VECTORS

- **Vector**: ordered finite list of numbers. Elements (entries, coefficients, components) are values of array.
  - Size (dimension / length) = no. of elements it contains.
  - n-vector: vector of size 'n' (1-vector is same as the no. itself)
  - a_i → symbol used to represent the ith (runs from 1−n) element of array a.

- **Real Vector**: vector in which all scalars are real numbers.
  ↳ set of real no. = R ↳ set of real n-vectors = R^n ($\therefore a \in R^n$)

- **Stacked Vector**: formed by concatenating 2 or more vectors. $a = (b, c, d)$ where $a, b, c, d \in R^n$.
  → They can include scalars. (a = 3-vector ; then $b = (1, a)$ = 4-vector)

- **Sub vectors**: a = vetor. ; $\boxed{a_{r:s} = \text{vector of size } s-r+1 \text{, with entries } a_r \dots a_s.}$
  ↑ index range

  → $(a_i)_j$ = jth element in the a_i vector of the (ith element (vector) in a)

- **Zero Vector**: vector with all elements equal to 0. (0_n = zero vector of size n)

Overloading: when we use the same symbol to represent different things (0)

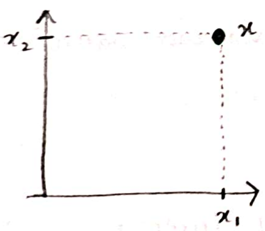
- **Unit Vector**: all elements are equal to 0, except 1 element.

$$(e_i)_j = \begin{cases} 1 & j = i \\ 0 & j \neq i \end{cases} \quad \text{Ex: } e_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad e_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad e_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

- **Ones Vector**: 1_n = vector of size n with all elements as 1.

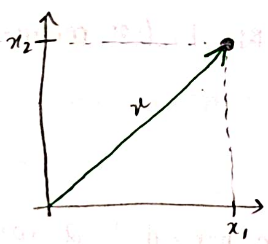
SPARSITY: vector is sparse if ~~any~~ many of its entries are zero.
  * Sparsity pattern = set of indices of non-zero entries.
  = $nnz(x)$ non-zero entries of n-vector x

Ex:



This is a 2-vector representing a point or position in 2D plane

POSITION



This is a 2-vector x which is showing displacement in 2D.

DISPLACEMENT

- Vectors can represent colors (RGB for ex)
  Red = (1, 0, 0) ; Blue = (0, 0, 1)
- Quantities : shopping list = [1, 3, 5, 1, 2, 3] (bread ↓ 1, eggs ↓ 5)
- portfolio : (100, 50, 20)
  - 100 shares of asset 1.
  - 50 shares of asset 2.... so on.
- Values across population: (BP of n-patients)

- Proportions : vector 'w' which outcomes of n choices, where w_i as fraction for choice of i.
  → All entries are non-negative and add upto 1.

- Daily Returns : of a stock (its fractional ↑ or ↓).
  Ex: (−0.022, +0.014, +0.004) ← when down by 2.2% on first day went up by 1.4% the next & so on.

- **Time-series:**
  - Value of some quantity at different times.
  - The entries in such a vector are called 'samples'.

Other examples → images, videos, word count & histogram, customer purchases, attributes, etc.

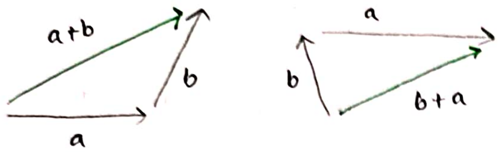
Vector Addition:

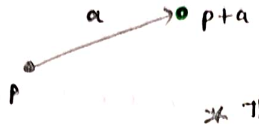
→ only 2 vectors of the same size can be added by adding corresponding elements.

→ vector subtraction is called the difference of vectors.

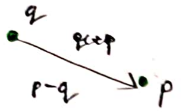
Properties:
1. commutative : $a+b = b+a$
2. Associative : $(a+b)+c = a+(b+c)$
3. $a+0 = 0+a = a$
4. $a-a = 0$

zero vector of size same as a.





Here, p represents a position.
a represents a displacement.
* Then $p+a$ = position of p displaced by a.



This represents the displacement from the point q to point p.

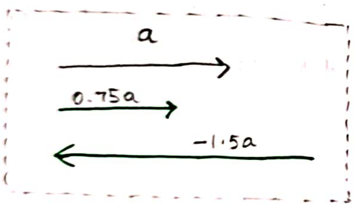
Scalar-vector Multiplication:

→ This is done by multiplying every element of the vector by the scalar value.
(In this book, scalars are represented by $(\alpha, \beta, \gamma \ldots)$ & vectors are represented with $(a, b, c \ldots)$)

Properties:
1. commutative $(\alpha a = a\alpha)$
2. associative $((\alpha\beta)a = \alpha(\beta a))$
3. Distributive. $(\alpha+\beta)a = \alpha a + \beta a$

$a(\beta+\alpha) = \beta a + \alpha a$

$\alpha(a+b) = \alpha a + \alpha b$



- In audio scaling βa = a with loudness ↑ of β.
- Linear combinations: $\beta_1 a_1 + \beta_2 a_2 + \ldots + \beta_m a_m$. ($\beta_1, \beta_2 \ldots$ are coefficients)
- Linear combination of unit vectors $= \begin{bmatrix} 1 \\ -3 \\ 5 \end{bmatrix} = 1\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + -3\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + 5\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$

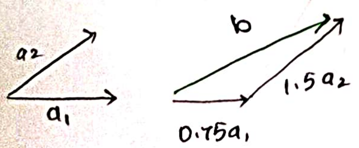
- weighted avg = occurs when $\beta_i \geq 0$ and $\sum \beta_i = 1$ (or mixture / convex combination)
- affine combination = when $\sum \beta_i = 1$ (these coefficients can sometimes be given as %)

Ex: • Replicating Cash flow:

$c_1 = (1, -1.1, 0)$ is a \$1 loan from period 1-2 with 10% interest.

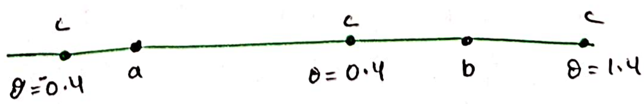
$c_2 = (0, 1, -1.1)$ is a \$1 loan from period 2-3 with 10% interest.

$d = c_1 + 1.1c_2 = (1, 0, -1.21)$ is a 2 period loan with 10% CI.



$b = 0.75a_1 + 1.5a_2$

- Line and Segment: $c = (1-\theta)a + \theta b$ (affine combination) describes point on line passing through a, b.



Inner Product:

The (standard) inner product (also called dot prodult) of 2 n-vectors is defined as:

$$\boxed{a^T b = a_1 b_1 + a_2 b_2 + a_3 c_3 + \cdots a_n b_n}$$

Properties: 1. commutative: $a^T b = b^T a$

2. Associative with scalar mul: $\gamma a^T b = (\gamma a^T) b = \gamma (a^T b)$

3. Distributive with vector add: $(a+b)^T c = a^T c + b^T c$.

$$(a+b)^T (c+d) = a^T c + a^T d + b^T c + b^T d.$$

Unit vector: $e_i^T a$

Sum: $1^T a$

Average: $(1/n)^T a$

powers: $a^T a$

selective sum: $b \rightarrow$ vector with 0/1.

$b^T a$ = such of elements in a where $b_i = 1$

- Block Vectors: $a^T b = \begin{bmatrix} a_1 \\ \vdots \\ a_K \end{bmatrix}^T \begin{bmatrix} b_1 \\ \vdots \\ b_K \end{bmatrix} = a_1^T b_1 + a_2^T b_2 + \cdots + a_K^T b_K$

Complexity:

- computers store real numbers in floating point format (decimals)

→ basic operations on these (like add, sub, ...) are called flops.

→ complexity of algorithm = no. of flops needed as function of input dim.

Floating point round of errors: very small error in computed result of flops.
(ways to mitigate this is studied in numerical analysis)

→ speed with which computer can carry out flops is called gflops. (gigaflops per second)
(10^9 flops/sec)

→ crude approx = flops needed / computer speed.

So... $x+y$ needs n additions $\longrightarrow$ n flops

$x^T y$ needs n multiplications, n-1 additions = 2n-1 flops

we simplify this to 2n (or even n) & much less if it is sparse.

$$a^T x = \sum_{i=4,8,12,16,20} x_i \quad \sum_{i=7,14,21} x_i$$