The Weighted Mean and the Median: Takeaways

Syntax

- Computing the weighted mean for a distribution `distribution_X` with weights `weights_X`:

```
### Using numpy ###
from numpy import average
weighted_mean_numpy = average(distribution_X, weights =
weights_X)

### By 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 for a `Series`:

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

- Finding the median for any numerical array:

```
from numpy import median
median_numpy = 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:

- 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()