

## Exercise 1 Answer

**Question1. Where you did the same calculations with NumPy and Pandas, which did you find easier to work with? Which code do you think is easier to read?**

Pandas is easier since its higher-level functionality provides intuitive data manipulation capabilities.

Pandas code is also easier to read because it allows concise operations on the data, such as grouping, aggregation, and rotation, and these integrated functions will improve the reading experience

**Question2. What were the running times of the two `pivot_months_*` functions? How can you explain the difference?**

Running time of the `pivot_months_pandas(data)`: **26.7 ms ± 2.36 ms** per loop  
Running time of the `pivot_months_loops(data)`: **235 ms ± 11.2 ms** per loop

The `pivot_months_loops` function is expected to run slower than the `pivot_months_pandas` function. The `pivot_months_pandas` function takes advantage of the efficient grouping, aggregation, and pivoting functions provided by Pandas for faster execution.

```
In [6]: abs(totals_pd.values - totals_loop.values).sum() # the two totals should be the same
```

```
Out[6]: 0.0
```

```
In [7]: abs(counts_pd.values - counts_loop.values).sum() # ...counts too.
```

```
Out[7]: 0.0
```

```
In [8]: %timeit pivot_months_pandas(data)
```

```
26.7 ms ± 2.36 ms per loop (mean ± std. dev. of 7 runs, 10 loops each)
```

```
In [9]: %timeit pivot_months_loops(data)
```

```
235 ms ± 11.2 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
```