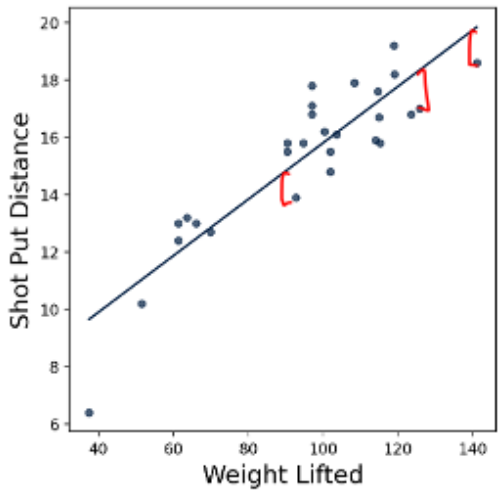
$$SD\ of\ x = \sqrt{\frac{\sum_{i=1}^{N}(x - \bar{x})^2}{N}}$$

```
mean = sum(x) / len(x)
total = 0
for i in range(len(x)):
    total += (x[i] - mean) ** 2
sd = math.sqrt(total / len(x))
```

$$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \end{bmatrix} - \begin{bmatrix} \bar{x} \\ \bar{x} \\ \bar{x} \end{bmatrix}$$

```
np.sum((x - np.mean(x)) ** 2)
```

## Line of best fit




1. Regression Line

$$slope = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{N}(x_i - \bar{x})^2}$$

$$intercept = \bar{y} - slope \cdot \bar{x}$$

2. Line minimizing mean squared error (MSE)

$$mse = \frac{\sum_{i=i}^{N}(y_i - (slope \cdot x_i + intercept))^2}{N}$$

3. Result of library functions for fitting

Judge, et al. International Journal of Exercise Science, 6, 2013.

Data from 28 female collegiate shot put athletes: biggest amount (in kilograms) that the athlete lifted in the “1RM power clean” in the pre-season and their personal best shotput distance (in meters)

The “line of best fit”, is:
1. The the “regression line”, i.e., the line that predicts the shotput distance as a function of weight lifted. And specifically simple linear regression, i.e., a linear regression model with one independent or “explanatory” variable, x, and one dependent variable, y.
2. The line that minimizes the mean squared error (MSE) between the fitted or predicted values and the actual values.
3. The result of library functions for computing line of best fit

$$slope = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{N}(x_i - \bar{x})^2}$$

Which of the following snippets are equivalent to this expression for NumPy arrays x and y? Recall $\bar{x}$ is the mean of $x$.

A.
```
x_m = np.mean(x)
y_m = np.mean(y)
num = np.sum(x-x_m)*np.sum(y-y_m)
slope = num / np.sum((x-x_m)**2)
```

B.
```
x_m = np.mean(x)
y_m = np.mean(y)
ratio = (x-x_m)*(y-y_m) / (x-x_m)**2
slope = np.sum(ratio)
```

C.
```
x_m = np.mean(x)
y_m = np.mean(y)
num = np.sum((x-x_m)*(y-y_m))
slope = num / np.sum((x-x_m)**2)
```

D.
```
x_m = np.mean(x)
y_m = np.mean(y)
num = np.sum((x-x_m)*(y-y_m))
slope = num / np.sum(x-x_m)**2
```

Answer: C

C is a direct translation of the expression. In (A), the numerator is the product of the sums, not the sum of the products. (B) has the incorrect sum, and (D) is dividing by the square of the sum, not the sum of the squares.

## Where did the slope of the regression line come from?

$$slope = \frac{\sum_{i=1}^{N}(x_i-\bar{x})(y_i-\bar{y})}{\sum_{i=1}^{N}(x_i-\bar{x})^2} = r \cdot \frac{SD\ of\ y}{SD\ of\ x} \qquad r = \frac{\sum_{i=1}^{N}\frac{(x_i-\bar{x})}{SD\ of\ x}\cdot\frac{(y_i-\bar{y})}{SD\ of\ y}}{N}$$

$$slope = \frac{\sum_{i}^{N}\frac{(x_i-\bar{x})}{SDx}\frac{(y_i-\bar{y})}{SDy}\cdot\frac{SDy}{SDx}}{N} \qquad \sqrt{\frac{\sum(x_i-\bar{x})^2}{N}}$$

$$= \frac{\sum_{i}^{N}(x_i-\bar{x})(y_i-\bar{y})}{(SDx)^2} = \frac{\sum_{i}^{N}(x_i-\bar{x})(y_i-\bar{y})\,N}{\sum(x_i-\bar{x})^2}$$

$$slope = \frac{\sum_{i=1}^{N}(x_i-\bar{x})(y_i-\bar{y})}{\sum_{i=1}^{N}(x_i-\bar{x})^2}$$

The slope is derived from the correlation coefficient, r, as shown. The correlation coefficient, *r*, is a measure of the strength and direction of a linear relationship between two variables. The correlation coefficient ranges from -1 to 1, with 1 indicating a perfect positive linear relationship, -1 indicating a perfect negative linear relationship, and 0 indicating no linear relationship. *r* is calculated as the average of the product of the two variables, when they are transformed into standard units. To transform a variable into standard units, we subtract the mean of the variable and divide by the standard deviation. Plugging that into the definition of r gives the expression on the right. We can now express the slope as …

$$mse = \frac{\sum_{i=i}^{N}(y_i - (slope \cdot x_i + intercept))^2}{N}$$

Which of the following snippets are equivalent to this expression for NumPy arrays x and y?

A.
```
fit = slope*x + intercept
mse = np.mean((y-fit)**2)
```

B.
```
fit = slope*x + intercept
mse = np.sum((y-fit)**2) / len(slope)
```

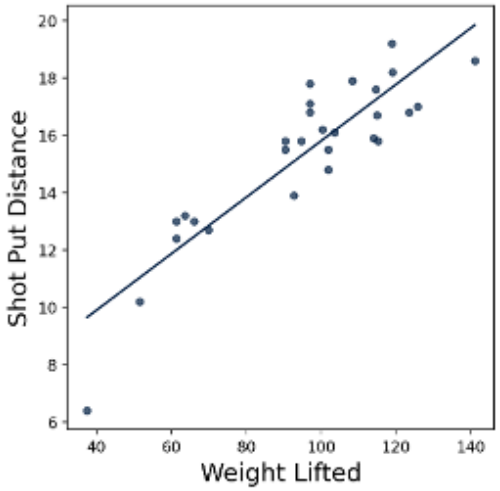
C.
```
fit = slope*x + intercept
mse = np.mean(y-fit)**2
```

D.
```
fit = slope*x + intercept
mse = np.mean(y-fit**2)
```

Answer: A

Answers A and B look very similar, but we notice that in (B) we are dividing by the length of slope. But slope is a single value (a scalar), not a vector so it doesn't have a length (this code would produce an error). We could fix it by taking the length of x instead.

## Linear or quadratic?




Judge, et al. International Journal of Exercise Science, 6, 2013.

"A Pilot Study Exploring the Quadratic Nature of the Relationship of Strength to Performance Among Shot Putters"