Statistical Features - Basics

We'll cover the following

Basic Concepts

The first step in analyzing data is to get familiar with it. Our good old NumPy provides a lot of methods that can help us do this easily. We are going to look at some of these methods in this lesson. Along the way, we are going to understand the meaning of important statistical terms as we encounter them.

The most basic yet powerful terms that you could come across are the **mean**, **mode**, **median**, **standard deviation**, and **correlation coefficient**. Let's understand these with an example dataset and using NumPy.

Say we have a dataset consisting of students' exams scores and the time they invested in studying for the exam. What can we learn about this data using statistics?

Run the code in the widget below and try to understand what's happening before reading the description that follows.

```
1  import numpy as np
2
3  # The dataset
4  learning_hours = [1, 2, 6, 4,
5  scores = [3, 4, 6, 5, 6]
6
7  # Applying some stats methods
8  print("Mean learning time: ",
9  print("Mean score: ", np.mean(
```

```
print("Median learning time: "
print("Standard deviation: ",
print("Correlation between lea
```

Mean

The mean value is the **average of a data set**, the sum the elements divided by the number of elements. As the name says, `np.mean()` returns the arithmetic mean of the dataset.

Median

The median is the **middle element** of the set of numbers. If the length of the array is odd, `np.median()` gives us the middle value of a sorted copy of the array. If the length of the array is even, we get the average of the two middle numbers.

Standard Deviation

Standard deviation is a **measure of how much the data is spread out**, and is returned by the `np.std()` method. More specifically, standard deviation shows us how much our data is spread out around the mean. Standard deviation could answer the questions *“Are all the scores close to the average?”* or, "*Are lots of scores way above or way below the average score?*" Using standard deviation we have a *standard* way of knowing what is normal and what is high or extra low.

In mathematical terms, standard deviation is the square root of the **variance**. So now you ask, *“What is variance?”*
Variance is defined as the average of the squared differences from the mean.** Let me break this down for you.

To calculate the variance manually we would follow these steps:

1. Compute the mean (the simple average of the numbers)
2. Then for each number, subtract the mean and square the result, i.e., the squared difference.
3. Then compute the average of those squared differences.

3. Then compute the average of these squared differences.

Let's calculate the standard deviation for learning hours manually. First let's get the mean value:

$$Mean = \frac{1+2+6+4+10}{5} = 4.6$$

Now to calculate the variance, get the difference of each element from the mean, square that, and then average the result:

$$Variance = \frac{(1-4.6)^2+(2-4.6)^2+(6-4.6)^2+(4-4.6)^2+(10-4.6)^2}{5} = 10.24$$

Finally, the standard deviation is just the square root of the variance, so:

$$StandardDeviation = \sqrt{10.24} = 3.2$$

Doing these calculations step by step makes us appreciate how easy NumPy makes our life. It allows us to perform statistical analysis without having to remember any mathy formulas and/or long steps — ***simple and neat!***

Correlation Coefficient

When two sets of data are strongly linked together we say they have a high correlation.

Fun fact: ***the word Correlation is made of Co,meaning "together", and Relation.***

Correlation is positive when the values for the two sets of data increase together, while correlation is negative when one value decreases as the other increases. For example, caloric intake and weight, or time spent studying and GPA, are highly correlated data; while a person's name and the type of food they prefer is an example of low correlation.

A correlation coefficient is a way to put a value to the relationship. Correlation coefficients have a value of between -1 and 1:

- 1 is a perfect positive correlation
- 0 is no correlation meaning the values don't seem linked at all
- -1 is a perfect negative correlation

`np.corrcoef()` returns a matrix with the correlation coefficients. This method

comes in handy when we want to see if there is a correlation between two or more variables in our dataset.

Correlation Is Not Causation!

Correlation Is Not Causation is a very common saying. What it really means is that a correlation does not prove one thing causes the other.

For example, say an ice-cream shop has data for how many sunglasses were sold by a big store in the neighborhood over a period of time, and they decide to compare sunglasses sales to their ice cream sales. From their results, they find a high correlation between the sales of the two. ***Does this mean that sunglasses make people want to buy ice cream?*** Hmm, no!

So, in layman terms, *"Correlation Is Not Causation"* tries to remind us that correlation does not prove one *thing*/*event* causes the other, but:

- One event **might** cause the other
- The other **might** cause the first to happen
- They **may** be linked by a different reason
- Or the result **could** have been random chance

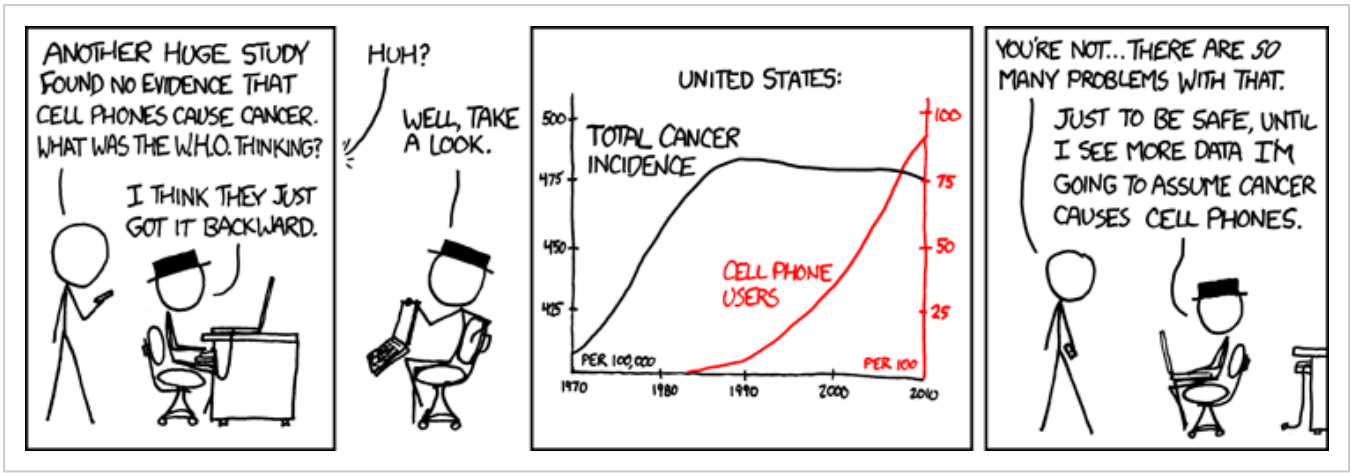

Image Credits: https://xkcd.com/925/

Getting back to our first example, performing a simple statistical analysis on our dataset gave us a lot of insights. In summary, we can see that:

- The mean learning hours is 4.6.
- The mean score is 4.8.
- The median learning hours is 4.0.

- The standard deviation for the learning hours is 3.2.

- There is a high correlation between how much a student studies in terms of hours and their final grade! Well, at least based on our made-up dataset.