

Module 5: Vectors and Matrices

Prof. Dr. Florian Haselbeck

Memoona Shakeel

Weihenstephan-Triesdorf University of Applied Sciences

Learning outcomes

Understand what vectors and matrices are, how to work with them in Python, and why they matter for data in agriculture.

Key Topics

- Scalar . Vector . Matrix
- Vectors in Python
- Vector operations
- Matrices in Python
- Dot Product
- NumPy basic overview

Scalar · Vector · Matrix What is the difference?

Key idea: These are three ways to organize numbers. You already know scalars (single numbers) and lists, vectors and matrices are just the mathematical names for them.

Scalar:

A scalar is a number, like 3, -5, 0.368, etc.

Example: temperature = 23.5

You already use these every day

Vector:

A vector is a list of numbers (can be in a row or column)

Example: [12.4, 9.8, 14.1]

Like a Python list.

Matrix:

A grid of numbers. A matrix is an array of numbers (one or more rows, one or more columns)

Example: sensor readings per field per day. Like a spreadsheet.

Scalar

24

Vector

$\begin{bmatrix} 2 & -8 & 7 \end{bmatrix}$

row

or
column $\begin{bmatrix} 2 \\ -8 \\ 7 \end{bmatrix}$

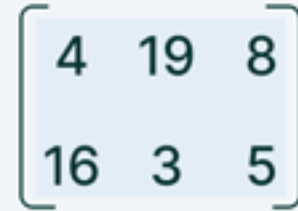
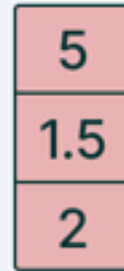
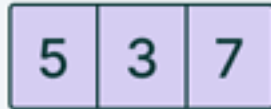
Matrix

$\begin{bmatrix} 6 & 4 & 24 \\ 1 & -9 & 8 \end{bmatrix}$

row(s) × column(s)

Scalar · Vector · Matrix What is the difference?

(11)



Scalar

Row Vector
(shape 1×3)

Column Vector
(shape 3×1)

Matrix

Vectors in Python

A vector is just a Python list. A sequence of numbers, each at a numbered position (index).

```
# plain Python list
# Vector 1
v1 = [2, 4, 6, 8, 10]
# Vector 2
v2 = [1, 3, 5, 7, 9]
```

Accessing Values Using Index

Remember:

index 0 → first value

index 1 → second value

index 2 → third value. And so on till the length

```
v1 = [2, 4, 6, 8, 10]
print(v1[0])    # first value. Output 2
print(v1[4])    # fifth value. Output 10
```

index =	0	1	2	3	4
v1 =	2	4	6	8	10

index =	0	1	2	3	4
v2 =	1	3	5	7	9

Vector operations (element by element)

Key idea: When you add, subtract or multiply a vector, the operation happens to every element. **NumPy** does this automatically.

Vector Addition

index =	0	1	2	3	4
v1 =	2	4	6	8	10

v2 =	1	3	5	7	9
------	---	---	---	---	---

v1 + v2 (element-wise addition)

	2	4	6	8	10
+	1	3	5	7	9
	3	7	11	15	19
	↓	↓	↓	↓	↓
v1 + v2 =	3	7	11	15	19

So, $v1 + v2 = [3, 7, 11, 15, 19]$

Vector operations (element by element)

DOT PRODUCT ($v1 \cdot v2$)

index =	0	1	2	3	4
$v1 =$	2	4	6	8	10
$v2 =$	1	3	5	7	9

- Step 1: Multiply corresponding elements

0	1	2	3	4
2	4	6	8	10
x 1	x 3	x 5	x 7	x 9
2	12	30	56	90

- Step 2: Add all the results

$$2 + 12 + 30 + 56 + 90 = 190$$

$$\text{So, } v1 \cdot v2 = 190$$

VECTOR SUBTRACTION ($v1 - v2$)

index =	0	1	2	3	4
$v1 =$	2	4	6	8	10
$v2 =$	1	3	5	7	9
$v1 - v2 =$	1	1	1	1	1

- Subtract corresponding elements

0	1	2	3	4
2	4	6	8	10
- 1	- 3	- 5	- 7	- 9
1	1	1	1	1
↓	↓	↓	↓	↓
1	1	1	1	1

$$\text{So, } v1 - v2 = [1, 1, 1, 1, 1]$$

Matrices

What it is: A matrix is a 2D grid (rows and columns). Think of it as a spreadsheet or a table of sensor readings

Matrix A =

	Column index →	0	1	2
0		1	2	3
1	Row index ↓	4	5	6

Matrix B =

	Column index →	0	1	2
0		7	8	9
1	Row index ↓	10	11	12

Matrices Operations

1) MATRIX ADDITION (A + B)

Matrix A				Matrix B				Result (A + B)			
Column index →				Column index →				Column index →			
0 1 2				0 1 2				0 1 2			
Row index ↓								Row index ↓			
0	1	2	3	7	8	9		0	8	10	12
1	4	5	6	10	11	12		1	14	16	18

$A + B =$

(Add corresponding elements)

2) MATRIX DOT PRODUCT (A • B)

Matrix A				Matrix B				Result (A • B)			
Column index →				Column index →				Column index →			
0 1 2				0 1 2				0 1 2			
Row index ↓								Row index ↓			
0	1	2	3	7	8	9		0	7	16	27
1	4	5	6	10	11	12		1	40	55	72

$A \bullet B =$

(Multiply corresponding elements)

3) MATRIX SUBTRACTION (A - B)

Matrix A				Matrix B				Result (A - B)			
Column index →				Column index →				Column index →			
0 1 2				0 1 2				0 1 2			
Row index ↓								Row index ↓			
0	1	2	3	7	8	9		0	-6	-6	-6
1	4	5	6	10	11	12		1	-6	-6	-6

$A - B =$

(Subtract corresponding elements)

NumPy (your toolkit for vectors and matrices)

What it is: NumPy is a Python library that makes working with vectors and matrices fast and simple. You will use it in almost every data science project.

install: `pip install numpy`,

import: `import numpy as np`
then use `np.array()`, `np.dot()`, etc.

Why not plain Python lists?

NumPy arrays are much faster for math's and work with the whole array at once.

```
# Import NumPy library
import numpy as np

# Vector 1
v1 = np.array([2, 4, 6, 8, 10])
|
# Vector 2
v2 = np.array([1, 3, 5, 7, 9])

# VECTOR Operations
addition = v1 + v2 # Adds corresponding elements
subtraction = v1 - v2 # Subtracts corresponding elements
multiplication = v1 * v2 # Multiplies matching positions
dot_product = np.dot(v1, v2)

print(addition) # Output [ 3  7 11 15 19]
print(subtraction) # [1 1 1 1 1]
print(multiplication) # [ 2 12 30 56 90]
print(dot_product) # 190
```

Overview of key content

- Scalar = one number · Vector = a list of numbers · Matrix = a grid of numbers
- A vector in Python is a NumPy array, accessed by index, starting at 0
`v = np.array([12.4, 9.8, 14.1])` → `v[0]` is 12.4, `v[1]` is 9.8
- Operations on vectors happen to every element automatically
`v + 2` adds 2 to every value. `v + w` adds element by element. No loops needed. NumPy does it all at once.
- Dot product = multiply pairs then add up, gives a single number
`np.dot(weights, readings)` gives a weighted sum. Both vectors must have the same length.
The result is always one number.

Thank you!

Contact Detail



Prof. Dr. Florian Haselbeck



florian.haselbeck@hswt.de



Raum D1.230a

For any questions, please feel free to use the **Moodle-Course Forum!**