

**in order to calculate sector and industry return i need to calculate the average return for industry (all stocks retuirn/count of stock, equal weight, or take into account the cap weight , no?)**

To calculate a **sector or industry return**, you have two standard methods:

**1. Equal-Weighted Average Return (Simple)**

This is the **mean** of all constituent stock returns, treating each stock equally.

where:

* is the yearly return of the -th stock in the industry/sector
* is the total number of stocks in the group

This is the most commonly used for concise comparison and avoids bias to larger companies. Use when you want all stocks to contribute equally [script\_3.py][script\_4.py].

**2. Market Cap-Weighted Average Return (Advanced)**

This gives more influence to larger companies by weighting their returns by their market cap.

where:

* is the **market cap** of stock
* Other terms as above

Use this for more realistic “index-like” sector performance, especially when comparing with benchmarks like SPY, QQQ, etc.

**Which Should You Use?**

* **Equal-weight**: When you want to show the "typical" stock's performance (most straightforward, good for heatmaps, simple tables).
* **Cap-weight**: When you need to match how sector ETFs/benchmarks are constructed, or reflect investors who hold sectors based on company size.

**Implementation in Pandas**

**Equal-weight:**

industry\_avg\_return = df[df['Industry'] == industry]['yearly\_return'].mean()

**Cap-weight:**

group = df[df['Industry'] == industry]  
weighted\_avg = (group['yearly\_return'] \* group['market\_cap']).sum() / group['market\_cap'].sum()

**Summary**

* Both methods are valid. Choose based on your analysis purpose and audience.
* For dashboards or “top performer” comparison, **equal-weight** is typically more intuitive.
* For portfolio/index construction or ETF benchmarking, use **cap-weighted**.

You may also want to **display both** (side by side) for maximum information and transparency.