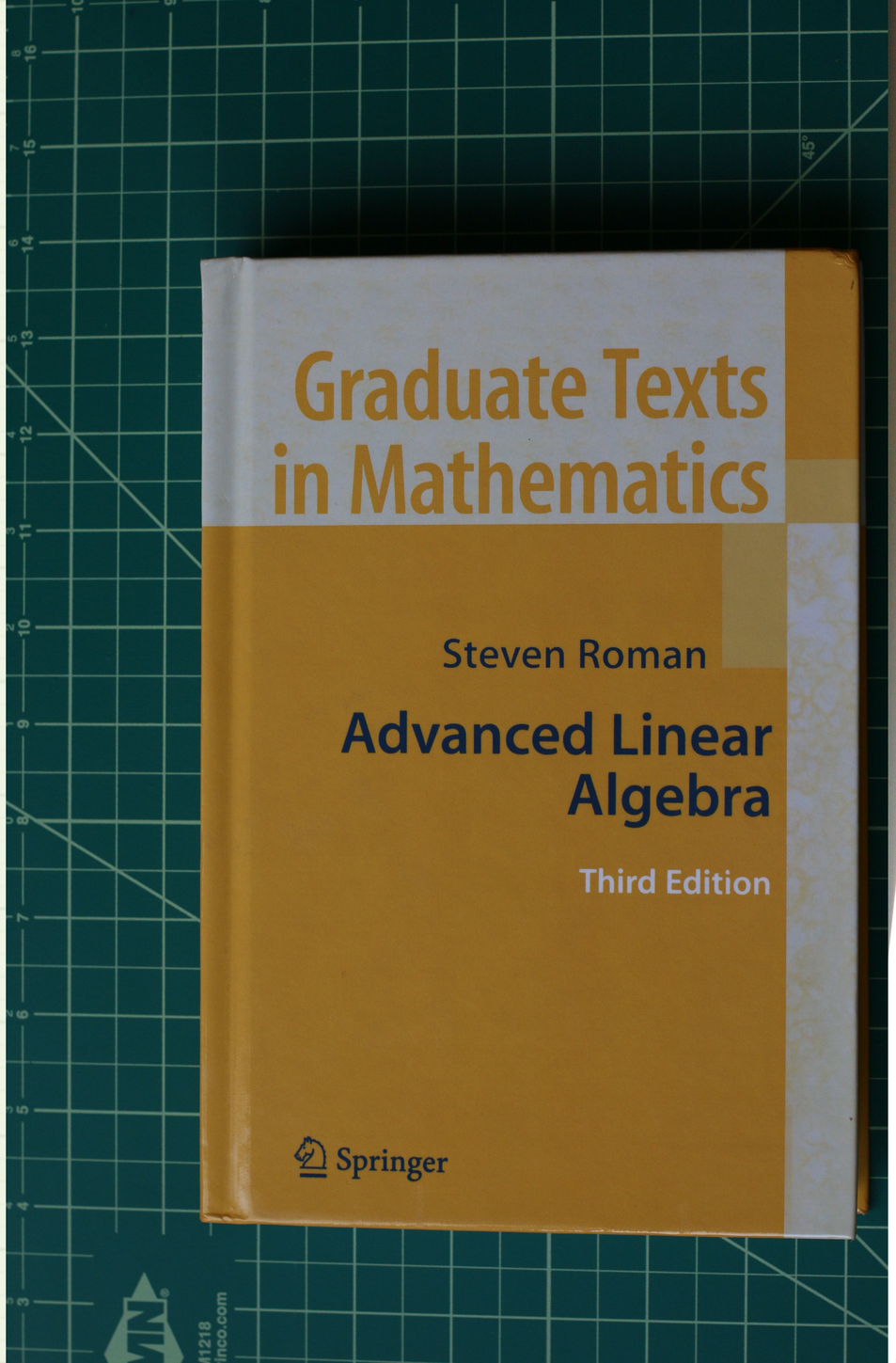
What is a vector?
For programmers and data scientists.

John Mount
Win-Vector LLC

Definition 1, by required properties

- A bit abstract and doesn't tell us why we should care.



Chapter 1
Vector Spaces

Vector Spaces

Let us begin with the definition of one of our principal objects of study.

Definition *Let F be a field, whose elements are referred to as* **scalars**. *A* **vector space** *over F is a nonempty set V, whose elements are referred to as* **vectors**, *together with two operations. The first operation, called* **addition** *and denoted by $+$, assigns to each pair (u, v) of vectors in V a vector $u + v$ in V. The second operation, called* **scalar multiplication** *and denoted by juxtaposition, assigns to each pair $(r, u) \in F \times V$ a vector ru in V. Furthermore, the following properties must be satisfied:*

1) (**Associativity of addition**) *For all vectors $u, v, w \in V$,*

$$u + (v + w) = (u + v) + w$$

2) (**Commutativity of addition**) *For all vectors $u, v \in V$,*

$$u + v = v + u$$

3) (**Existence of a zero**) *There is a vector $0 \in V$ with the property that*

$$0 + u = u + 0 = u$$

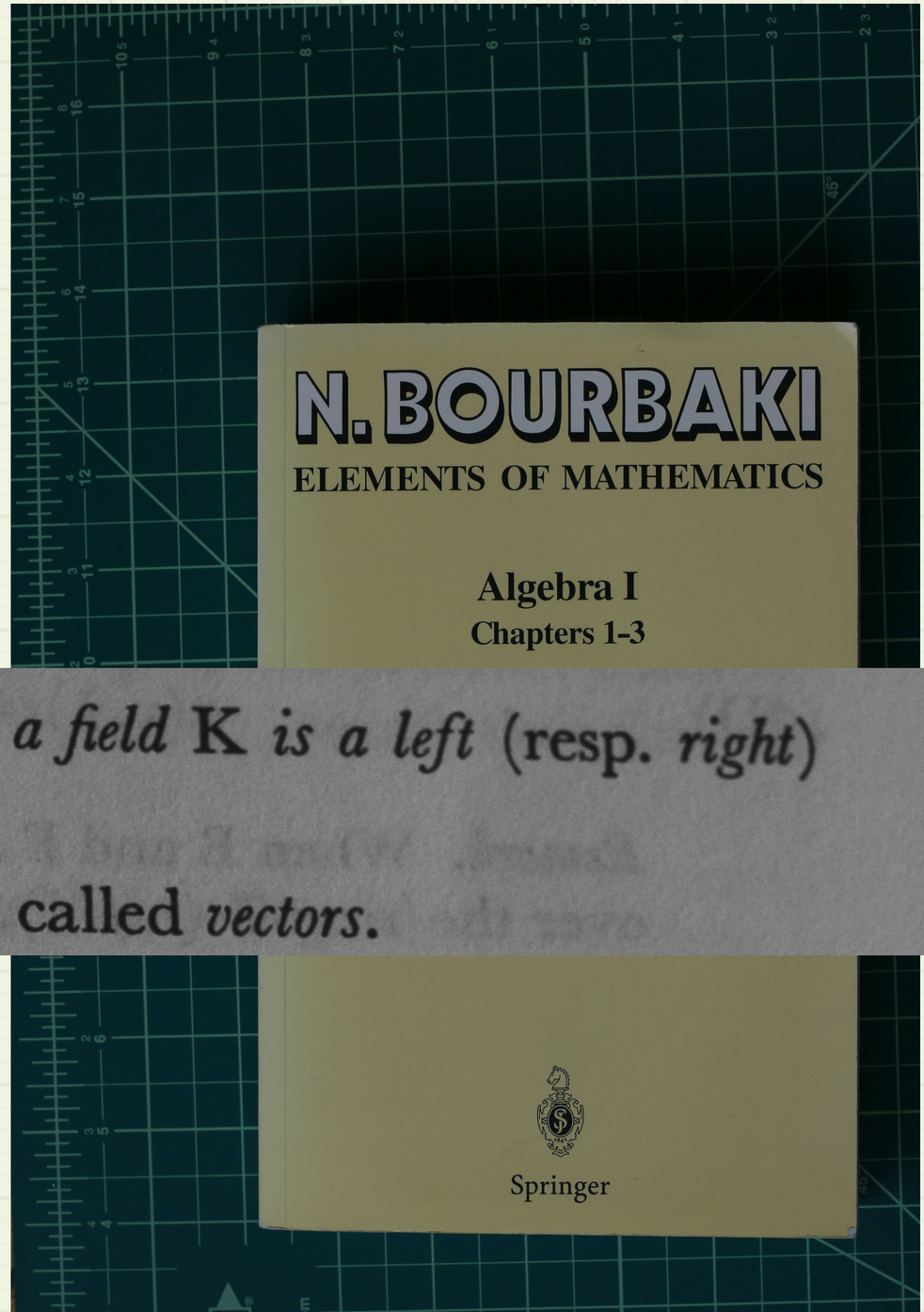
for all vectors $u \in V$.

4) (**Existence of additive inverses**) *For each vector $u \in V$, there is a vector in V, denoted by $-u$, with the property that*

$$u + (-u) = (-u) + u = 0$$

Definition 2: In terms of more fundamental (but even more obscure) abstractions

- Ouch, ouch, ouch!



DEFINITION 2. *A left* (resp. *right*) *vector space over a field* K *is a left* (resp. *right*) K-*module*.

The elements of a vector space are sometimes called *vectors*.

Demonstration by example

$$b = \begin{bmatrix} 1 \\ -2 \\ 7 \end{bmatrix}.$$

This is a ***three-dimensional column vector***. It is represented geometrically in Fig. 1.1, where the three components 1, −2, and 7 are the coordinates of a point in three-dimensional space. Any vector b can be identified in this way with a point in space; there is a perfect match between points and vectors.†

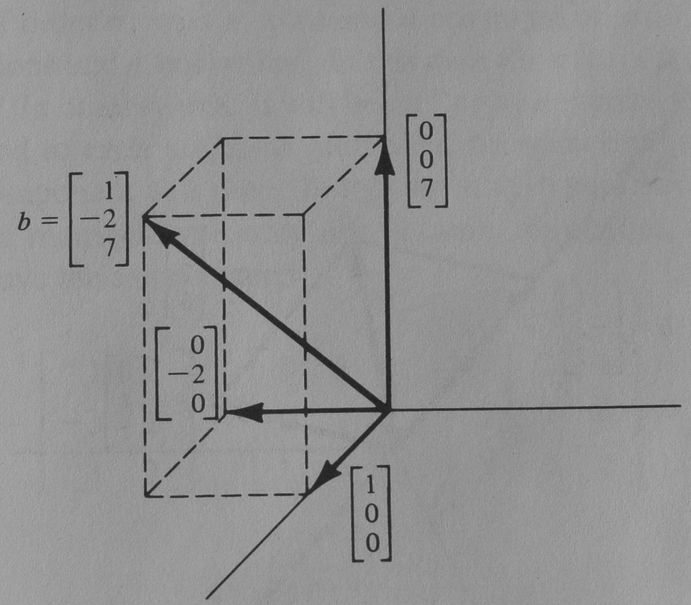


Fig. 1.1. A vector in three-dimensional space.

The basic operations are the addition of two such vectors and the multiplication of a vector by a scalar. Geometrically, $2b$ is a vector in the same direction as b but twice as

† Some authors prefer to say that the arrow is really the vector, but I think it doesn't matter; you can choose the arrow, the point, or the three numbers. (Note that the arrow starts at the origin.) In six dimensions it is probably easiest to choose the six numbers.

Vectors for computer science or data science purposes

- A vector is a data structure that maps a set of integers (0 through k-1 for Python, 1 through k for R) to scalar values (usually double precision floating point numbers).
    - Example: `x = [4, 4, 2.5]`.
- If `x` is a vector then `x[i]` is the i^{th} value of interest (mathematicians write this as x_i).
    - Programming examples:
        - Python: `[4, 4, 2.5][2] = 2.5`
        - R: `c(4, 4, 2.5)[3] = 2.5`
- In data science we mostly use vectors for their ability to hold numbers in order (the implementation detail!). They are treated mostly as lists or arrays.
- We care about vectors because they are incredibly useful data structures.

Vectors: Mathematical Properties

- Mathematicians like to define or describe things in terms of properties, instead of in terms of implementation.
  - We expect to be able to add vectors cell by cell or coordinate by coordinate.
    - `[1, 2, 3] + [1, -1, 0] = [2, 1, 3]`
  - We expect to be able to multiply vectors by scalars
    - `10 * [1, 2, 3] = [10, 20, 30]`
  - Adding or Subtracting a scalar is shorthand for performing the same operation on all cells:
    - `[1, 2, 3] - 2 = [-1, 0, 1]`

Additional operations

- Dot-product:
  - Math: `[1, 2, 3] . [4, 4, 2.5] = 1*4 + 2*4 + 3*2.5`

    `= 4 + 8 + 7.5 = 19.5`
  - Python: `numpy.dot([1, 2, 3], [4, 4, 2.5])`
  - R: `c(1, 2, 3) %*% c(4, 4, 2.5)` (though we have to call as.numeric() on this)
- Euclidean or 2-norm:
  - The Euclidean or 2-norm of a vector is denoted as $|x|_2$ and equal to $\sqrt{x \cdot x}$
- Mean:
  - If x is an "n-vector" (has n entries) then 1_n is shorthand for the vector of n ones.
  - x = [4, 4, 2.5], and 1_n=`[1, 1, 1]`
  - `mean(x)` = average value of x_i. Also equals $x \cdot 1_n / 1_n \cdot 1_n$.

Distance measure

- Sum of squared differences:
  - $sum_i \ (x_i - y_i)^2$

    $= (x-y).(x-y)$
- This is the Euclidean distance squared.
- Zero means identical, large values mean large differences.

Similarity measure

- Pearson Correlation Coefficient

$$\frac{(x - \text{mean}(x)) \cdot (y - \text{mean}(y))}{\sqrt{(x - \text{mean}(x)) \cdot (x - \text{mean}(x))}\sqrt{(y - \text{mean}(y)) \cdot (y - \text{mean}(y))}}$$

- Notice: invariant under shifts and positive scaling
  - `cor(y, x) = cor(x, y)`
  - `cor(x + c 1n, y) = cor(x, y)` (c a scalar value).
  - `cor(c x, y) = cor(x, y)` (`c > 0`).
- If `mean(x)==0`, `x.x == 1`, `mean(y)==0`, and `y.y == 1` then:

  `cor(x, y) = x.y`.

- Notice how each idea is a small extension of earlier ones.
- We will try to make this more concrete by working concrete examples, and then coding the examples in the R or Python programming languages.
- Please come back to these slides as definitions/ notes after seeing how things work and how they are used. Repetition breeds familiarity (which is what feels like understanding).

Take-aways

- Vectors and data matrices are very useful data structures.
- Scatter plots are a graphical representation of the similarities shared by two vectors.
- All materials for this lecture can be found here:

https://github.com/WinVector/VectorDemo