

-- Step 0: Steps data (rows = people, columns = days)

```
steps = np.array([
    [5000, 7000, 8000, 6000, 7500, 9000, 10000],
    [10000, 11000, 9500, 12000, 11500, 9000, 10500],
    [3000, 4000, 3500, 4500, 4000, 4200, 3800],
    [7000, 8000, 7500, 7200, 7100, 8000, 8500],
    [6000, 6500, 7000, 6400, 6800, 7200, 6900],
    [9000, 9500, 8800, 9700, 9300, 9100, 9400],
    [2000, 3000, 2500, 2700, 2200, 2600, 2800],
    [8000, 8500, 9000, 8700, 8800, 8900, 8600],
    [4000, 4500, 5000, 4700, 4300, 4800, 4600],
    [10000, 12000, 11000, 11500, 11700, 11800, 11900]
])
```

Heart rate data (average per day per person)

```
heart_rate = np.array([
    [70, 72, 75, 71, 73, 74, 70],
    [65, 66, 68, 67, 66, 65, 66],
    [80, 82, 81, 79, 80, 83, 81],
    [72, 74, 73, 71, 72, 75, 74],
    [78, 77, 79, 76, 77, 78, 76],
    [68, 69, 70, 68, 69, 67, 68],
    [85, 86, 84, 83, 85, 84, 86],
    [70, 71, 72, 70, 71, 73, 72],
    [80, 81, 79, 82, 80, 79, 81],
    [66, 65, 67, 66, 66, 65, 66]
])
```

-- **Step 1: Basic inspection**

- Print the shape of steps and heart_rate
- Print the first row (person 1's data)

-- **Step 2: Total steps per person**

- Calculate **total steps for each person** over the week
- Hint: np.XXX(steps, axis=1)

Answer example:

[52500, 73500, 27000, 53300, 46800, 64800, 17800, 60500, 31900, 79900]

-- **Step 3: Average heart rate per person**

- Calculate **average heart rate per person**
- Hint: np.XXX(heart_rate, axis=1)

Answer example:

[72.14285714, 66.14285714, 80.85714286, 73. , 77.28571429, 68.42857143, 84.71428571, 71.28571429, 80.28571429, 65.85714286]

-- **Step 4: Find the most active person**

- Identify the person with **highest total steps**
- Hint: np.XXX()

Answer: 9

-- **Step 5: Daily average steps**

- Calculate **average steps per day across all people**

Answer example:

[6400., 7400., 7180., 7340., 7320., 7460., 7700.]

-- **Step 6: Heart rate warning**

- Find all heart rates **>80 bpm**
- Hint: `heart_rate > 80`
- Count how many such values exist

Answer: 14

-- **Step 7: Replace very low step counts**

- Replace all step values **< 3000** with **3000**
- Hint: boolean indexing

-- **Step 8: Normalize steps**

- Divide all step values by **1000** to convert to “thousands of steps”

-- **Step 9: People who are active every day**

- A person is **active** if they walk ≥ 7000 **steps every day**
- Find indices of people who satisfy this condition
- Hint: boolean mask + `np.XXX(axis=1)`

Answer example: [1, 3, 5, 7, 9]

-- **Step 10: High effort days**

Create a boolean array where:

- True if steps ≥ 9000
- False otherwise
- Count total number of True values

Answer example:

23

-- Step 11: Data cleaning (realistic)

- Assume values > 15000 steps are errors
- Replace them with the **mean steps of that person**
- Hint: boolean indexing + np.mean(axis=1, keepdims=True)