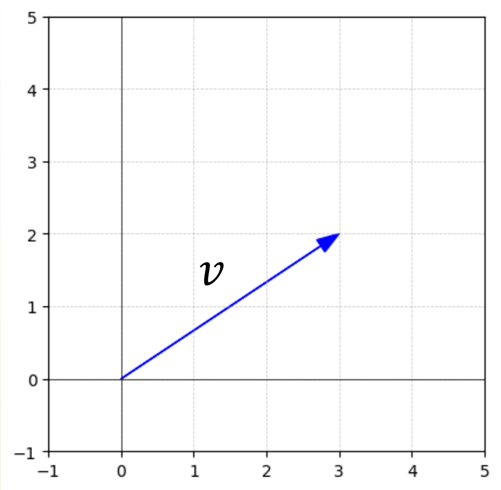
Data 345

Applied Linear Algebra for Statistical Learning

Class 2 (Aug. 28, 2025)

Vectors

- Let n be a positive integer. A (real) **vector** of dimension n is an ordered list of n real numbers.
- We will let $\mathbb{R}^n$ denote the set of all real n-dimensional vectors. The set $\mathbb{R}^n$ is referred to as a (real) n-dimensional **vector space**.
- The set $\mathbb{R}$ of real numbers is referred to as the underlying **scalar field** of the vector space $\mathbb{R}^n$. Elements of $\mathbb{R}$ are also referred to as **scalars**.
- Vectors are often represented as directed line segments from the origin of n-dimensional space. For instance, the vector $v = [3,2]$ can be represented as a directed line segment from $(0,0)$ to $(3,2)$ on the Euclidean plane.



Vectors

- From a formal mathematical perspective, a vector is just an ordered list of numbers. When it comes to computation, we'll also need to be aware of vector **orientation**, i.e., whether a vector is a **row vector** ($v = [3,2]$) versus a **column vector** ($v = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$), but we can get to this later.
- The individual entries in the list that forms a vector are called the vector's **coordinates**.
- The two-dimensional vector $v = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$ has 3 as its first coordinate and 2 as its second coordinate.
- **Note:** We will base our <vector> class on the Python <list>, but one needs to be aware that a <list> in Python is 0-indexed, which means the 0th item in the list refers to the first.

First item in list

$$\text{list} = [1, 10, 20, 200, 400]$$

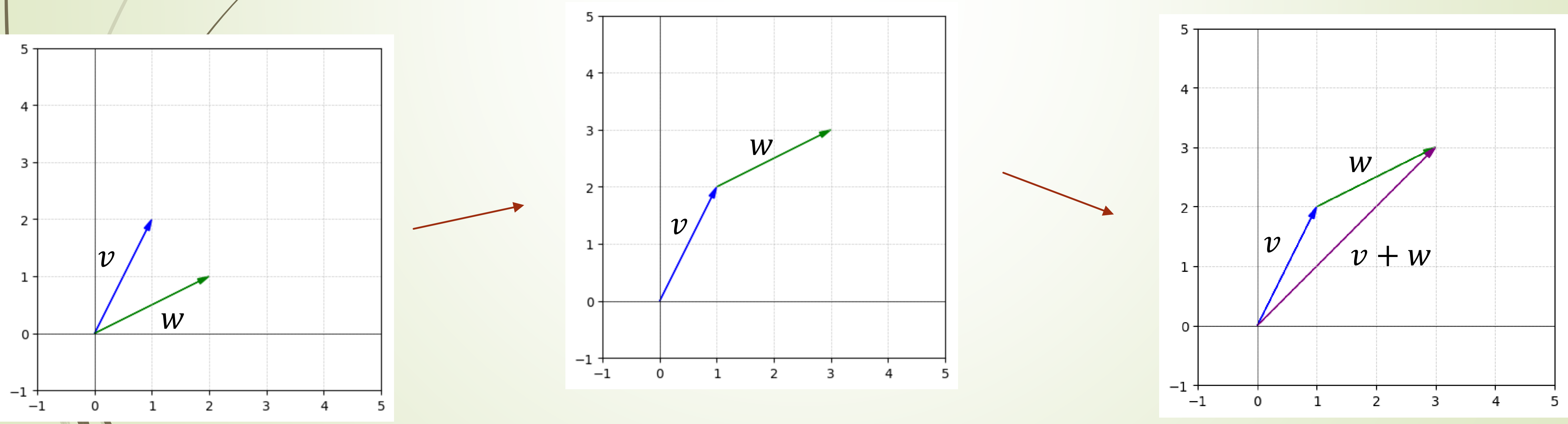
list[0]

Vectors

- Vectors are also imbued with some additional structure: we can *add* two vectors of the same dimension, and we can also *scale* a vector by a real number (which is why it's called a scalar).
- Let $v = [v_1, v_2, v_3, \dots, v_n]$ and $w = [w_1, w_2, w_3, \dots, w_n]$ be two n-dimensional vectors. Then the vector sum of v and w, $v + w$, is given as $v + w = [v_1 + w_1, v_2 + w_2, v_3 + w_3, \dots, v_n + w_n]$. That is, we simply add corresponding components of v and w together.
- Letting c be a real number, we may also define cv, the scalar multiple of v by c, as $cv = [cv_1, cv_2, cv_3, \dots, cv_n]$. This means each individual component of v is simply multiplied by c.
- **Note:** These two operations correspond to the row operations we used last time to "row-reduce" a matrix. This is important!

Vectors

- Remember that we could realize vectors geometrically as directed line segments.
- The vector sum $v + w$ is also realized this way. We can slide the initial vertex of w to the end of v, without changing direction.



Vectors

- As mentioned previously, we will be basing our `<vector>` class on Python's `<list>` object.
- At minimum, our `<vector>` class should:
  - Define a `<vector>` object in Python.
  - Allow for component-wise addition of two same-sized vectors.
  - Allow for scalar multiplication.
- Does a Python `<list>` already have this functionality?

```python
v = [1,2]
w = [2,1]
v + w
```

✓ 0.0s Python

```
[1, 2, 2, 1]
```

"Addition"

```python
v = [1,2]
3*v
```

✓ 0.0s Python

```
[1, 2, 1, 2, 1, 2]
```

"Scalar multiplication"

`<vector>` Behavior

Action	Old Behavior (`<list>`)	Desired Behavior (`<vector>`)
Add two like-sized lists (numbers)	Concatenate lists	Add numbers componentwise
Add two unlike lists (numbers)	Concatenate lists	Error
Add two lists (other)	Concatenate lists	Error
Multiply list by positive integer n	Concatenate list with itself n times	Multiply each component by n
Multiply list by negative integer n	Empties list	Multiply each component by n
Multiply list by non-integer number c	Error	Multiply each component by c

<vector> Declaration

- To create a new type of object in Python, we can define a **class**.

The <class> keyword tells Python we are creating a new object.

This is the name of the object and it's how we'll call new instances (case-sensitive!)

```python
class vector:
    def __init__(self, data):
        self.data = data
```

✓ 0.0s

Python

This is a special command (method) that constructs the object when called & initializes its attributes.

<self> is always the first parameter, referring to the instance itself.

This is the data we will use to define a vector; i.e., a list of numbers.

Vector Properties

- Let $V = \mathbb{R}^n$, with the previously mentioned operations of addition and scalar multiplication. Then, for any v, w, and u in V, and any $a, b \in \mathbb{R}$ the following properties hold.
- **Associativity (vector addition):** $u + (v + w) = (u + v) + w$
- **Commutativity (vector addition):** $v + w = w + v$
- **Identity (vector addition):** There exists a vector $0 \in V$ with the property that $0 + v = v + 0 = v$, for any $v \in V$.
- **Inverse (vector addition):** For every $v \in V$, there exists an element $-v \in V$, called the additive inverse of v, so that $v + (-v) = (-v) + v = 0$.
- **Compatibility of real number and scalar multiplication:** $a(bv) = (ab)v$.
- **Identity (scalar multiplication):** $1v = v$.
- **Distributivity (1):** $a(v + w) = av + aw$.
- **Distributivity (2):** $(a + b)v = av + bv$.