Linear Algebra Part I

Dhruva V. Raman

This is a minimal lecture to get you started

Some things are
better read/watched

3blue1brown lecture series

Cheatsheet

Notebook

Too much lecture
content is bad

Flipped learning:
questions at end of
lecture

Functions

Recap

Example 1

$$f(x) = x^2$$

$f : \mathbb{R} \to \mathbb{R}^+$

Domain: all real numbers

Range: all positive real numbers

Example 2

$$+(x, y) = x + y$$

$f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$

Domain: all pairs of real numbers

Range: all real numbers

Terminology

Domain: set of possible inputs

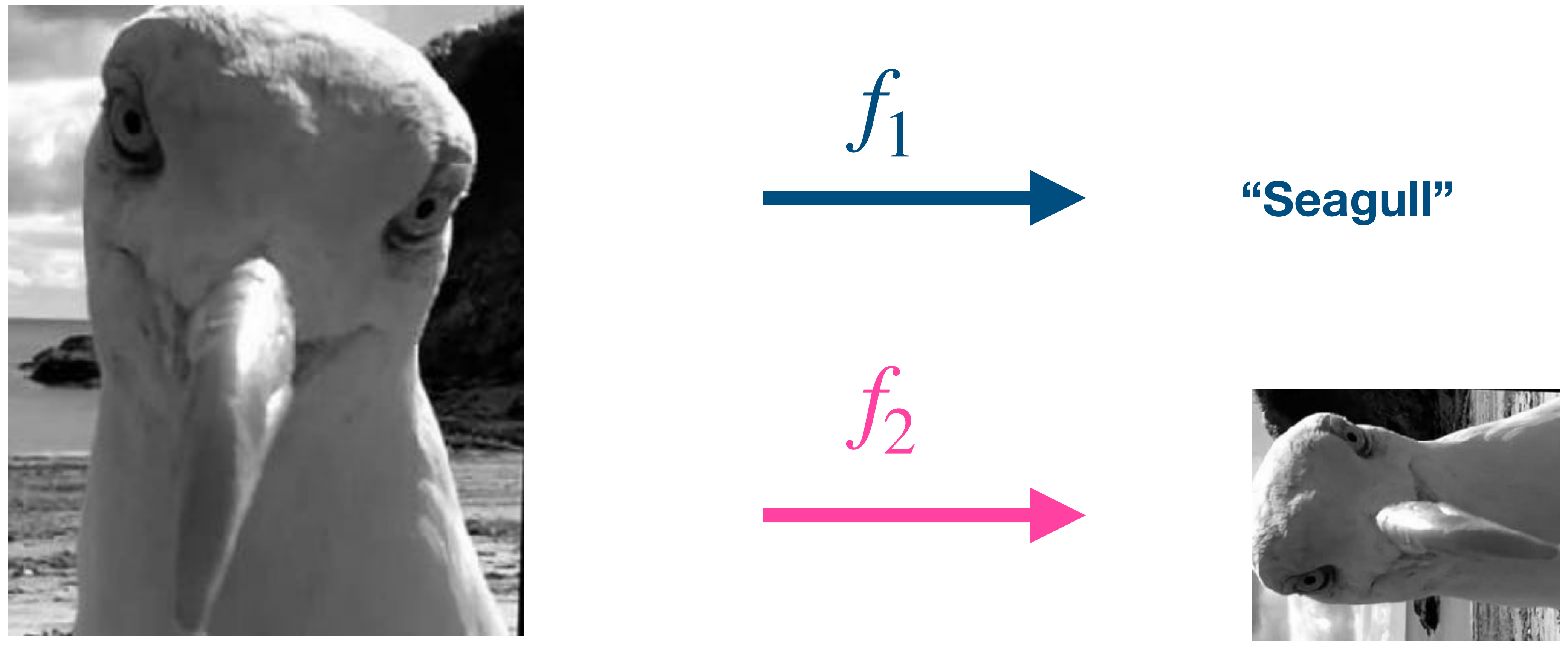
Range: set of possible outputs

Examples express transformations/relationships between numbers

Functions

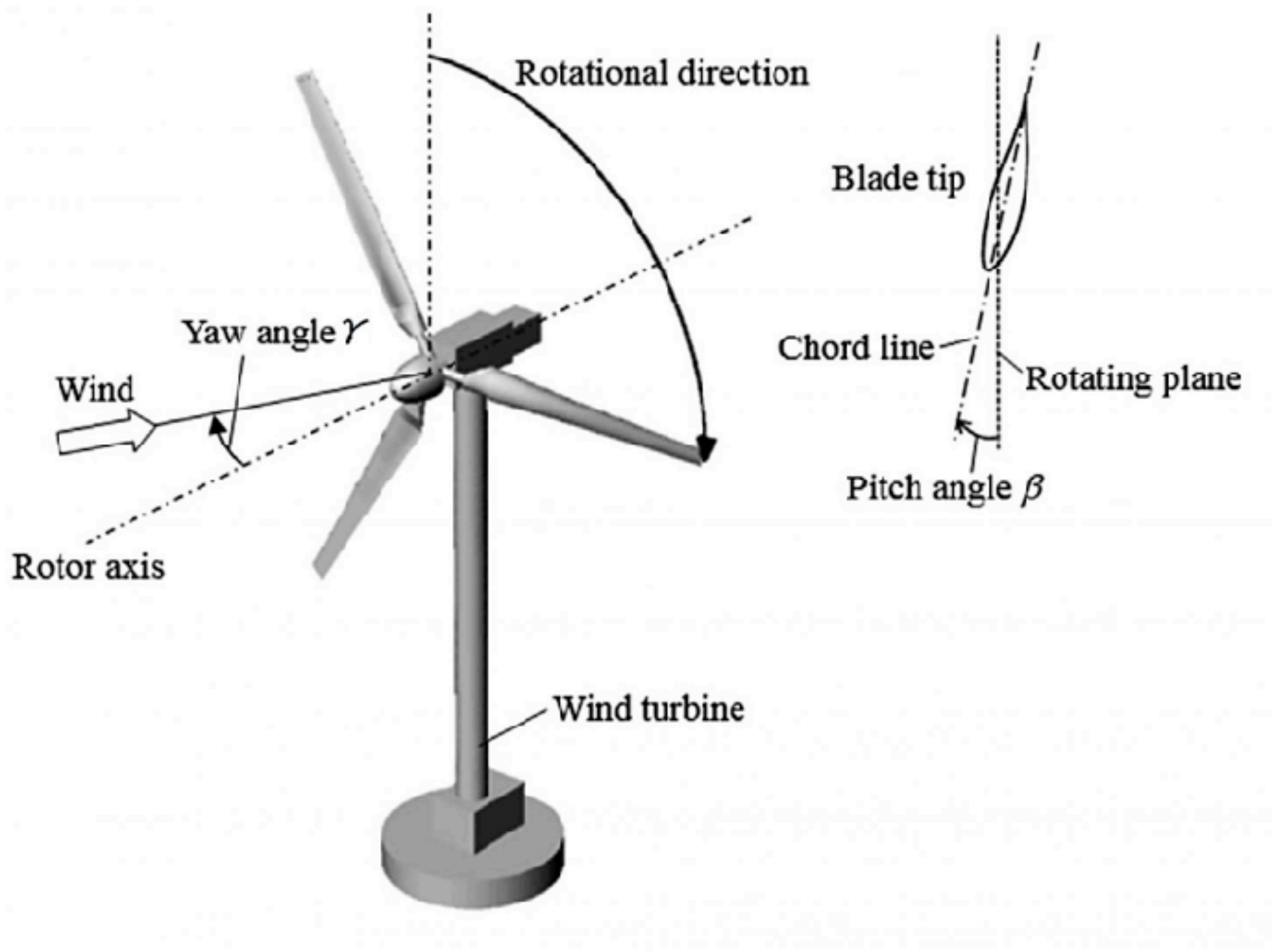
f_1 : images → strings (text)

f_2 : images → images



Real life: transformations / relationships between more complicated objects

Functions

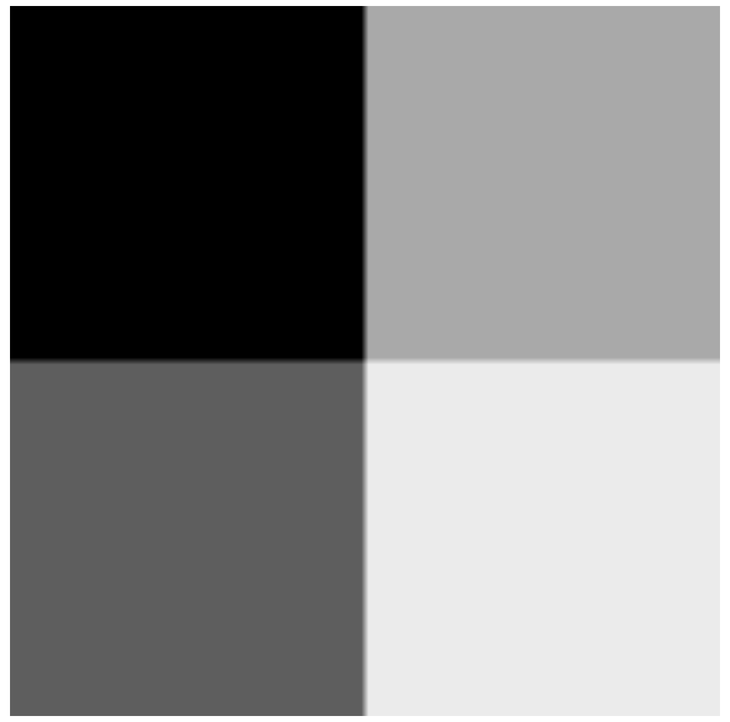


→ Best angle/pitch/yaw to catch the wind

Real life: transformations / relationships between more complicated objects

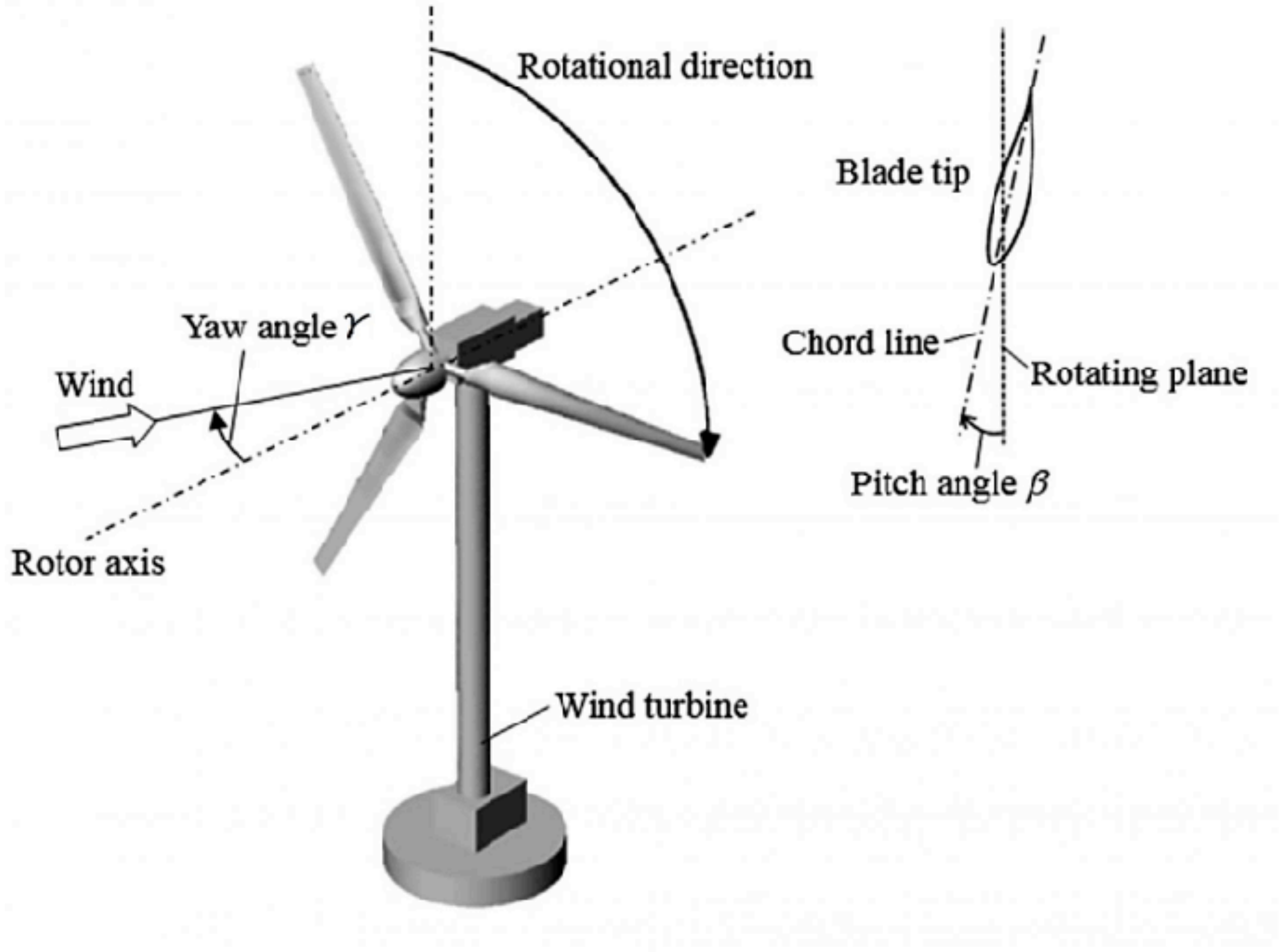
How do we do mathematics on ‘complicated objects’?

Complicated objects often boil down to ~~collections~~ arrays of numbers



$$\begin{bmatrix} 0.0 & 0.3 \\ 0.6 & 0.9 \end{bmatrix}$$

[**Lots of numbers!**]

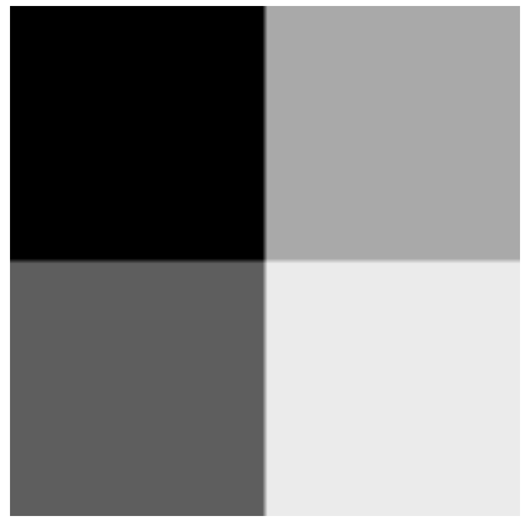


Windspeed,
Wind direction,
Motor torque,
Yaw angle
…

Arrays have a shape

Which we change as convenient!

Image



$$A = \begin{bmatrix} 0.0 & 0.3 \\ 0.6 & 0.9 \end{bmatrix} \qquad A^T = \begin{bmatrix} 0.0 & 0.6 \\ 0.3 & 0.9 \end{bmatrix} \qquad \mathrm{vec}(A^T) = \begin{bmatrix} 0.0 \\ 0.3 \\ 0.6 \\ 0.9 \end{bmatrix}$$

Numerical representation

0 is black,
1 is white

`numpy.shape(A)`

Transposition:

Rows

Columns

`numpy.transpose(A)`

Vectorisation:

Concatenate columns

`numpy.reshape(A, 4)`

Arrays have a **shape**

Which we change as convenient!

The number 4

A scalar: no shape

`numpy.shape(4)`

The array [4]

An array containing one element: the number 4

`numpy.shape([4])`

Not the same!!!

Shapes have a tensor dimension

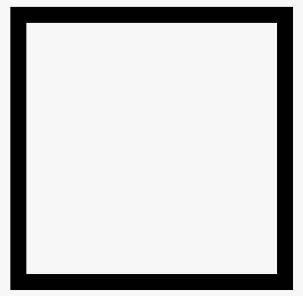
Vector (1d-array)

$$[0.0 \quad 0.3 \quad 0.6 \quad 0.9]$$

Matrix (2d-array)

$$\begin{bmatrix} 0.0 & 0.3 \\ 0.6 & 0.9 \end{bmatrix}$$

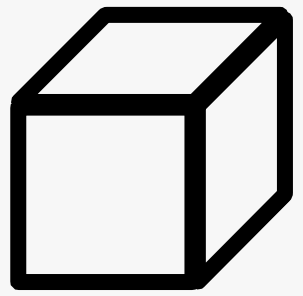
Rows and columns



3-Tensor (3d-array)

$$\left[\begin{bmatrix} 0.0 & 0.3 \\ 0.6 & 0.9 \end{bmatrix} \begin{bmatrix} 0.0 & 0.3 \\ 0.6 & 0.9 \end{bmatrix} \begin{bmatrix} 0.0 & 0.3 \\ 0.6 & 0.9 \end{bmatrix} \right]$$

Rows, columns,….layers?



`A[3,7,2]` *- accessing element in 3d array*

`numpy.reshape([4], (1,1,1,1,1))` *- how many dimensions?*

Shapes have a tensor dimension

Vector (1d-array)

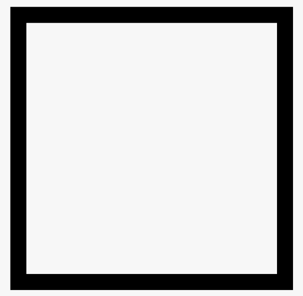
$$[0.0 \quad 0.3 \quad 0.6 \quad 0.9]$$

`A.shape = (4,)`

Matrix (2d-array)

$$\begin{bmatrix} 0.0 & 0.3 \\ 0.6 & 0.9 \end{bmatrix}$$

Rows and columns

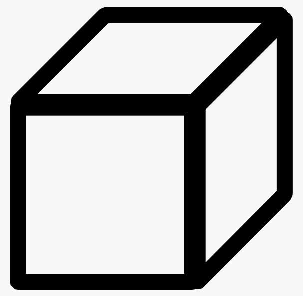


`A.shape = (2,2)`

3-Tensor (3d-array)

$$\left[\begin{bmatrix} 0.0 & 0.3 \\ 0.6 & 0.9 \end{bmatrix} \begin{bmatrix} 0.0 & 0.3 \\ 0.6 & 0.9 \end{bmatrix} \begin{bmatrix} 0.0 & 0.3 \\ 0.6 & 0.9 \end{bmatrix}\right]$$

Rows, columns,….layers?



`A.shape = (2,2,3)`

Dimension is: `len(A.shape)`

Confusing terminology alert

Vector (1d-array)

$$[0.0 \quad 0.3 \quad 0.6 \quad 0.9]$$

Python/Julia: - tensor/array dimension is 1
- length is 4

Mathematician: - It's a 4-dimensional vector.
- A vector is a 1d tensor (i.e. array)
- Vector's length depends on its entries

Array indexing alert

Avoid incredible Python frustration!!!

Maths/Julia/Common sense:

$$\begin{array}{ccc} & \text{Column 1} & \text{Column 2} \\ \begin{array}{c}\text{Row 1} \\ \text{Row 2}\end{array} & \multicolumn{2}{c}{\begin{bmatrix} 0.0 & 0.3 \\ 0.6 & 0.9 \end{bmatrix}} \end{array}$$

$$A_{21} = 0.6$$

Python:

$$\begin{array}{ccc} & \text{Column 0} & \text{Column 1} \\ \begin{array}{c}\text{Row 0} \\ \text{Row 1}\end{array} & \multicolumn{2}{c}{\begin{bmatrix} 0.0 & 0.3 \\ 0.6 & 0.9 \end{bmatrix}} \end{array}$$

`A[1,0] = 0.6`

Ingredients to do maths on arrays?

Let’s look back to how we do maths on numbers!

Addition/subtraction

is straightforward!

- *Only makes sense on vectors with same shape*
- *Otherwise, just like for numbers!*

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} + \begin{bmatrix} w \\ x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a+w \\ b+x \\ c+y \\ d+z \end{bmatrix}$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} - \begin{bmatrix} w & x \\ y & z \end{bmatrix} = \begin{bmatrix} a-w & b-x \\ c-y & d-z \end{bmatrix}$$

Scaling

is straightforward!

- *Multiply an array (of any shape) by a scalar (number / field element)*

$$x \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} x * a \\ x * b \\ x * c \\ x * d \end{bmatrix}$$

$$-3 \begin{bmatrix} 1 \\ 4 \\ -2 \end{bmatrix} = \begin{bmatrix} -3 \\ -12 \\ 6 \end{bmatrix}$$

Dot product

…also called inner product

$$\underline{a} = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$$

$$\underline{x} = \begin{bmatrix} w \\ x \\ y \\ z \end{bmatrix}$$

$$a^T x = [a, b, c, d] \begin{bmatrix} w \\ x \\ y \\ z \end{bmatrix}$$

$$= aw + bx + cy + dz \in \mathbb{R}$$

Also denoted

$$\langle \underline{a}, \underline{x} \rangle$$

$$\underline{a} \bullet \underline{x}$$

Pause

In Marimo: build two vectors and two matrices.

Take the dot products of the vectors. Subtract the matrices from each other

What's the point of a matrix?

Two ways to look at a matrix

A 2x3 matrix

$$\begin{bmatrix} 4 & 2 & 5 \\ -3 & 6 & 1 \end{bmatrix}$$

- *Two row vectors*

$$\left[\begin{array}{c:c:c} 4 & 2 & 5 \\ -3 & 6 & 1 \end{array}\right]$$

- *Three column vectors*

Matrix multiplication transforms vectors

$$\begin{bmatrix} 4 & 2 & 5 \\ -3 & 6 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 4 \\ 3 \end{bmatrix} = ?$$

Do this in Marimo now

Matrix multiplication transforms vectors

Inner product of each row vector in the matrix, with the vector

$$\begin{bmatrix} 4 & 2 & 5 \\ -3 & 6 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 4 \\ 3 \end{bmatrix} = \begin{bmatrix} 8+8+15 \\ -6+24+3 \end{bmatrix}$$

- ***Requirement:*** *number of matrix columns = length of vector*
- ***Outcome:*** *length of new vector = number of rows in matrix*

Alternative

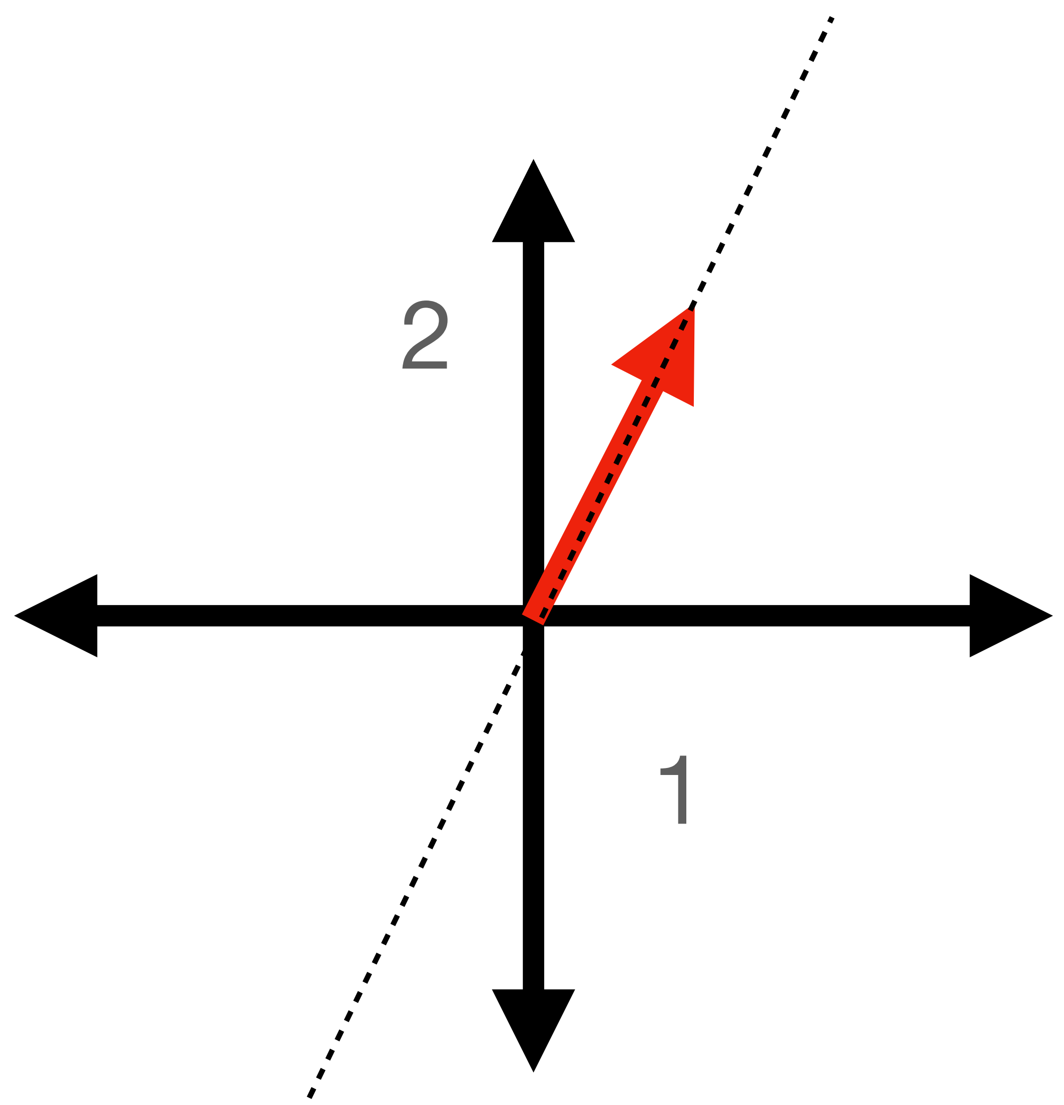
$$\begin{bmatrix} 4 & 2 & 5 \\ -3 & 6 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 4 \\ 3 \end{bmatrix} = 2\begin{bmatrix} 4 \\ -3 \end{bmatrix} + 4\begin{bmatrix} 2 \\ 6 \end{bmatrix} + 3\begin{bmatrix} 5 \\ 1 \end{bmatrix}$$

- ***Requirement:*** *number of matrix columns = length of vector*
- ***Outcome:*** *length of new vector = number of rows in matrix*

What can you say about this matrix?

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

All input vectors are pushed onto this line



$$A\underline{v} \propto \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

Matrices transform vectors

…by matrix multiplication

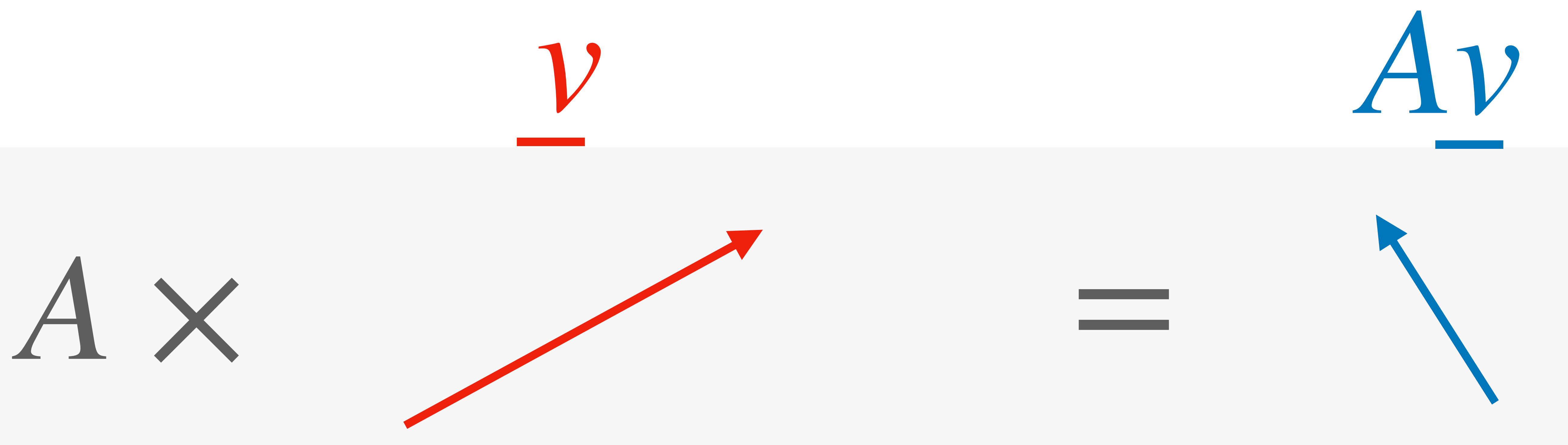
Use np.matmul to transform your vector by your matrix in marimo

$$f(\underline{v}) = (A\underline{v})$$

Vector

Vector

Matrix



Can change length,
direction, even shape
(eg 2-vector -> 5 vector)

Square matrix properties

$$\begin{bmatrix} 4 & 2 & 5 \\ -3 & 6 & 1 \\ 1 & 3 & -2 \end{bmatrix} \begin{bmatrix} 2 \\ 4 \\ 3 \end{bmatrix} = \begin{bmatrix} \bullet \\ \bullet \\ \bullet \end{bmatrix}$$

Input and output vectors have same shape

When can we multiply matrices?

Equal! So can multiply them

Number of columns

$$\begin{bmatrix} 4 & 2 & 5 \\ -3 & 6 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 4 & -2 \\ 3 & -3 \end{bmatrix}$$

Number of rows

How to multiply matrices

a) Figure out the shape of the output

(m x n) matrix multiplied by (n x p) matrix = (m x p) matrix

$$\begin{array}{cc} & \text{n=3 columns} \\ \text{m=2 rows} & \begin{bmatrix} 4 & 2 & 5 \\ -3 & 6 & 1 \end{bmatrix} \end{array} \quad \begin{array}{cc} \text{p=2 rows} & \\ \begin{bmatrix} 2 & 1 \\ 4 & -2 \\ 3 & -3 \end{bmatrix} & \text{n=3 rows} \end{array}$$

***Requirement**: Number of columns (A) = Number of rows (B)*

How to multiply matrices

a) Figure out the *shape* of the output

$$\overset{A}{\begin{bmatrix} 4 & 2 & 5 \\ -3 & 6 & 1 \end{bmatrix}} \overset{B}{\begin{bmatrix} 2 & 1 \\ 4 & -2 \\ 3 & -3 \end{bmatrix}} = \overset{C}{\begin{bmatrix} \bullet & \bullet \\ \bullet & \bullet \end{bmatrix}}$$

***Outcome**: Number of rows (A) x Number of columns (B)*

C_{ij} *Depends on i^{th} row and j^{th} column*

How to multiply matrices

b) Inner product of each row vector of A with each column vector of B

$$\underset{A}{\begin{bmatrix} 4 & 2 & 5 \\ -3 & 6 & 1 \end{bmatrix}} \underset{B}{\begin{bmatrix} 2 & 1 \\ 4 & -2 \\ 3 & -3 \end{bmatrix}} = \underset{C}{\begin{bmatrix} (8+8+15) & (4-4-15) \\ (-6+24+3) & (-3-12-3) \end{bmatrix}}$$

$$= \begin{bmatrix} 31 & -15 \\ 21 & -18 \end{bmatrix}$$

$$C_{ij} = \langle A_{i,\bullet}, B_{\bullet,j} \rangle$$

i^{th} row

j^{th} column

How to multiply matrices

Question: Is this allowable?

$$[2,3]\begin{bmatrix} 4 & 2 & 5 \\ -3 & 6 & 1 \end{bmatrix} = [\,\bullet\,,\,\bullet\,,\,\bullet\,]$$

How to multiply matrices

Verify yourself:

$$\left([2,3] \begin{bmatrix} 4 & 2 & 5 \\ -3 & 6 & 1 \end{bmatrix} \right)^T = \begin{bmatrix} 4 & -3 \\ 2 & 6 \\ 5 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

Law of matrix transposition

$$\left(A\underline{v}\right)^T = \underline{v}^T A^T$$

Transposition swaps the multiplication order

Important: non-commutativity of multiplication

i.e. $AB \neq BA$ *unlike scalars*

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix} = ?$$

$$\begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = ?$$

Different row/column vectors!

The identity matrix

I_n **is an** $(n \times n)$ **matrix with ones on the diagonal**

$$I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad I_n = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & 0 & \vdots \\ \vdots & 0 & \ddots & \vdots \\ 0 & \dots & \dots & 1 \end{bmatrix}$$

Exercise:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix} = ? \qquad \begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = ?$$

The identity matrix

I_n **is an** $(n \times n)$ **matrix with ones on the diagonal**

$$AI = IA = A$$

- analogous to the number 1 in scalar multiplication

$$x(1) = 1(x) = x$$

The scalar inverse

$$f(x) = 4x$$

$$g(x) = \frac{1}{4}x$$

$$\Rightarrow \quad g \circ f(x) = \frac{1}{4}(4x) = 1x = x$$

$\frac{1}{4}$ *is the inverse of* 4

$$\Updownarrow$$

$f \circ g$ *is the identity function:* $f \circ g(x) = 1(x)$

The matrix inverse (for square matrices)

```
numpy.linalg.inv(A)
```

Inverse of A is A^{-1} implies:

$$AA^{-1} = A^{-1}A = I$$
$$\Rightarrow AA^{-1}\underline{x} = \underline{x}$$

Example (verify yourself)

$$\begin{bmatrix} -2.5 & 1.5 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix} = \begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix} \begin{bmatrix} -2.5 & 1.5 \\ 2 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

The matrix inverse

…is useful

Problem: find a, b

$$\begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 7 \\ 13 \end{bmatrix}$$

Cancels to identity!

$$\boxed{\begin{bmatrix} -2.5 & 1.5 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix}} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} -2.5 & 1.5 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} 7 \\ 13 \end{bmatrix}$$

$$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} -2.5 & 1.5 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} 7 \\ 13 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

The matrix inverse

…is useful

General problem: find $\underline{x}$, given $\underline{y}$

$$A\underline{x} = \underline{y}$$

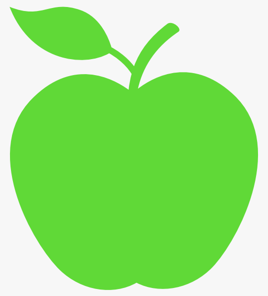
$$\Rightarrow A^{-1}A\underline{x} = \underline{x} = A^{-1}\underline{y}$$

This is called a matrix equation

The matrix inverse

toy application

$$\overset{A}{\begin{bmatrix} £3.00/kg & £2.00/kg \\ 1 & 1 \end{bmatrix}} \overset{\underline{x}}{\begin{bmatrix} a\ kg \\ b\ kg \end{bmatrix}} = \overset{\underline{y}}{\begin{bmatrix} £12 \\ 5kg \end{bmatrix}}$$



$= £3.00/kg$

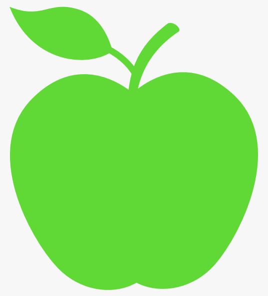
$= £2.00/kg$

holds $5kg$

- *Number of rows of A is number of constraints*
- *Number of columns of A is number of free variables*

The matrix inverse

toy application



$= £3.00/kg$

$= £2.00/kg$

holds $5kg$

$$\begin{array}{ccccccc} \underline{x} & & A^{-1} & \underline{y} & & \\ \begin{bmatrix} a\ kg \\ b\ kg \end{bmatrix} & = & \begin{bmatrix} 1\ kg/£ & -2 \\ -1\ kg/£ & 3 \end{bmatrix} & \begin{bmatrix} £12 \\ 5kg \end{bmatrix} & = & \begin{bmatrix} 2\ kg \\ 3\ kg \end{bmatrix} \end{array}$$

- *Number of rows of A is number of constraints*
- *Number of columns of A is number of free variables*

The matrix inverse

…doesn't always exist!

$$A\underline{x} = \underline{y}$$

$$\Rightarrow A^{-1}A\underline{x} = \underline{x} = A^{-1}\underline{y}$$

Proof by contradiction: inverse can't exist

$$\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} 2 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 2 \\ -1 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \end{bmatrix} ?$$ …but $$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

The matrix inverse

…doesn’t always exist!

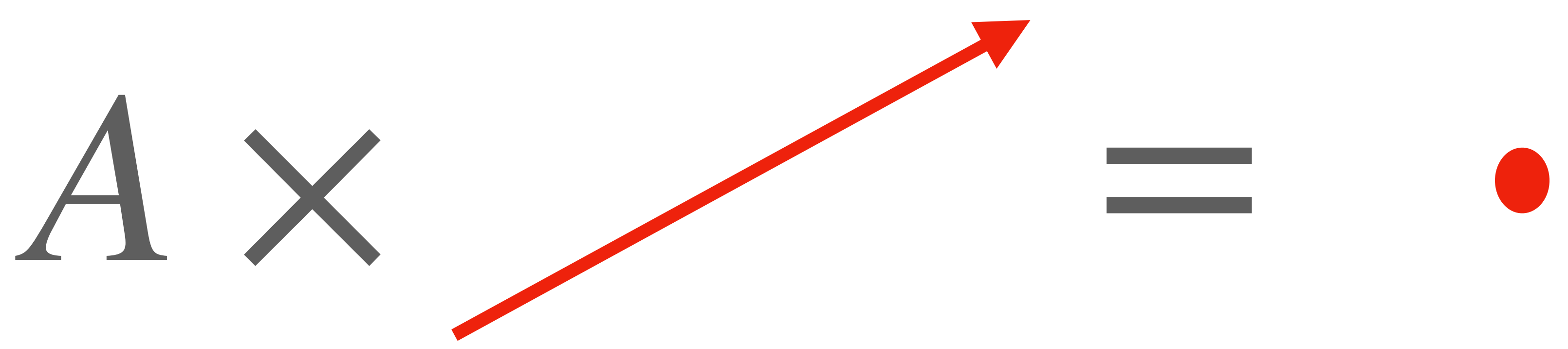
Square matrices

Invertible matrices

$$A\underline{x} = \underline{0} \Rightarrow \underline{x} = \underline{0}$$

Non-invertible matrices

$A\underline{x} = \underline{0}$ for nonzero $\underline{x}$



Important note on array algebra

$$A\underline{x} = \underline{y}$$

$$\Rightarrow A^{-1}A\underline{x} = \underline{x} = A^{-1}\underline{y}$$

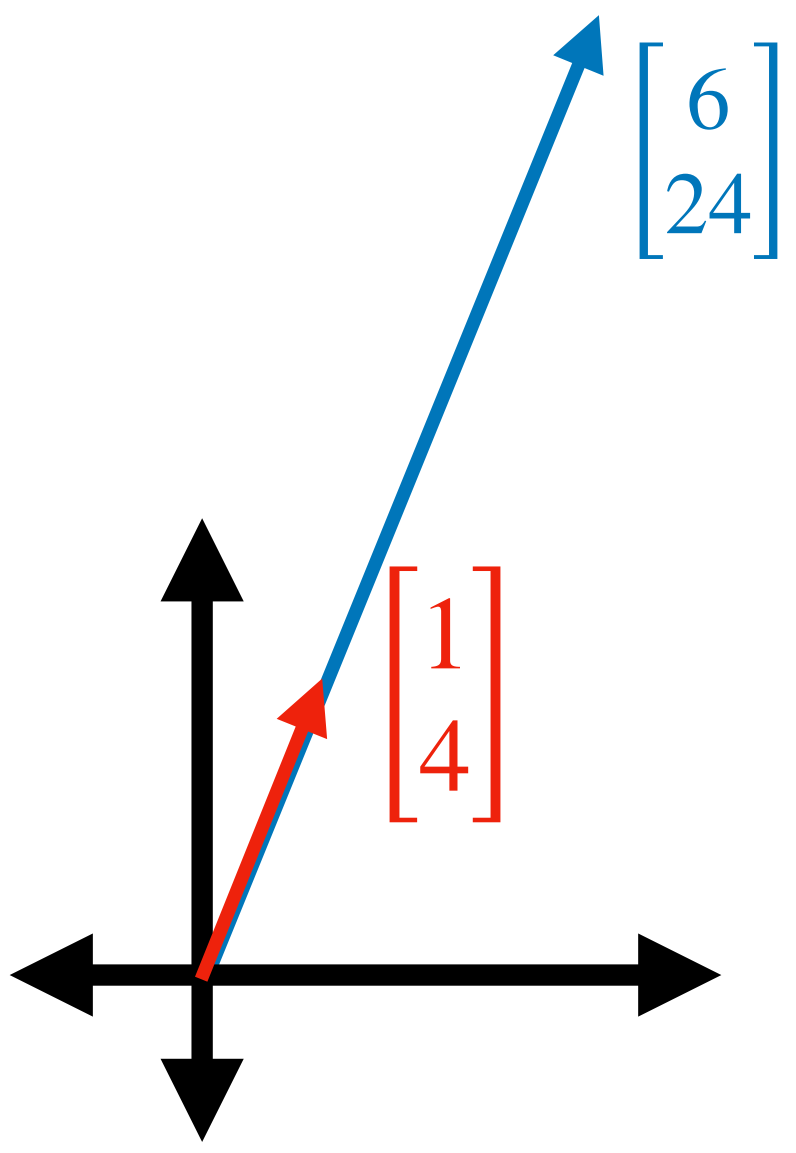
$$A^{-1}A\underline{x} = \underline{x} \neq A\underline{x}A^{-1}$$

Equation manipulation:

- *Can do same thing to both sides of the equation (like scalar algebra)*
- *Except, left multiplication and right multiplication are different*

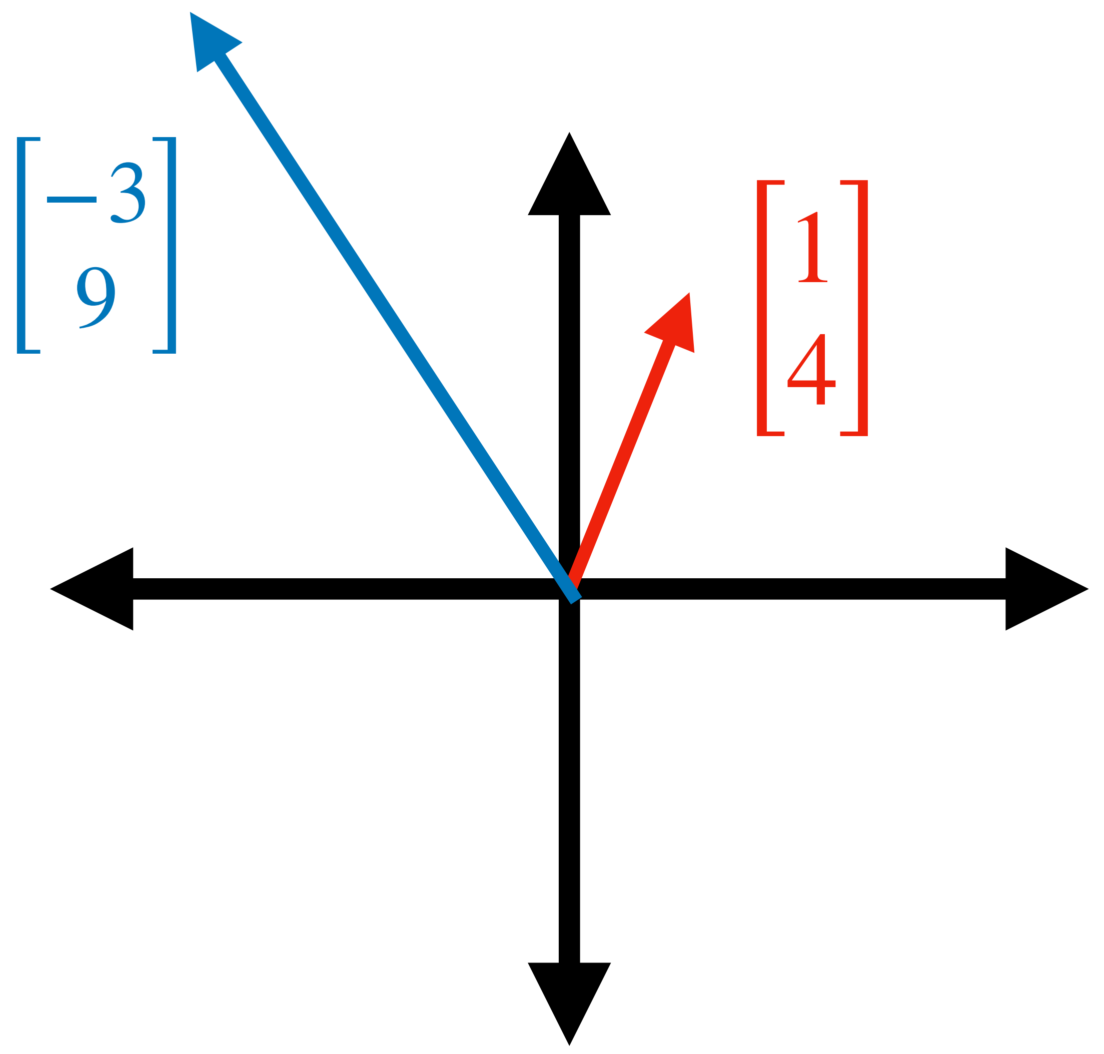
Worked examples

$$\begin{bmatrix} -6 & 3 \\ 4 & 5 \end{bmatrix} \begin{bmatrix} 1 \\ 4 \end{bmatrix} = ? \qquad = \begin{bmatrix} 6 \\ 24 \end{bmatrix} = 6 \begin{bmatrix} 1 \\ 4 \end{bmatrix}$$



- *Pure scaling, no rotation*

$$\begin{bmatrix} -6 & 3 \\ 4 & 5 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = ? \qquad = \begin{bmatrix} -3 \\ 9 \end{bmatrix}$$

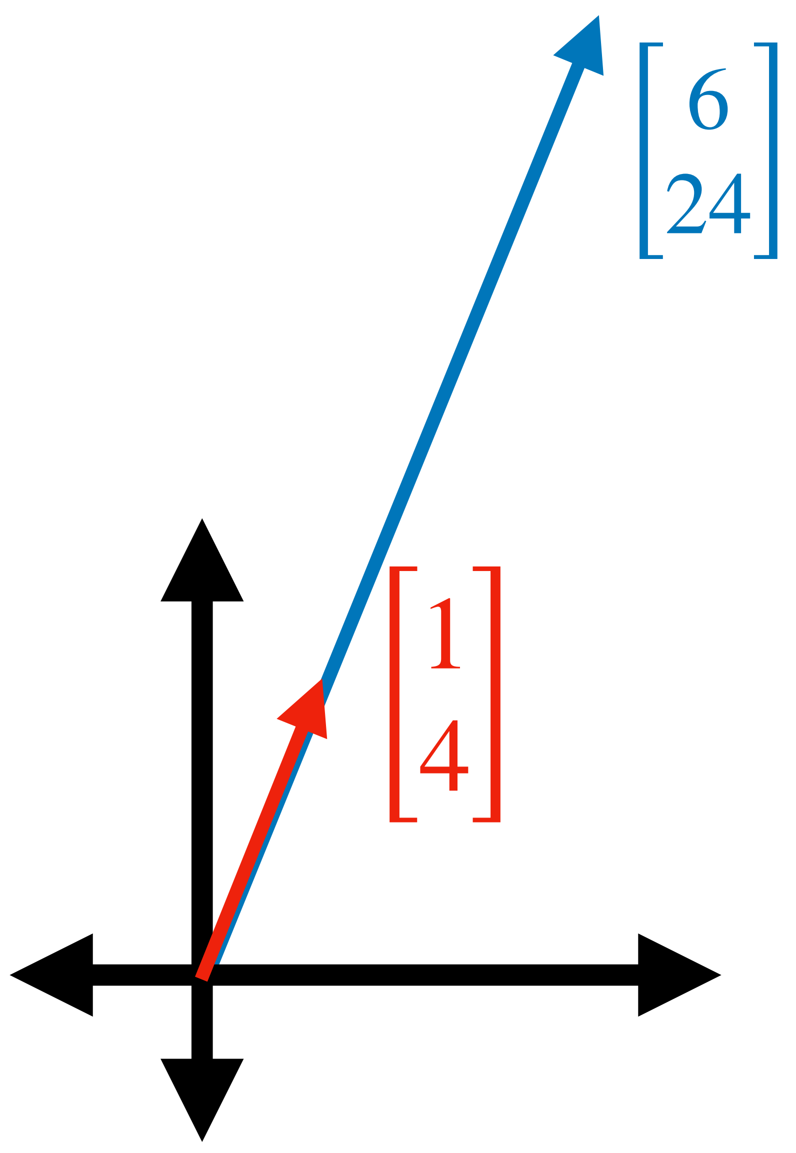


- *Scaling and rotation*

Worked examples

$$\begin{bmatrix} -6 & 3 \\ 4 & 5 \end{bmatrix} \begin{bmatrix} 1 \\ 4 \end{bmatrix} = ? \qquad = \begin{bmatrix} 6 \\ 24 \end{bmatrix} = 6 \begin{bmatrix} 1 \\ 4 \end{bmatrix}$$

- *Pure scaling, no rotation*



$$A\underline{v} = \lambda \underline{v}$$

Eigenvalue

Eigenvector

- *Eigenvectors of a matrix are those that don't rotate under transformation*

```
numpy.linalg.eig(A)
```

Worked examples

Exercise: find eigenvectors of

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} d_1 & 0 \\ 0 & d_2 \end{bmatrix}$$

Worked examples

Exercise: find eigenvectors of

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} d_1 & 0 \\ 0 & d_2 \end{bmatrix}$$

Everything! $I\underline{v} = \underline{v}$

$$\begin{bmatrix} a \\ 0 \end{bmatrix} \quad \begin{bmatrix} 0 \\ b \end{bmatrix}$$