

TEORIA

VECTORES

Vector: An ordered list of numbers

Dimensionality: the number of numbers
(elements)

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \neq \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix}$$

↑ ↑
"Column Vector" "Row Vector"

$$\begin{bmatrix} 1 & 3 & 2 & 5 & 4 \end{bmatrix} \rightarrow \text{"Row Vector"}$$
$$\begin{bmatrix} -1 & -1 & 4 \end{bmatrix} \rightarrow$$

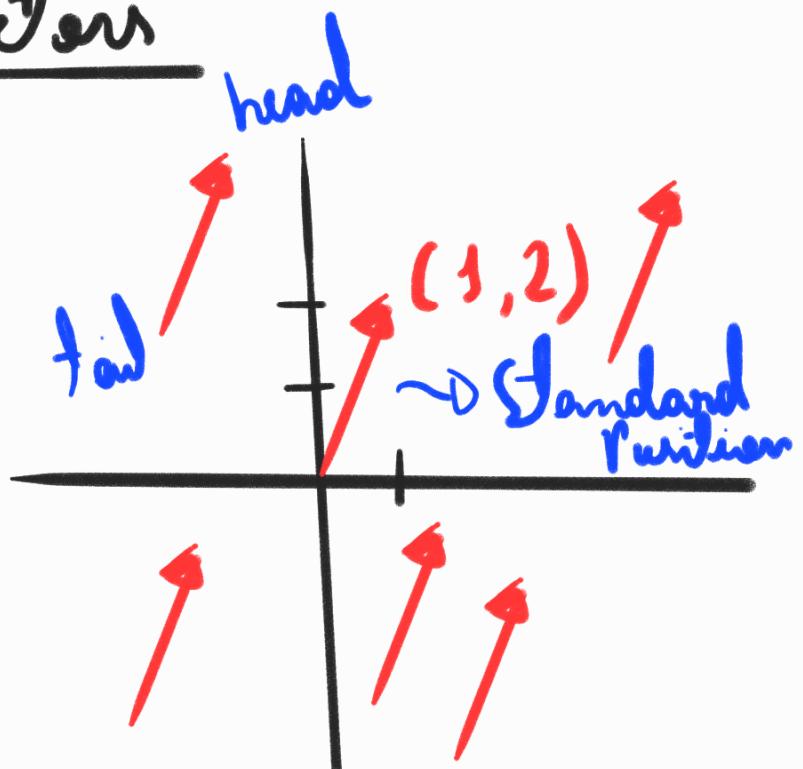
Vector Notation

$$\begin{bmatrix} 1 \\ 0 \\ \pi \\ 5 \\ -2 \end{bmatrix} = V \text{ or } \vec{v} \text{ or } \overrightarrow{v} \text{ or } V$$

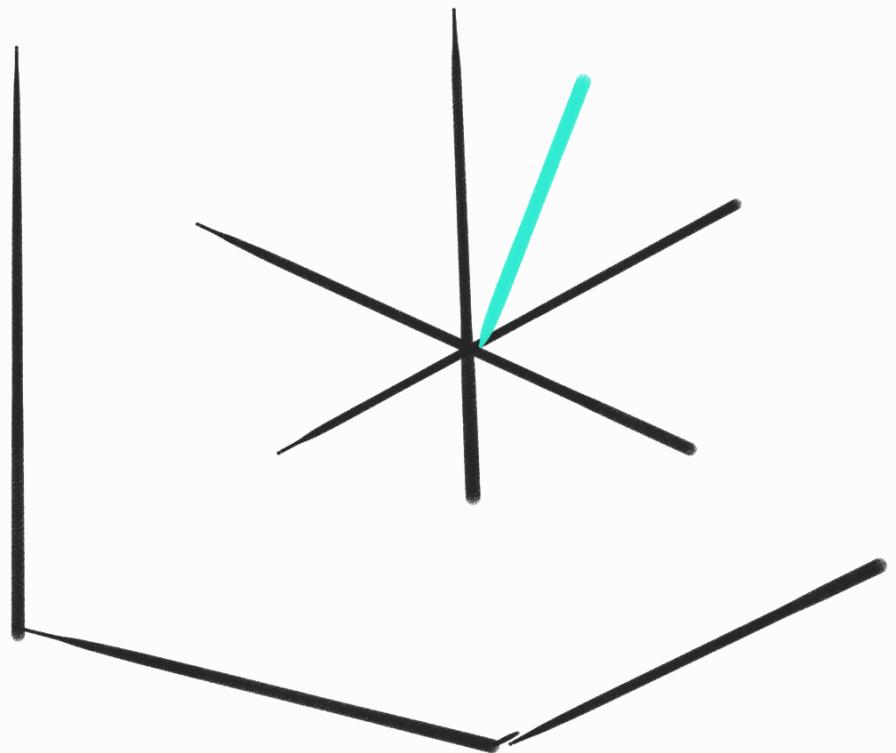
→ I have written

Geometric Vectors

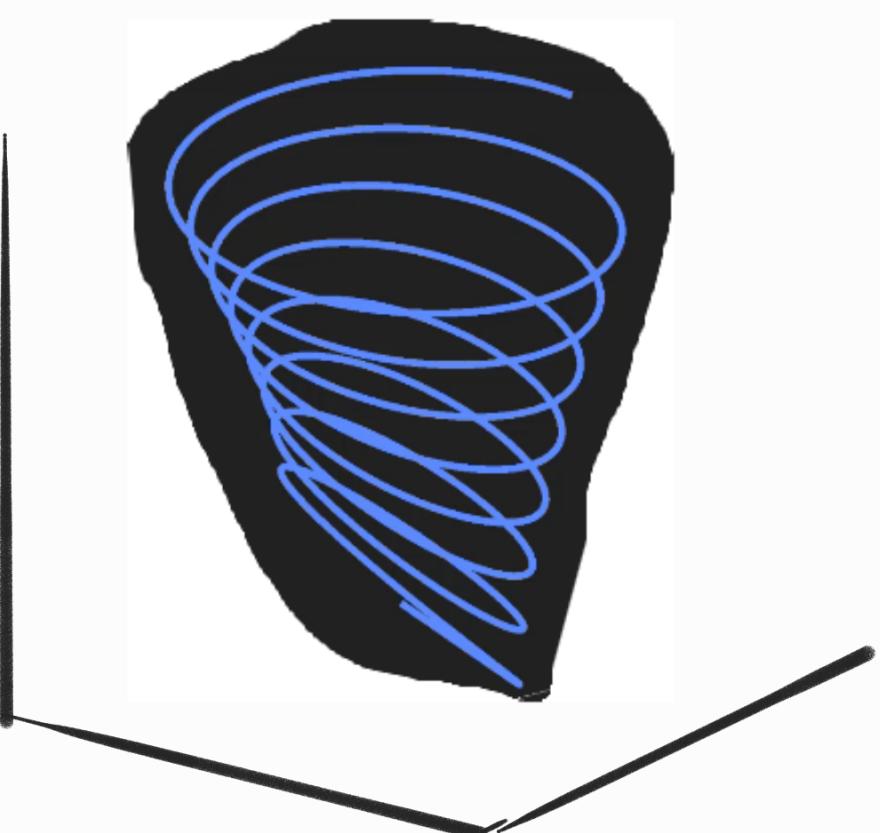
$$[1 \ 2]$$



$$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$$



$$\begin{bmatrix} \sin(x) \\ x\cos(x) \\ x \end{bmatrix}$$



Creating Vectors in Python

$V_2 = [3, -2]$ (2D) $x = 3, y = -2$

$V_3 = [4, -3, 2]$ (3D) $x = 4, y = -3, z = 2$

Transpose (Row to Column (or vice-versa))

$V_3^T = np.transpose(V_3)$

Creating Graphs

2D

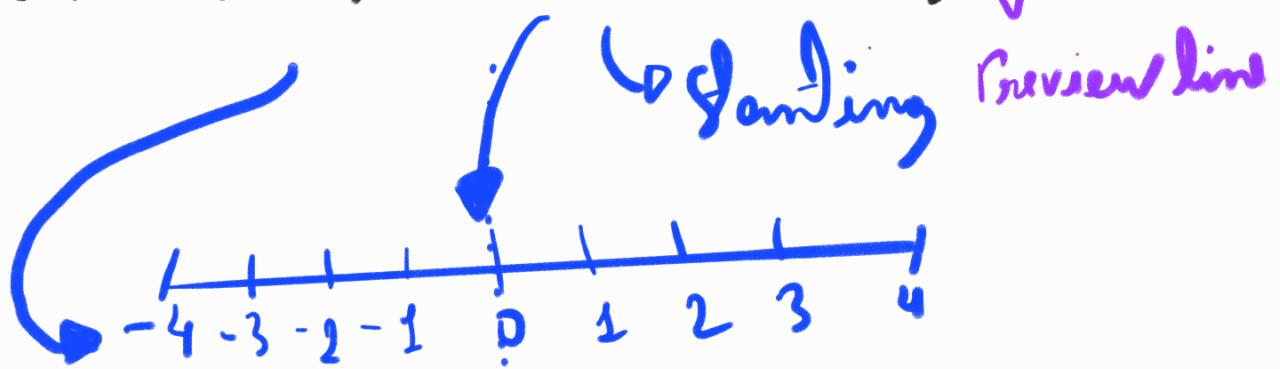
`plt.plot([0, V2[0]], [0, V2[1]])`

: It starts at the origin

`plt.axis('equal')`

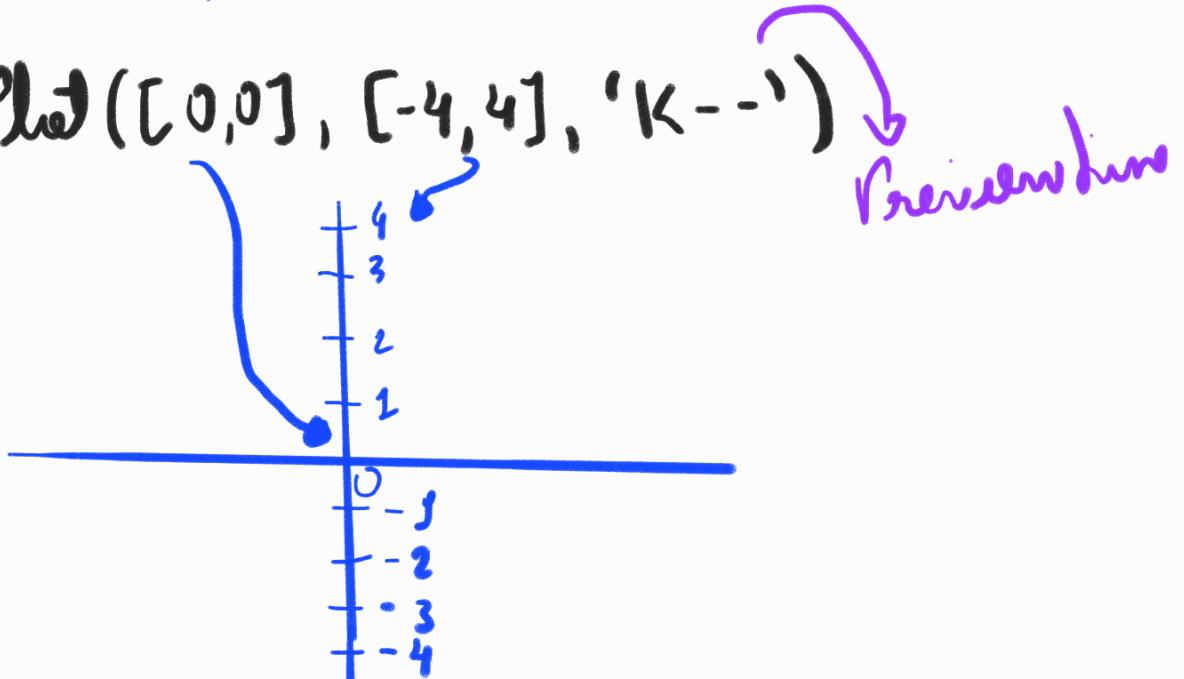
Ensures that the scale of the X-axis is equal to that of the Y-axis.

`PLT.Plot([-4,4],[0,0],'K--')`



this draws a horizontal line starting -4 to 4
on the X-axis, Perpendicular at 0 on the Y-axis

`PLT.Plot([0,0],[-4,4],'K--')`



`PLT.grid()` → Add lines.

→ facilitates the reading of coordinates

→ this command turns on the background grid lines

```
plt.axis((-4, 4, -4, 4))
```

This sets the boundaries of your graph's view.
It tells Python exactly where to "sit" the camera
frame.

(x-min, x-max, y-min, y-max)

