The Weighted Mean and the Median: Takeaways

Syntax

COMPUTING THE WEIGHTED MEAN FOR A DISTRIBUTION WITH WEIGHTS

- Using numpy:

```
import numpy as np
weighted_mean_numpy = np.average(distribution_X, weights=weights_X)
```

- Coding a function from scratch:

```
def weighted_mean(distribution, weights):
    weighted_sum = []
    for mean, weight in zip(distribution, weights):
        weighted_sum.append(mean * weight)
    return sum(weighted_sum) / sum(weights)
weighted_mean_function = weighted_mean(distribution_X, weights_X)
```

FINDING THE MEDIAN

- Finding the median for a series:

```
median = Series.median()
```

- Finding the median for any numerical array:

```
import numpy as np
median_numpy = np.median(array)
```

Concepts

- When data points bear different weights, we need to compute **the weighted mean**. The formulas for the weighted mean are the same for both samples and populations, with slight differences in notation:

$$\bar{x} = \frac{\displaystyle\sum_{i=1}^{n} x_i w_i}{\displaystyle\sum_{i=1}^{n} w_i} = \frac{x_1 w_1 + x_2 w_2 + \ldots + x_n w_n}{w_1 + w_2 + \ldots + w_n}$$

$$\mu = \frac{\sum_{i=1}^{N} x_i w_i}{\sum_{i=1}^{N} w_i} = \frac{x_1 w_1 + x_2 w_2 + \ldots + x_n w_N}{w_1 + w_2 + \ldots + w_N}$$

- It's difficult to define the median algebraically. To compute the median of an array, we need to:
    - Sort the values in an ascending order.
    - Select the middle value as the median. If the distribution is even-numbered, we select the middle two values, and then compute their mean — the result is the median.
- The median is ideal for:
    - Summarizing numerical distributions that have **outliers**.
    - **Open-ended** distributions.
    - **Ordinal data**.

Resources

- An intuitive introduction to the weighted mean.
- The Wikipedia entry on the weighted mean.
- The Wikipedia entry on the median.
- Useful documentation:
    - numpy.average()
    - Series.median()
    - numpy.median()