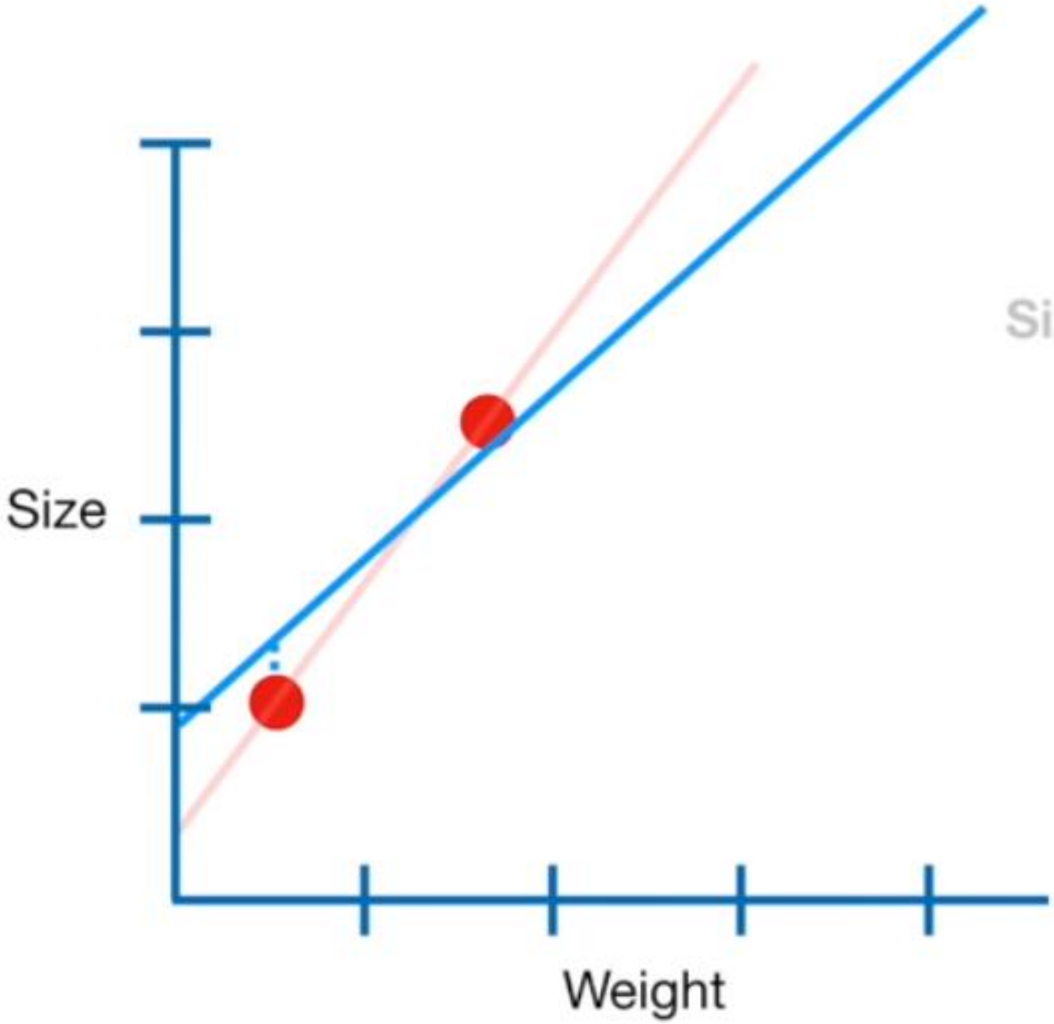
the sum of the squared residuals

+

λ × the slope²


Size = 0.9 + 0.8 × Weight
0.3² for this residual…
0.3² + 0.1²
+
λ × the slope²
Size
Weight

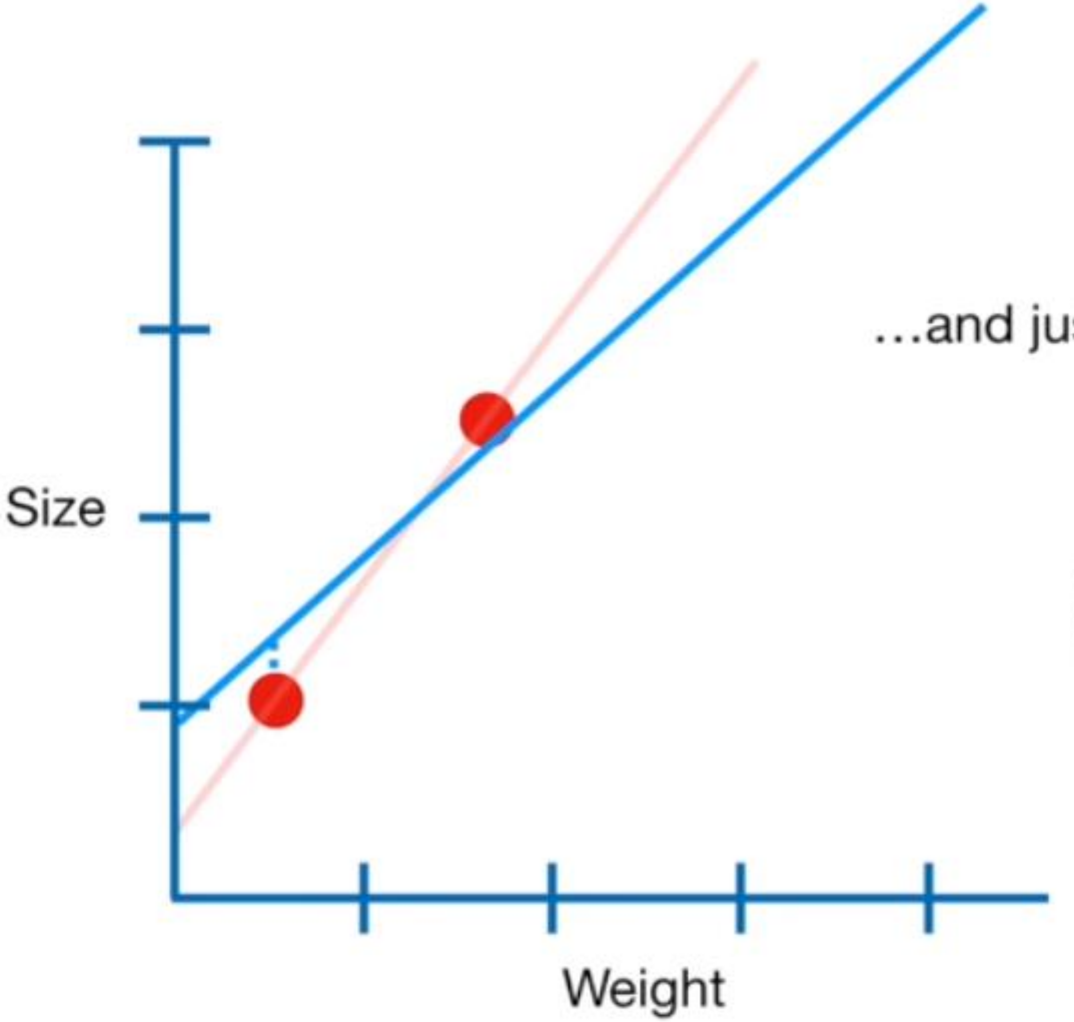

Size
Weight
...plus 0.1² for this residual.
Size = 0.9 + 0.8 x Weight
0.3² + 0.1²
+
λ x the slope²



The slope is **0.8**…

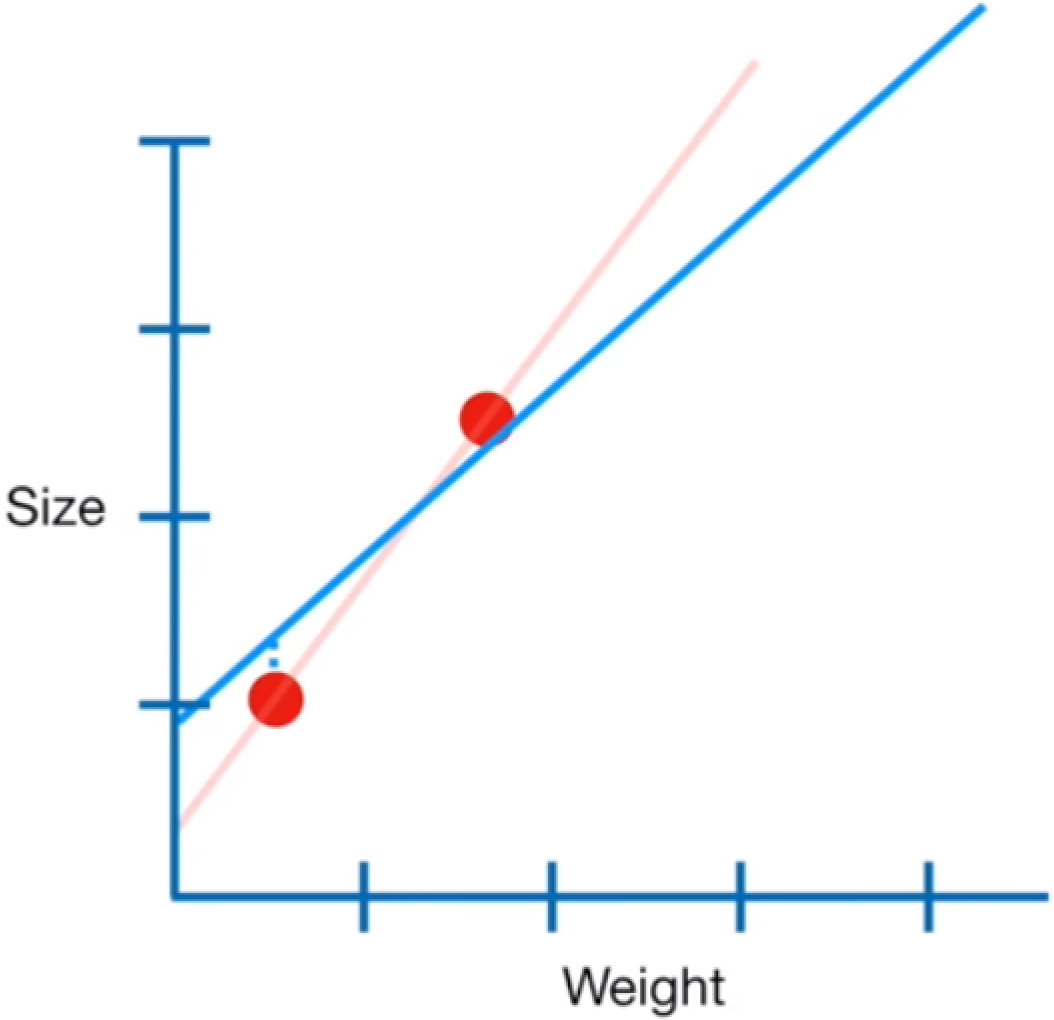
$$\text{Size} = 0.9 + \boxed{0.8} \times \text{Weight}$$

$$\mathbf{0.3^2 + 0.1^2}$$
$$\mathbf{+}$$
$$\lambda \times \boxed{\textbf{the slope}^2}$$



…and just like before, we'll let $\boldsymbol{\lambda}$ = **1**.

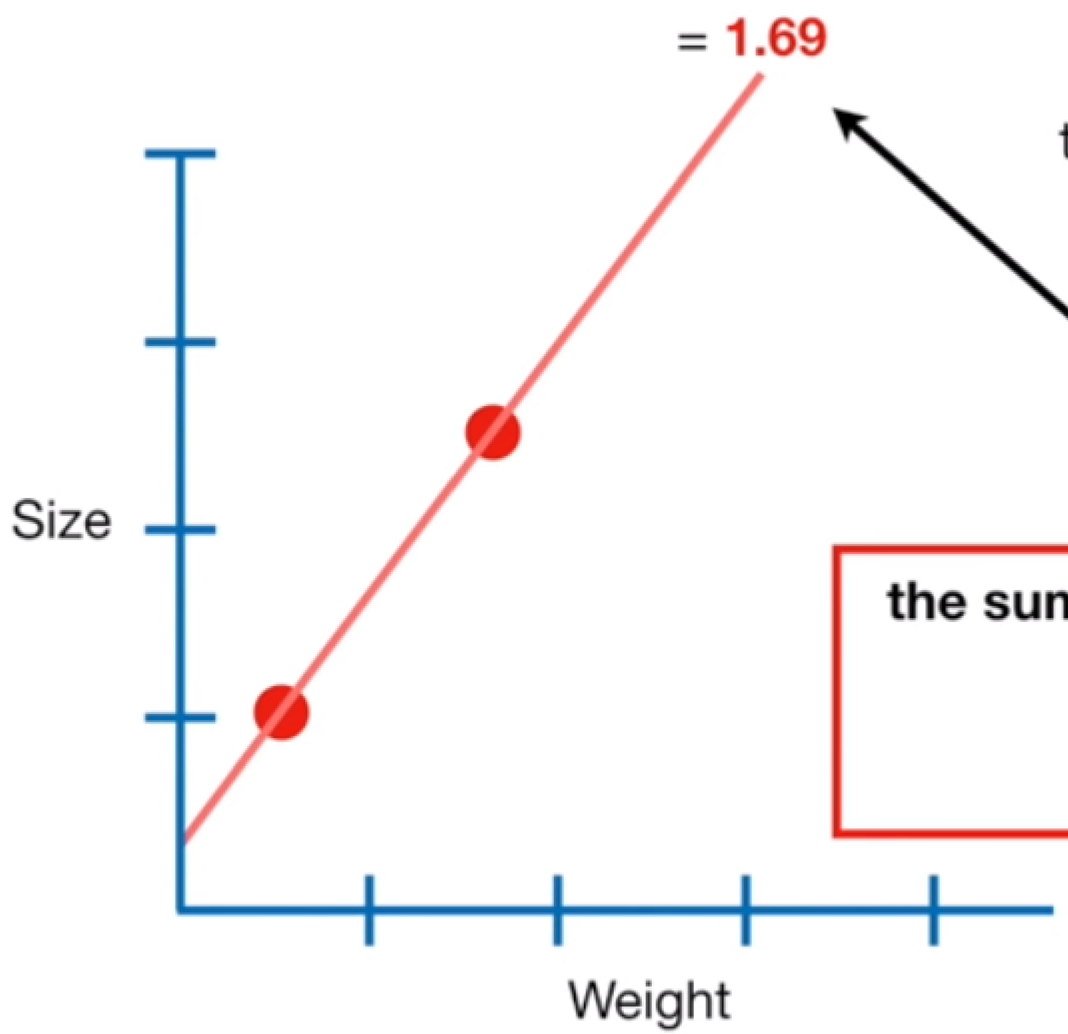
$$0.3^2 + 0.1^2$$
$$+$$
$$\boxed{1} \times 0.8^2$$



…and when we do the math we get…

$$\begin{aligned} &0.3^2 + 0.1^2 \\ &\quad + \\ &1 \times 0.8^2 \end{aligned} = 0.09 + 0.01 + 0.64$$

$$\boxed{= 0.74}$$

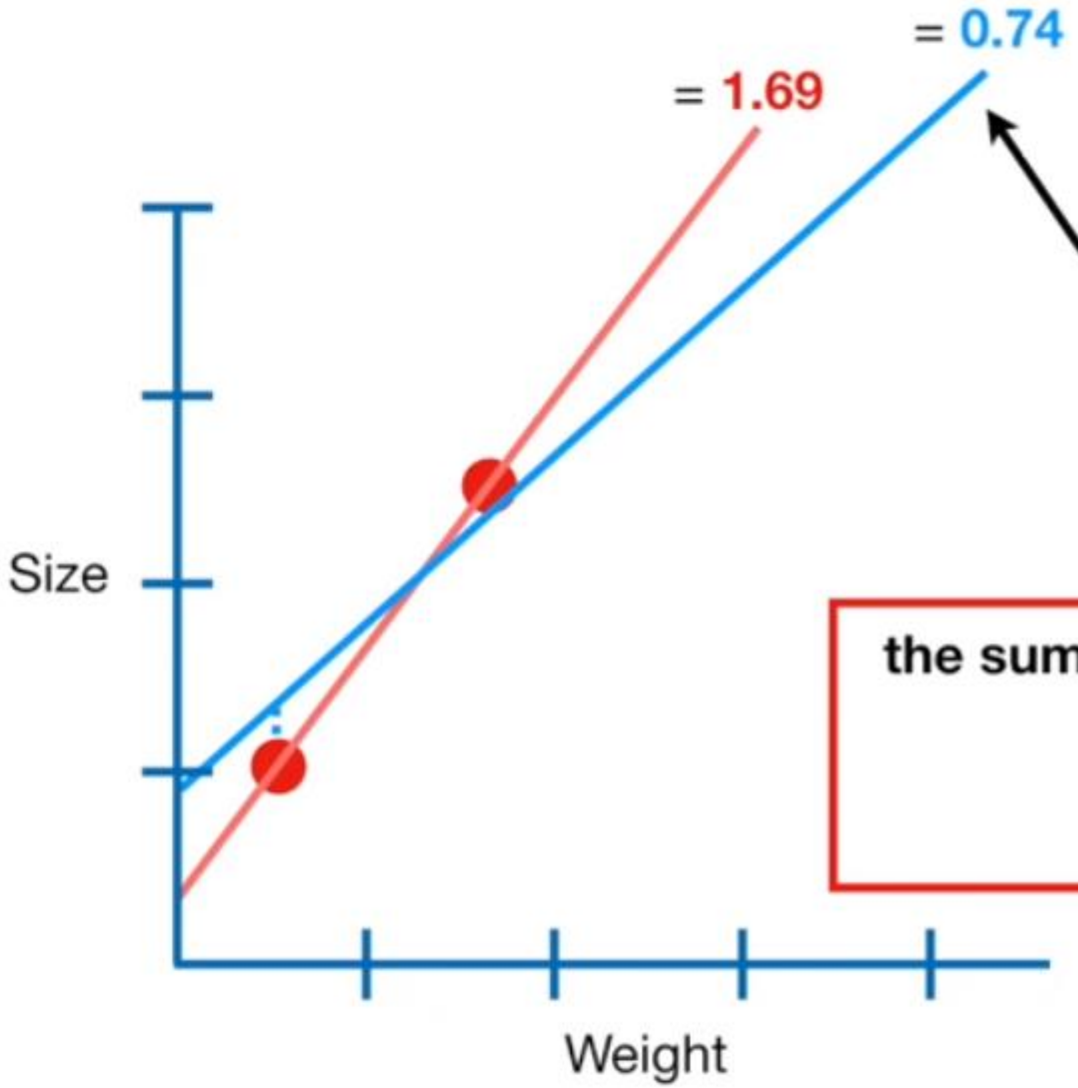


For the **Least Squares Line**, the sum of squared residuals plus the **Ridge Regression Penalty** is **1.69**…

the sum of the squared residuals

+

$\lambda \times$ **the slope**^{2}

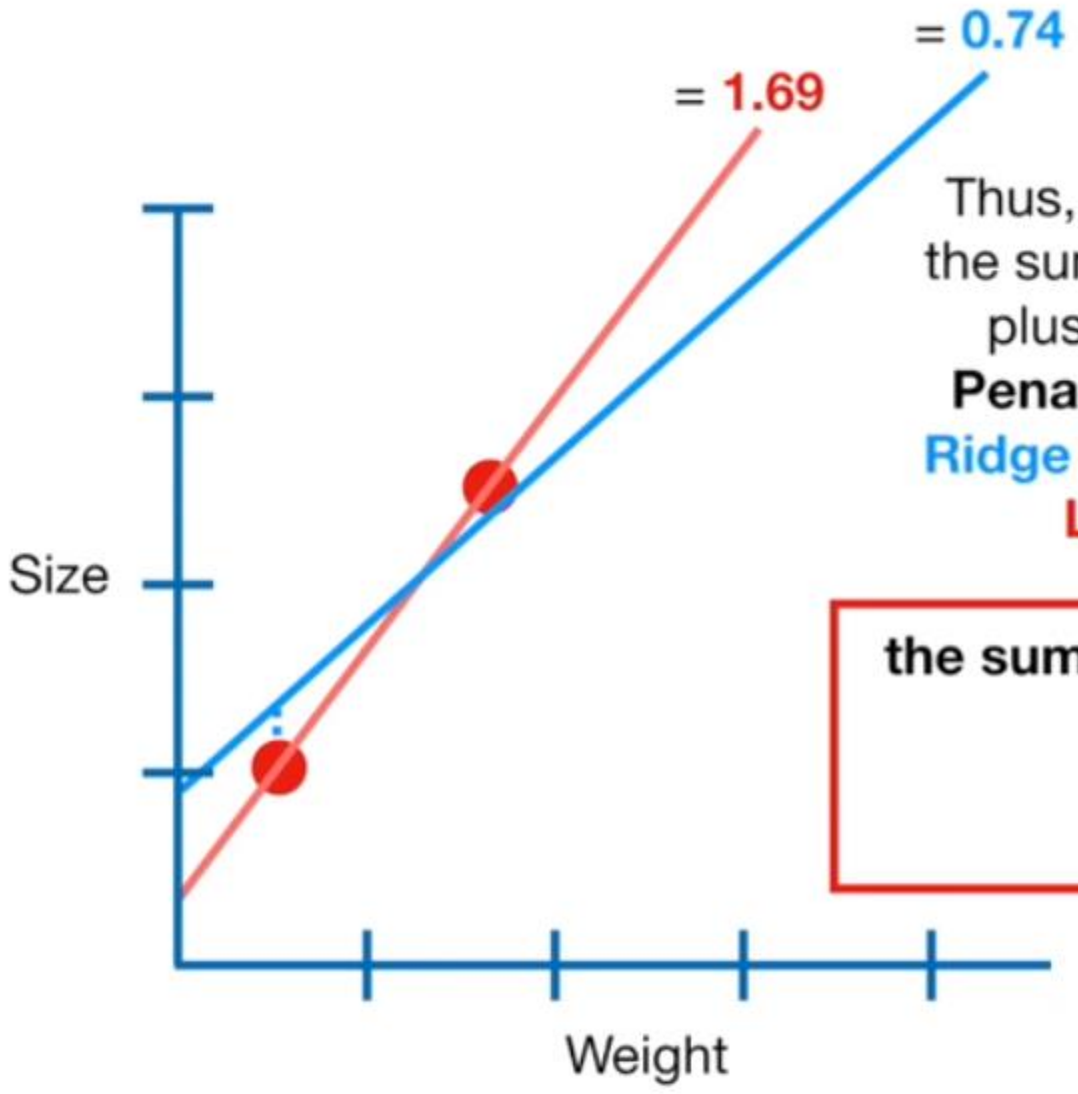


For the **Ridge Regression Line**, the sum of squared residuals plus the **Ridge Regression Penalty** is **0.74**…

the sum of the squared residuals

+

$\lambda \times$ the slope^2

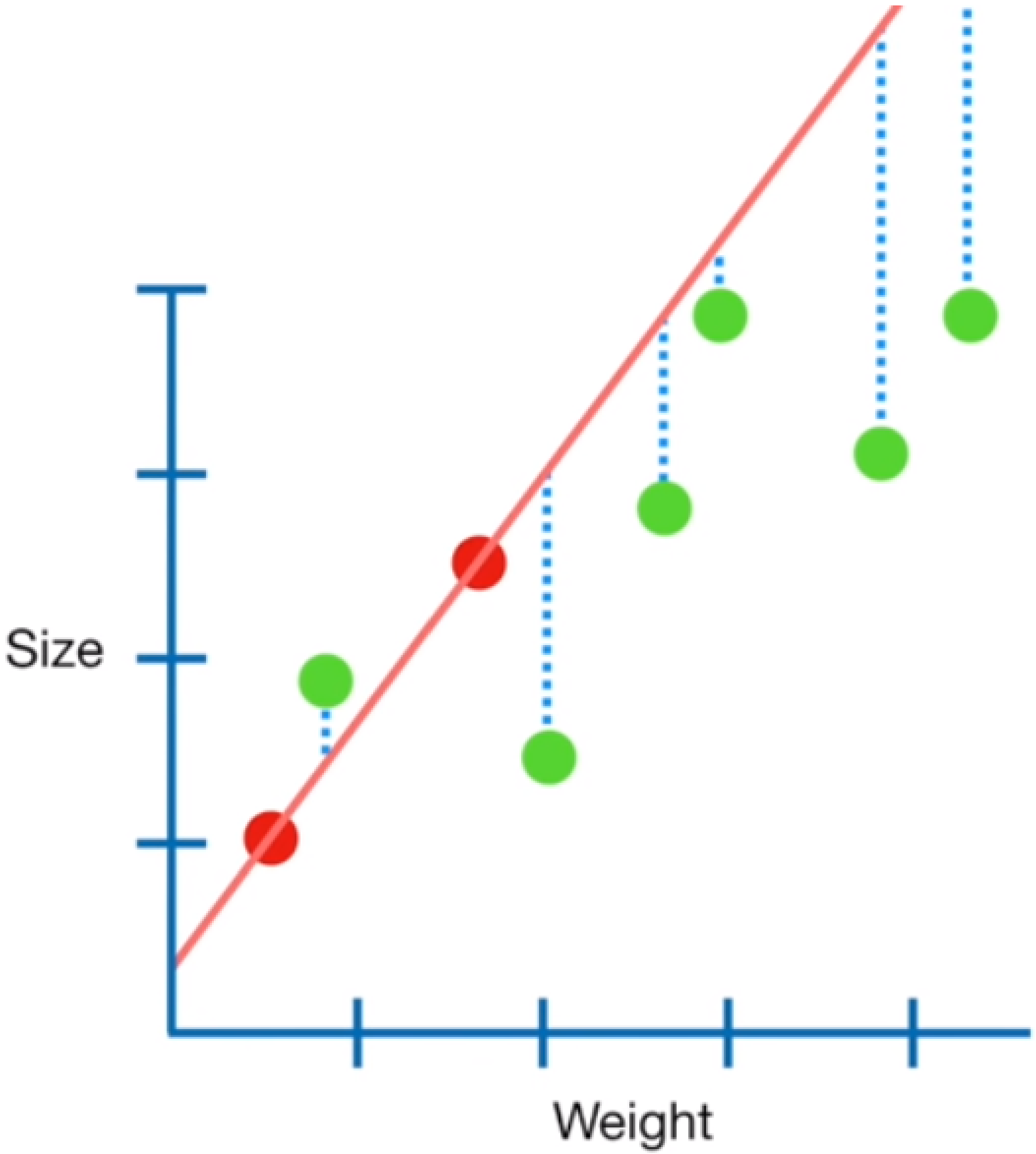


Thus, if we wanted to minimize the sum of the squared residuals plus the **Ridge Regression Penalty**, we would choose the Ridge Regression Line over the Least Squares Line.

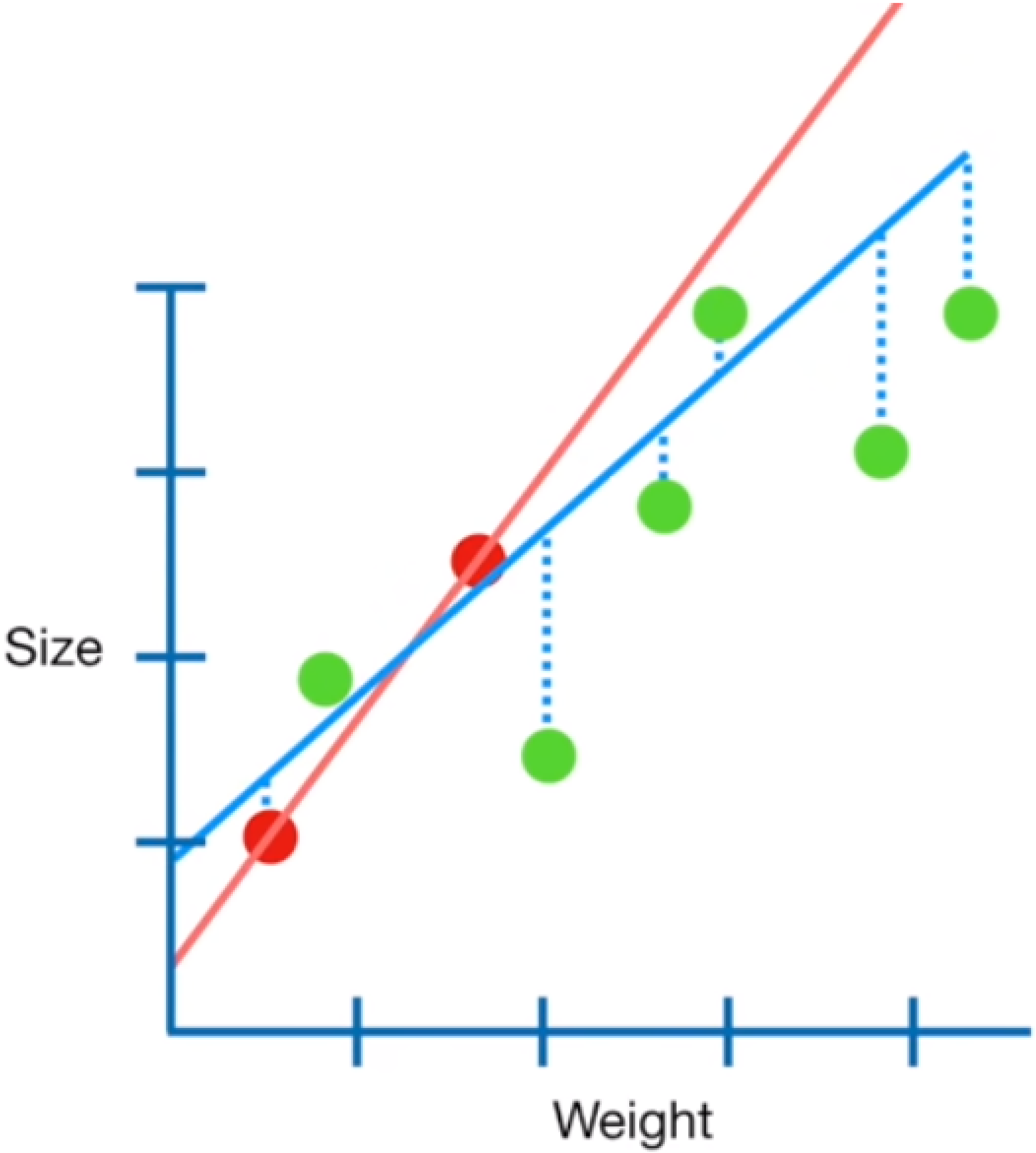
the sum of the squared residuals

+

$\lambda \times$ the slope^2

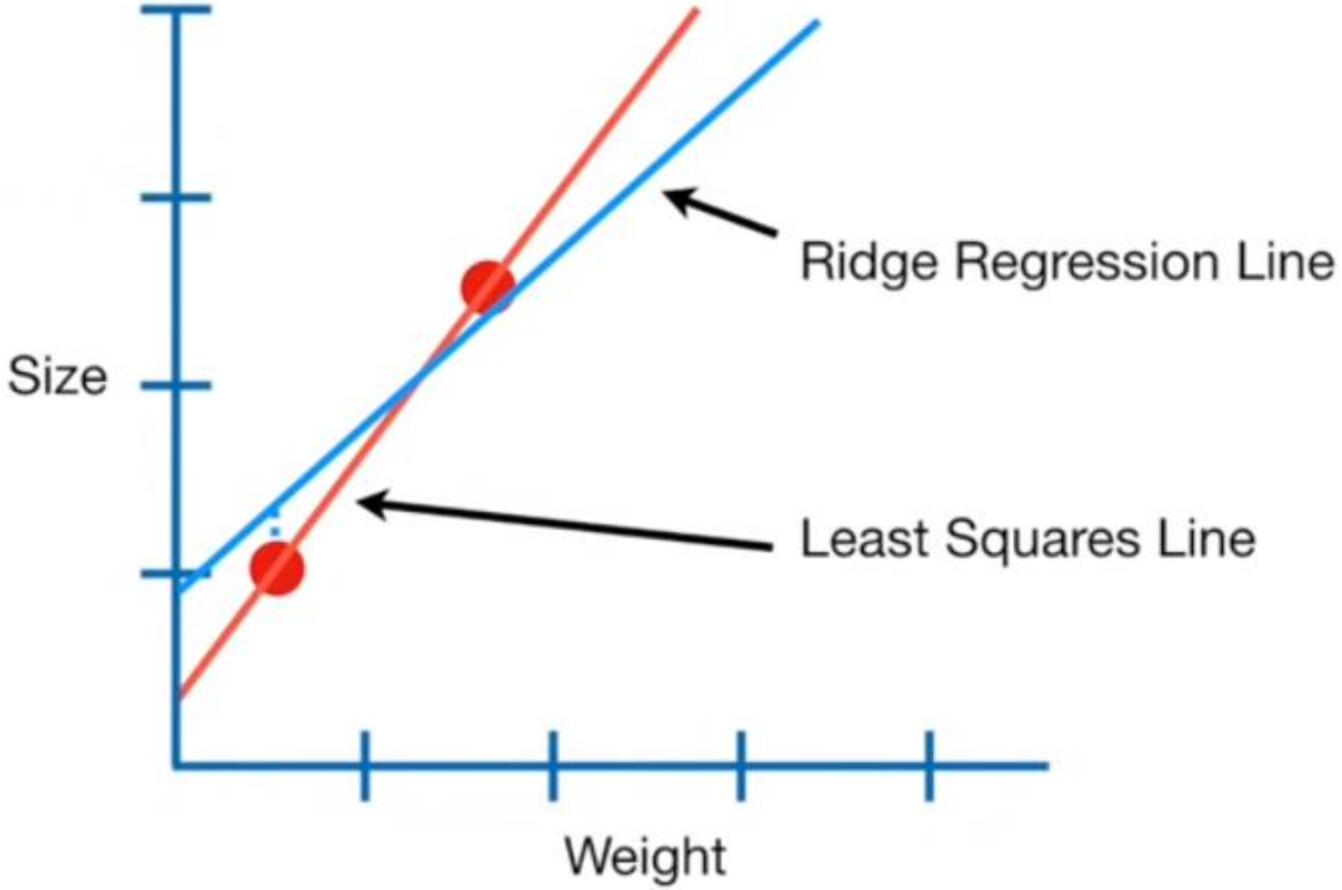


Without the small amount of **Bias** that the penalty creates, the **Least Squares Fit** has a large amount of **Variance**.

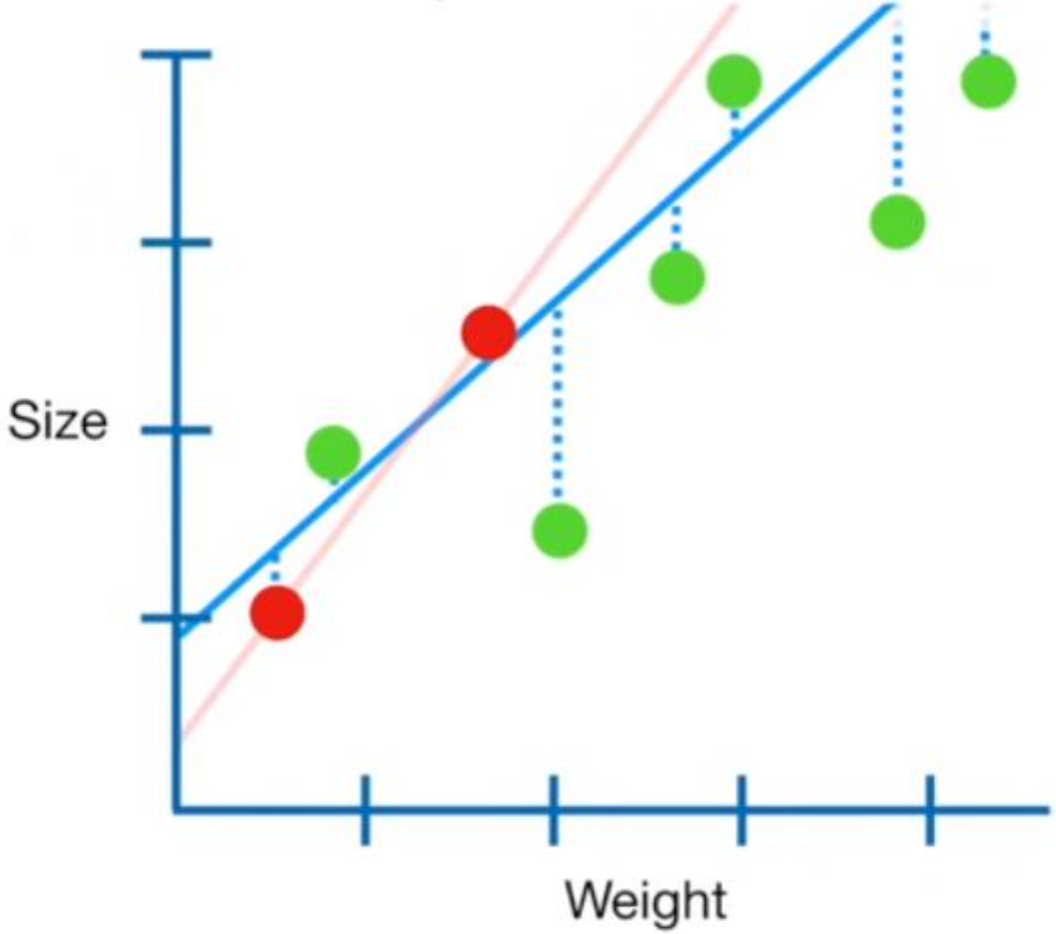


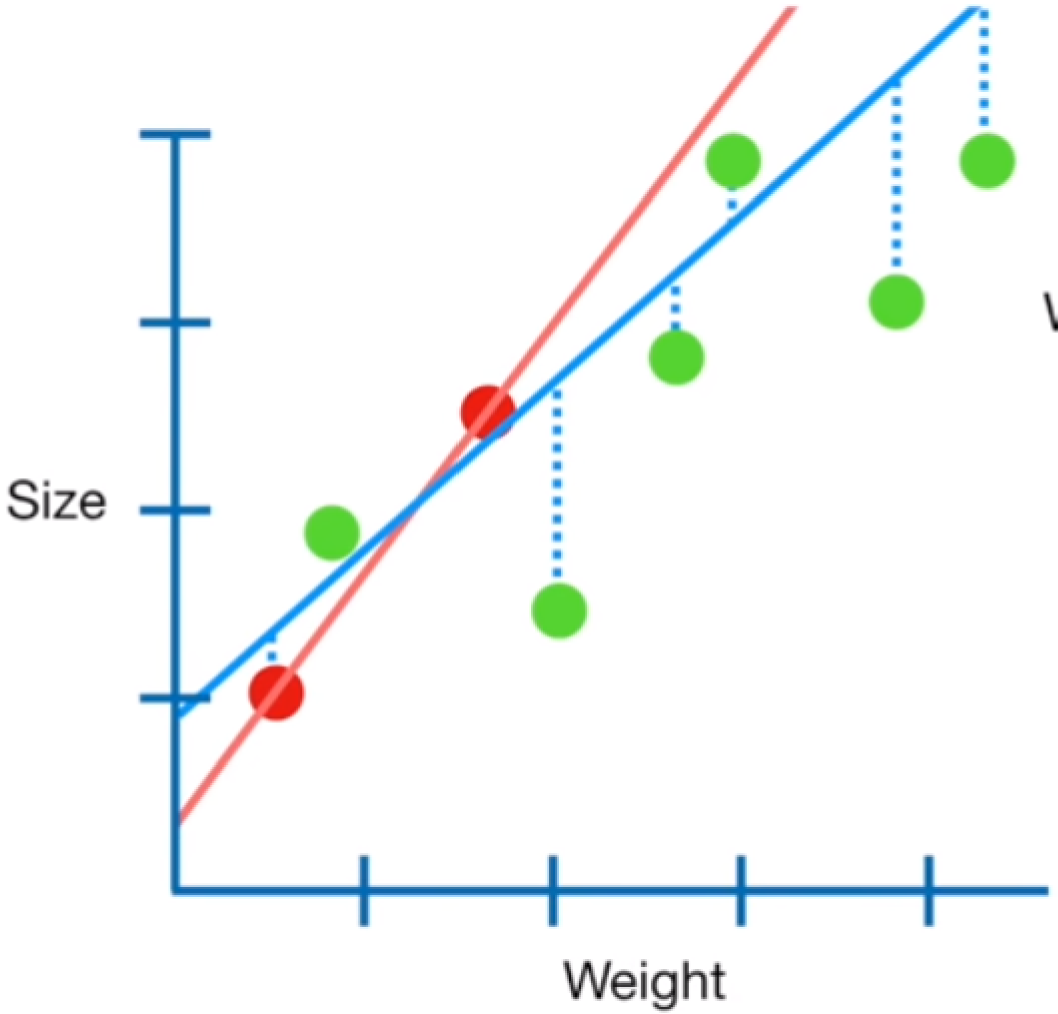
In contrast, the **Ridge Regression Line**, which has the small amount of **Bias** due to the penalty, has less **Variance**.

In other words, **Ridge Regression** had more **Bias** than **Least Squares**...

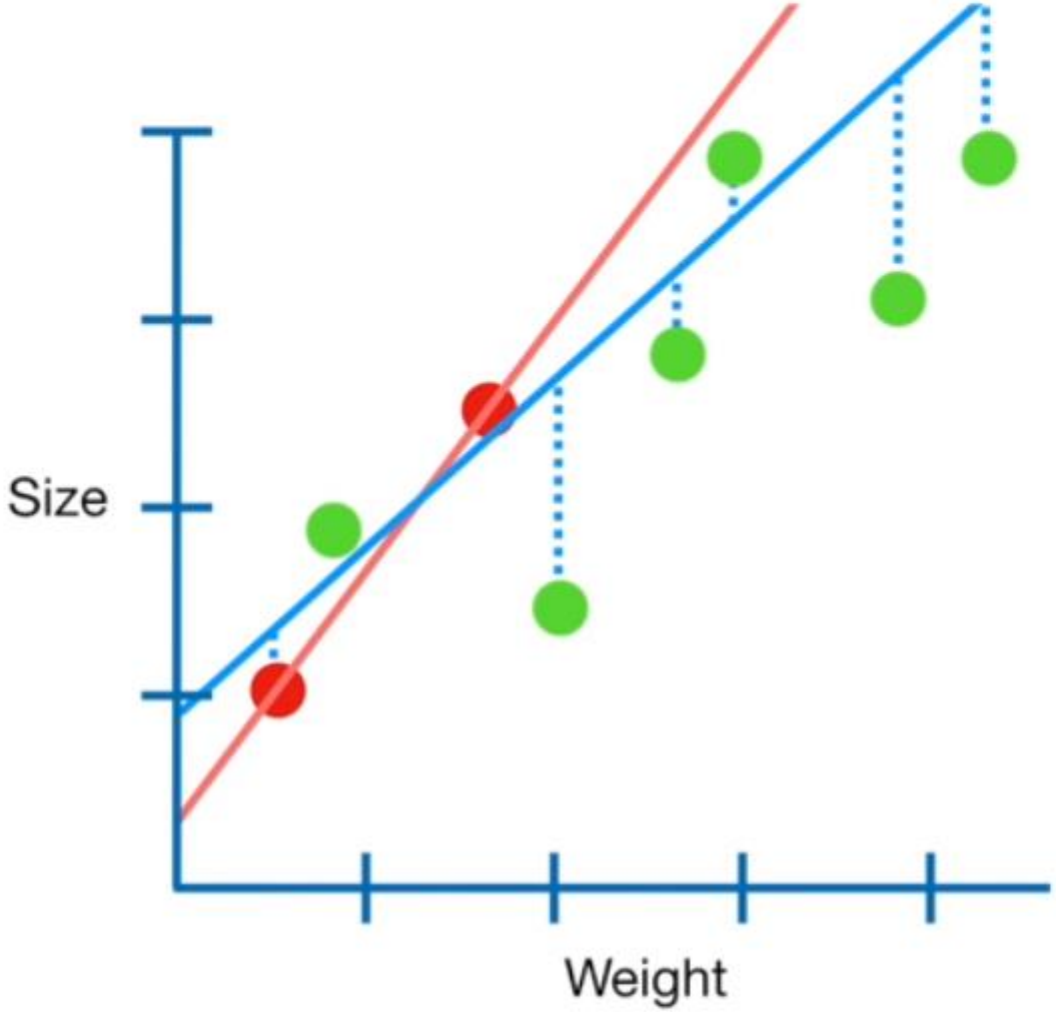


The main idea was that by starting with a slightly worse fit, **Ridge Regression** provided better long term predictions.





When the sample sizes are relatively small, then **Ridge Regression** can improve predictions made from new data (i.e. reduce **Variance**) by making the predictions less sensitive to the **Training Data**.



The **Ridge Regression Penalty** itself is **λ** times the sum of all squared parameters, except for the y-intercept…

the sum of the squared residuals

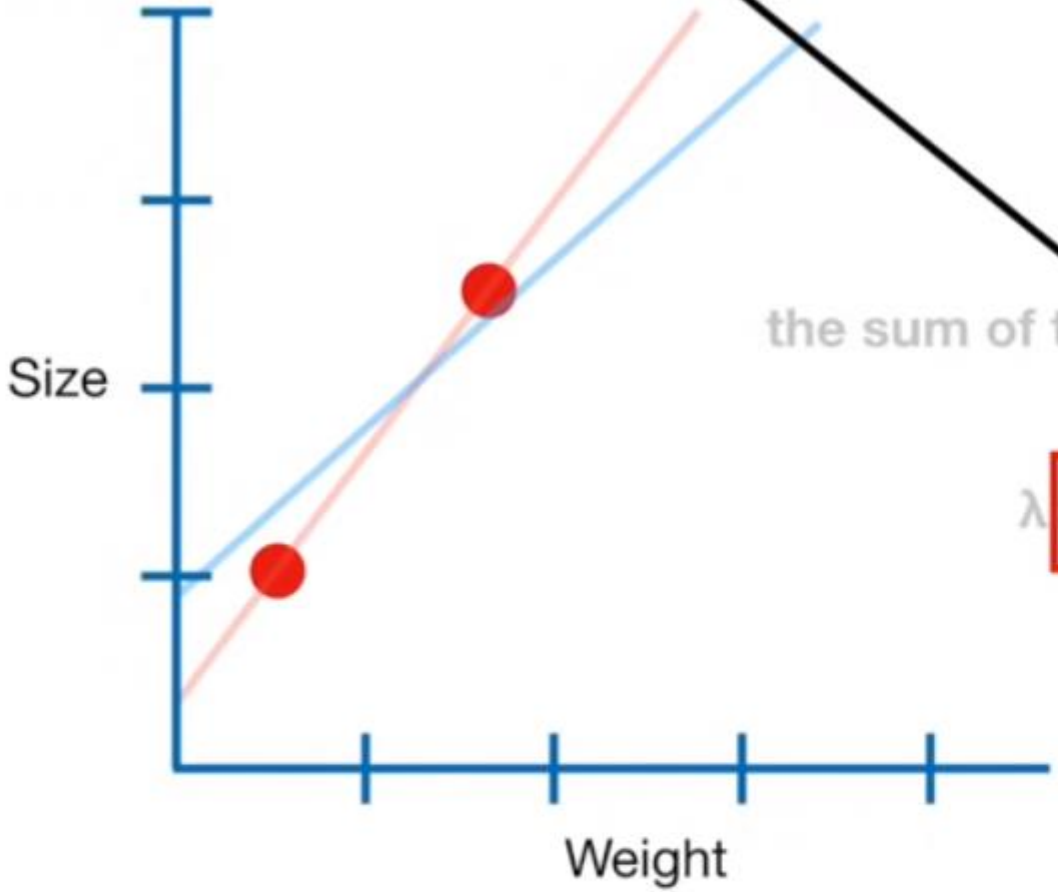
+

$\boxed{\boldsymbol{\lambda} \times \textbf{Slope}^2}$

λ can be any value from **0** to
positive infinity.

Lasso Regression is very, very similar to **Ridge Regression**, but it has some very, very important differences…

If, instead of squaring the slope…

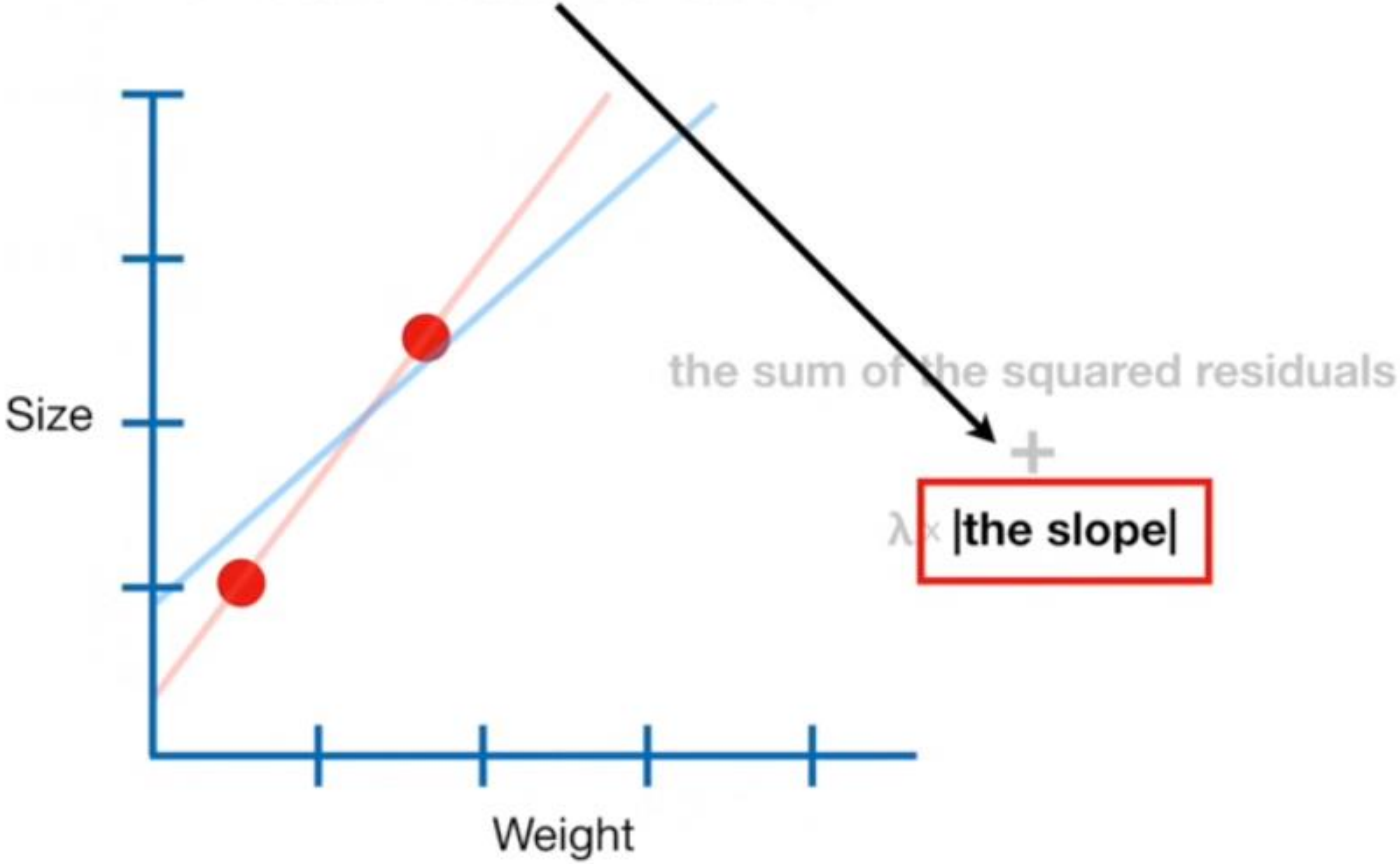


the sum of the squared residuals

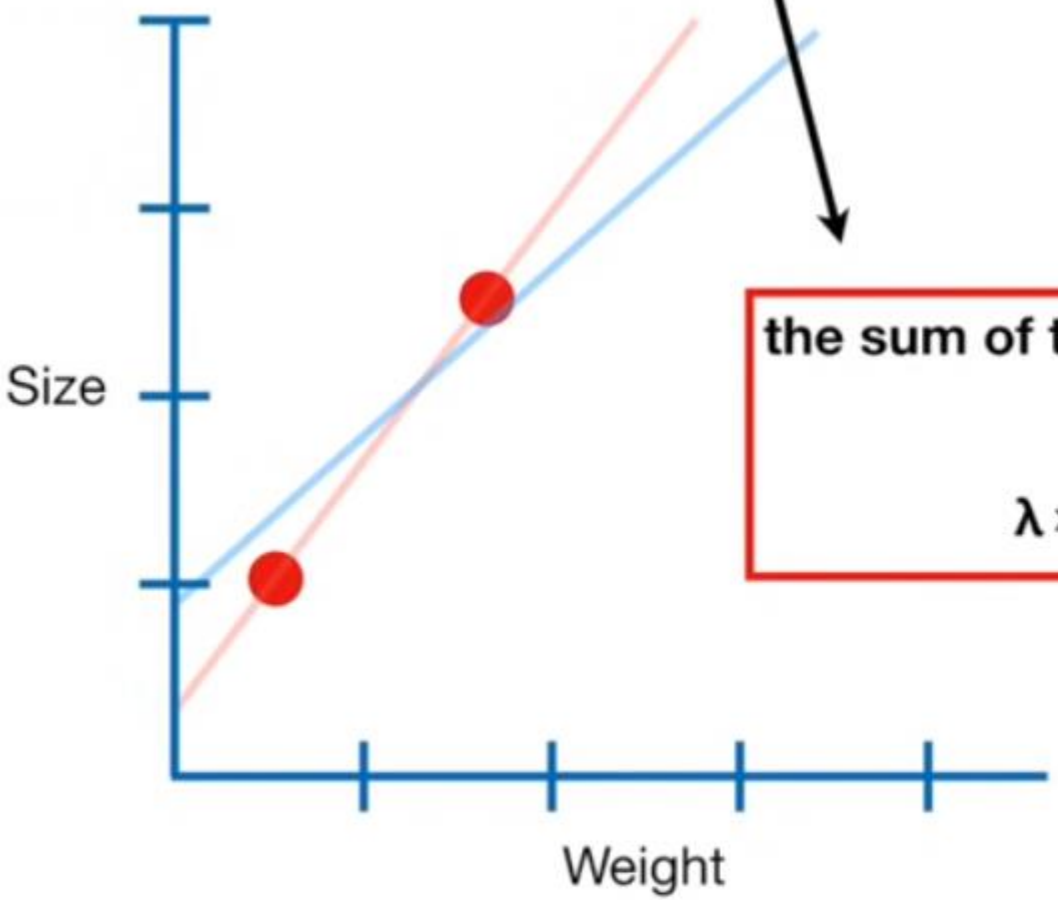
+

$\lambda \times$ $\boxed{\text{the slope}^2}$

…we take the absolute value…



…then we have **Lasso Regression**!!!

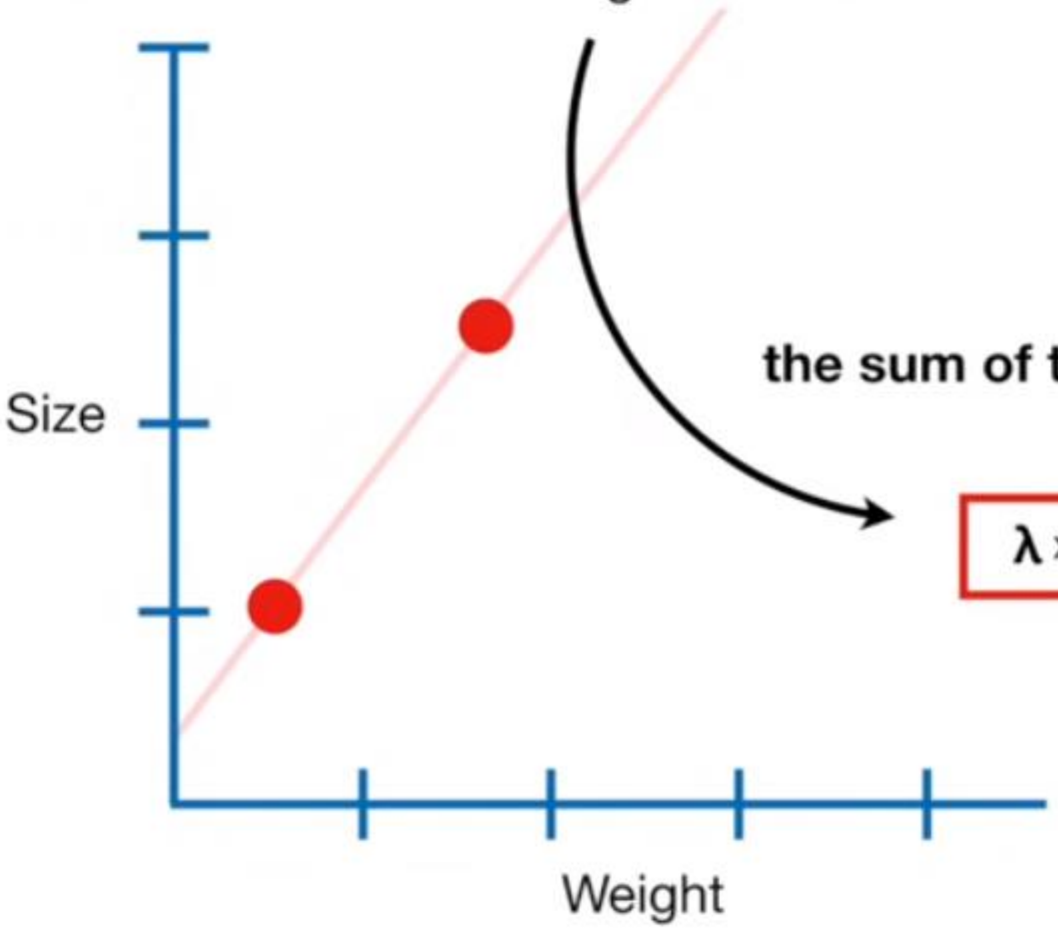


the sum of the squared residuals

+

$\boldsymbol{\lambda} \times$ **|the slope|**

NOTE: Just like with **Ridge Regression**, **λ** can be any value from **0** to **positive infinity** and is determined using **Cross Validation**.

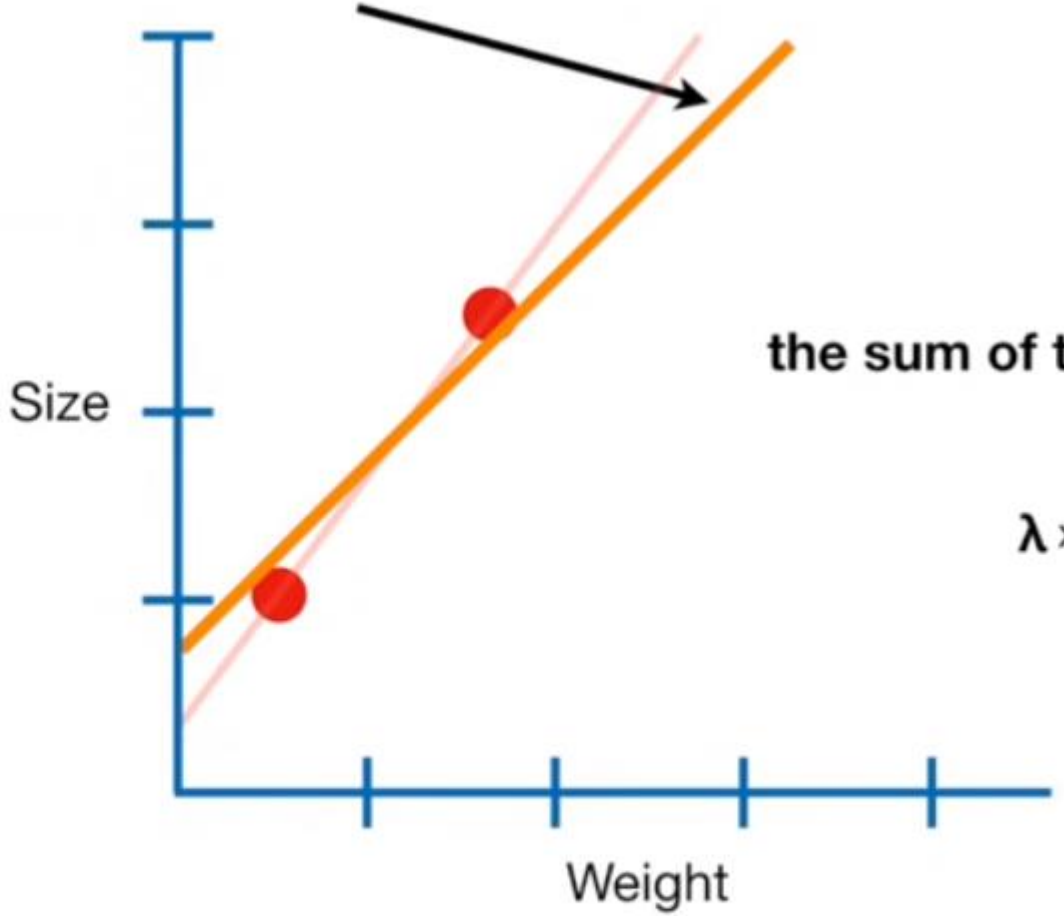


the sum of the squared residuals

+

$\boxed{\boldsymbol{\lambda} \times |\textbf{the slope}|}$

Like **Ridge Regression**, **Lasso Regression** (the **Orange Line**) results in a line with a little bit of **Bias**…

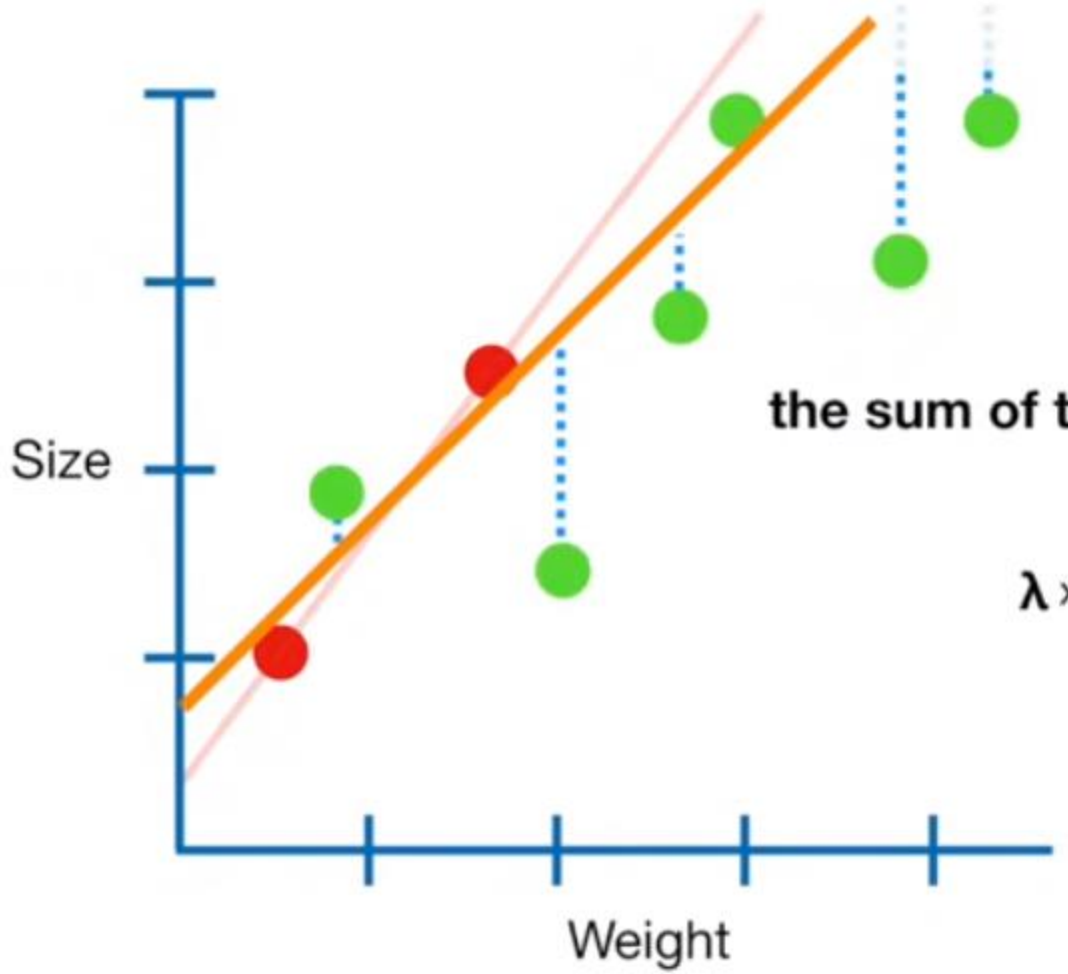


the sum of the squared residuals

+

λ × **|the slope|**

...but less **Variance** than **Least Squares**.



the sum of the squared residuals

+

$\lambda \times$ |the slope|

...look very similar....

the sum of the squared residuals

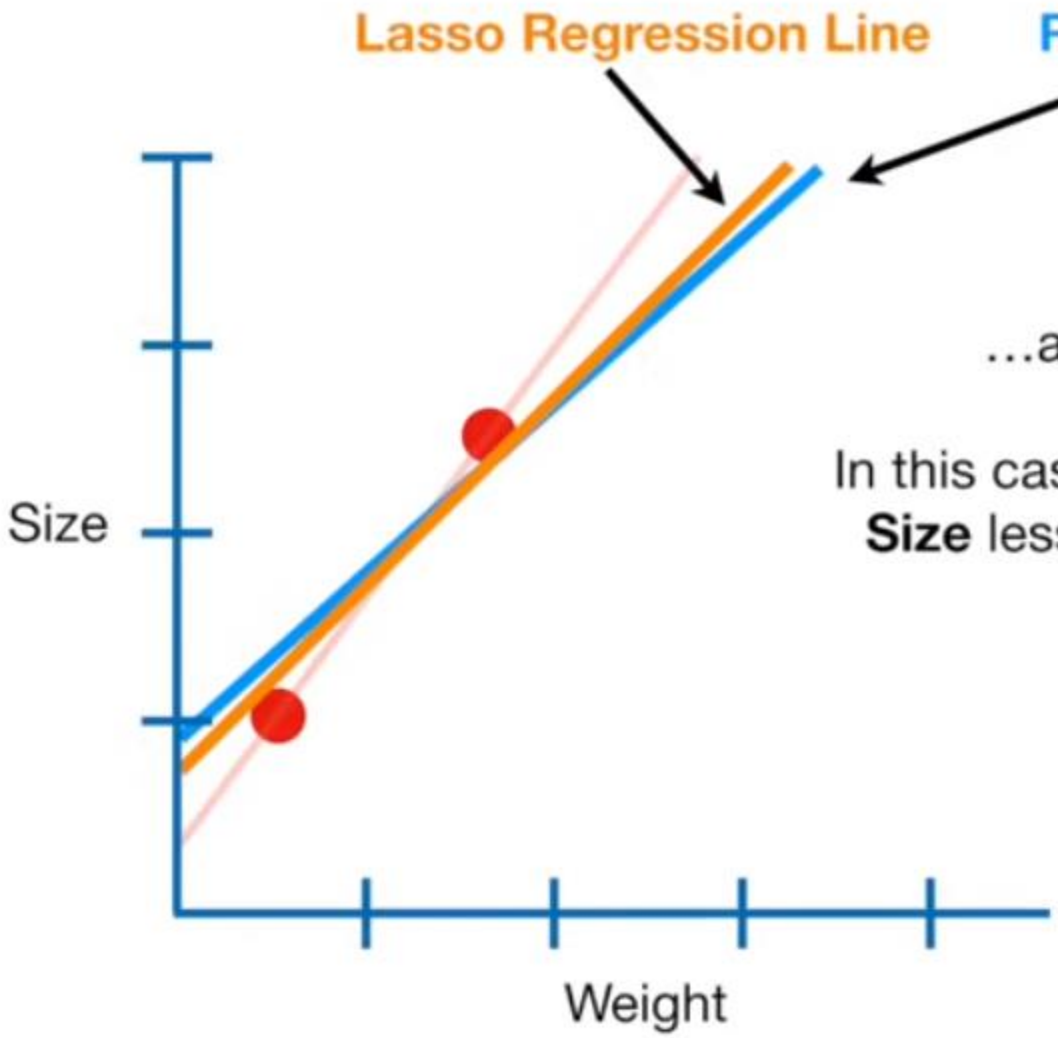
+

$\lambda \times$ the slope^2

the sum of the squared residuals

+

$\lambda \times$ |the slope|



…and they do similar things.

In this case, they make our predictions of **Size** less sensitive to this tiny **Training Dataset**.