# Vectorization

1. Consider

$$\sum_{i=1}^{m} a_i b_i$$

   - $a_i \in \mathbb{R}, \quad b_i \in \mathbb{R}$

2. For Python and other softwares, the computation efficiency of using for-loops for summation is quite low
3. However, matrix calculation is very efficient
4. Thus, we use vectorization for the summation

# Vectorization

1. Vectorization result

$$\sum_{i=1}^{m} a_i b_i = \boldsymbol{a}^{\mathrm{T}} \cdot \boldsymbol{b}$$

- $\boldsymbol{a} = (a_1, \ldots, a_m)^{\mathrm{T}} \in \mathbb{R}^{m \times 1}$
- $\boldsymbol{b} = (b_1, \ldots, b_m)^{\mathrm{T}} \in \mathbb{R}^{m \times 1}$

# Vectorization

1. Example 1: Consider

$$\sum_{i=1}^{m} a_i$$

   - $a_i \in \mathbb{R}$

2. Vectorization result

$$\sum_{i=1}^{n} a_i = \sum_{i=1}^{n} a_i \times 1 = \boldsymbol{a}^{\mathrm{T}} \cdot \mathbf{1} (= \mathbf{1}^{\mathrm{T}} \cdot \boldsymbol{a})$$

   - $\boldsymbol{a} = (a_1, \ldots, a_m)^{\mathrm{T}} \in \mathbb{R}^{m \times 1}$
   - $\mathbf{1} = (1, \ldots, 1)^{\mathrm{T}} \in \mathbb{R}^{m \times 1}$: a vector of 1's wiwth length $m$

# Vectorization

1. Example 2: Consider

$$\sum_{i=1}^{m} a_i \boldsymbol{b}_i$$

   - $a_i \in \mathbb{R}$: a scalar, which may be 1
   - $\boldsymbol{b}_i \in \mathbb{R}^{k\times 1}$

2. Vectorization result

$$\sum_{i=1}^{n} a_i \boldsymbol{b}_i = \boldsymbol{B} \cdot \boldsymbol{a}$$

   - $\boldsymbol{a} = (a_1, \ldots, a_m)^{\mathrm{T}} \in \mathbb{R}^{m\times 1}$
   - $\boldsymbol{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m) \in \mathbb{R}^{k\times m}$

# Vectorization

1. Example 3: Consider

$$\sum_{i=1}^{m} a_i \boldsymbol{b}_i{}^{\mathrm{T}}$$

- $a_i \in \mathbb{R}$: a scalar, which may be 1
- $\boldsymbol{b}_i \in \mathbb{R}^{k\times 1}$

2. Vectorization result

$$\sum_{i=1}^{n} a_i \boldsymbol{b}_i = \boldsymbol{a}^{\mathrm{T}} \cdot \boldsymbol{B}$$

- $\boldsymbol{a} = (a_1, \ldots, a_m)^{\mathrm{T}} \in \mathbb{R}^{m\times 1}$
- $\boldsymbol{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m)^{\mathrm{T}} \in \mathbb{R}^{m\times k}$