Week 7: Linear Algebra (1)

Objectives:

- Introduce the fundamental concepts of vectors and matrices, focusing on their properties, operations, and real-world applications.
- Develop proficiency in solving linear equations and understanding vector spaces, including bases and dimensions.
- Prepare for advanced topics, including eigenvalues and eigenvectors, emphasizing practical skills.

What is Linear Algebra

- A branch of mathematics that deals with linear equations and their representations as vectors and matrices.

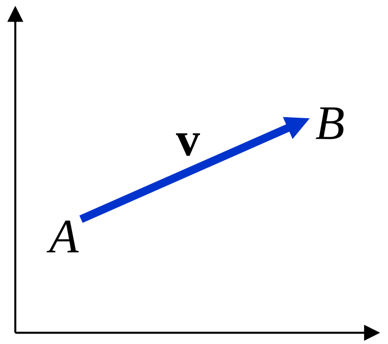
- An important math skill for machine learning.

  - Fit a model to a dataset
  - Data classification
  - Linear regression
  - Principal component analysis
  - Recommender systems
  - Image data processing
  - Deep learning

Vectors

- You can represent a vector $\mathbf{v}$ from point A to point B, as a directed line segment from A to B, written $\overrightarrow{AB}$.
  - Name a vector with a single boldface lowercase letter, such as $\mathbf{v}$.
  - Point A is the vector's initial point, or tail.
  - Point B is the vector's terminal point, or head.

- A vector corresponds to a displacement from point A to point B.

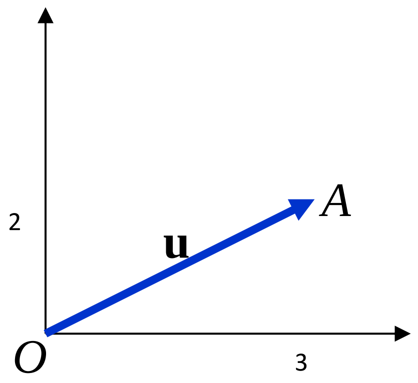


Vectors

- Every point A in the plane corresponds to a vector whose tail is at the origin O and the head is point A.

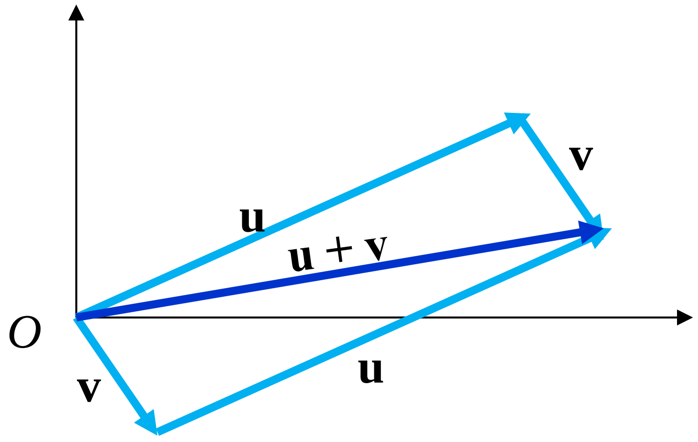
  - Example: If the coordinates of point A are (3, 2), we can write vector $\mathbf{u} = \overrightarrow{OA} = [3, 2]$.

  - We can represent the vector $\mathbf{v}$ as a row vector $[3,\ 2]$ or as a column vector $\begin{bmatrix} 3 \\ 2 \end{bmatrix}$



Vector Arithmetic

- Suppose vector $\mathbf{u} = [3, 2]$ and vector $\mathbf{v} = [1, -1]$.
  - Then $\mathbf{u} + \mathbf{v} = [3, 2] + [1, -1] = [3 + 1, 2 - 1] = [4, 1]$
  - Add the corresponding components: $3 + 1 = 4$ and $2 - 1 = 1$
  - Note that $\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$.



Scalar Multiplication of a Vector

- Scalar multiplication
  - To multiply a vector by a constant, multiply each component of the vector by the constant.
  - Example: If $\mathbf{u} = [3, 2]$, then $2\mathbf{u} = [6, 4]$

Linear Combination

A vector $\mathbf{v}$ is a linear combination of vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k$
if there are scalars $c_1, c_2, \ldots, c_\mathrm{k}$
such that $\mathbf{v} = c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \ldots + c_k\mathbf{v}_k$.

- Example: Vector $\begin{bmatrix} 2 \\ -2 \\ -1 \end{bmatrix}$ is a linear combination of vectors $\begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$, $\begin{bmatrix} 2 \\ -3 \\ 1 \end{bmatrix}$, and $\begin{bmatrix} 5 \\ -4 \\ 0 \end{bmatrix}$

since

$$3\begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} + 2\begin{bmatrix} 2 \\ -3 \\ 1 \end{bmatrix} - \begin{bmatrix} 5 \\ -4 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ -2 \\ -1 \end{bmatrix}$$

Vector and Linear Combination

Linear combination for any $\alpha_1$and α_2 (number)

$$V^* = \alpha_1 V_1 + \alpha_2 V_2$$

Vector

V^* is a linear combination of V_1 and V_2

$\Rightarrow$ V^*, V_1, V_2 are "linearly dependent"

$\Rightarrow$ V_1, V_2, V_3 If you "failed" to find any linear comb. from V_1, V_2, V_3

$\Rightarrow$ Linearly independent.

(ex) $V_1 = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$ $V_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ $V_3 = \begin{bmatrix} 4 \\ 5 \end{bmatrix}$

$$V_3 = V_1 + 2V_2$$

$$= \begin{bmatrix} 2 \\ 3 \end{bmatrix} + \begin{bmatrix} 2 \\ 2 \end{bmatrix} = \begin{bmatrix} 4 \\ 5 \end{bmatrix}$$

Linearly Independent

V_1, V_2, V_3, V_4

if you can find $V_4 = \alpha V_1 + \beta V_2 + \gamma V_3$ ⇒ linearly dep.

if you cannot find 〃 ⇒ linearly indep.

→ Check redundant feature (feature selection)

Dot Product

If vectors

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} \text{ and } \mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

then their dot product $\mathbf{u} \cdot \mathbf{v}$ is

$$\mathbf{u} \cdot \mathbf{v} = u_1 v_1 + u_2 v_2 + \cdots + u_n v_n$$

Dot Product

- Example: Let

$$\mathbf{u} = \begin{bmatrix} 1 \\ 2 \\ -3 \end{bmatrix} \text{ and } \mathbf{v} = \begin{bmatrix} -3 \\ 5 \\ 2 \end{bmatrix}$$

then $\mathbf{u} \cdot \mathbf{v} = 1 \cdot (-3) + 2 \cdot 5 + (-3) \cdot 2 = 1$

- Note that a dot product is a <u>scalar</u>.

Length or Norm

The length (or norm) of a vector $\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$

is the nonnegative scalar

$$\|\mathbf{v}\| = \sqrt{\mathbf{v} \cdot \mathbf{v}} = \sqrt{v_1^2 + v_2^2 + \cdots + v_n^2}$$

- Example: $\|[2,3]\| = \sqrt{2^2 + 3^2} = \sqrt{13}$

Normalize a Vector

- A unit vector is a vector of length 1.
  - Examples: $\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \mathbf{e}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$
    are unit vectors.

- If vector $\mathbf{v}$ is nonzero, we can find a unit vector in the same direction as $\mathbf{v}$ by dividing $\mathbf{v}$ by its own length.

- Finding a unit vector in the same direction as as vector $\mathbf{v}$ is called normalizing a vector.

Normalize a Vector

- Example: If vector $\mathbf{v} = \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix}$ then $\|\mathbf{v}\| = \sqrt{14}$.

The unit vector in the <u>same direction</u> as $\mathbf{v}$ is

$$\mathbf{u} = \left(\frac{1}{\|\mathbf{v}\|}\right)\mathbf{v} = \left(1/\sqrt{14}\right)\begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix} = \begin{bmatrix} 2/\sqrt{14} \\ -1/\sqrt{14} \\ 3/\sqrt{14} \end{bmatrix}$$

1_Vectors.ipynb

Vector

The dot/inner product (scalar) *output = a number.

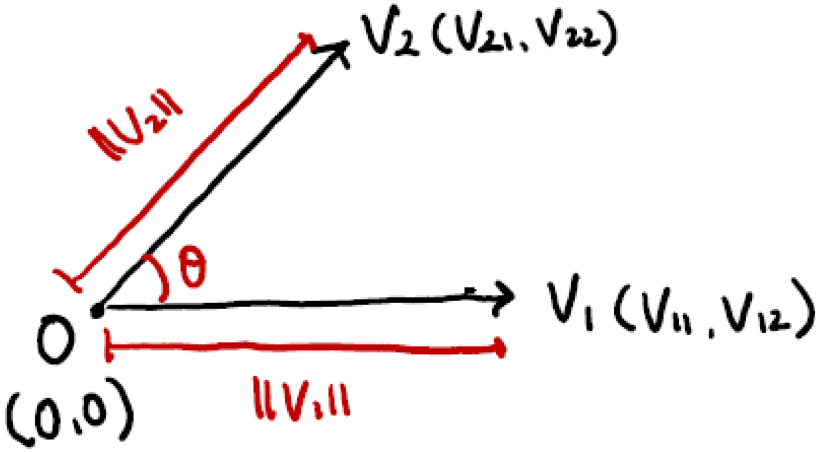
def. * $V_1 \bullet V_2 = \|V_1\| \|V_2\| \cdot \cos\theta$

$V_1 \bullet V_2 = V_2 \bullet V_1$

* NOTE Norm $\|V_1\| = \sqrt{V_{11}^2 + V_{12}^2 + \dots + V_{1n}^2}$

$V = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix}$. $W = \begin{bmatrix} W_1 \\ W_2 \\ W_3 \end{bmatrix}$

$V \bullet W = V_1 \cdot W_1 + V_2 W_2 + V_3 W_3$



$\boxed{\cos\theta} = \dfrac{V_1 \cdot V_2}{\|V_1\| \|V_2\|}$

Cosine similarity

$\boxed{V \bullet V}$.

$= V_1 \cdot V_1 + V_2 \cdot V_2 + V_3 \cdot V_3 + \dots + V_n \cdot V_n$

$= V_1^2 + V_2^2 + \dots V_n^2$

$\|V\| = \sqrt{V \cdot V}$

Compute the Cosine Similarity

Question

1. A =[3, 0 ,0] B= [5, 0, 0], the Cosine Similarity?

2. A =[3, 0 ,0] B= [-5, 0, 0], the Cosine Similarity?

Compute the Cosine Similarity

Question

3. A =[3, 0 ,0] B= [0, 5, 0], the Cosine Similarity?

4. A =[3, 0 ,0] B= [0, 0, -5], the Cosine Similarity?

Compute the Cosine Similarity

Question

5. P =[2, 3 ,1, 4, 2] Q= [1, 2, 3, 2, 1], the Cosine Similarity?

6. Determine which two vectors among X = [1, 0, 1], Y = [0, 1, 0], and Z = [1, 1, 1] are the most similar using cosine similarity.

Cosine Similarity

- Cosine similarity ranges from -1 to 1.
  - A value of 1 indicates perfect similarity (vectors pointing in the same direction).
  - A value of 0 indicates no similarity (perpendicular vectors).
  - A value of -1 indicates perfect dissimilarity (vectors pointing in opposite directions).

- In terms of similarity:
  - A cosine similarity of 0 means the vectors are completely independent (no similarity).
  - A negative cosine similarity means the vectors have some degree of opposite orientation, but they're not completely independent.

Example of Cosine Similarity

Calculate the cosine similarity between the following two documents. Each document is represented by word frequencies.

Document 1: {"apple": 2, "banana": 3, "cherry": 1}

Document 2: {"apple": 1, "banana": 2, "date": 3}

- Unique words: "apple", "banana", "cherry", "date“
- Vector representations:

doc1: [2, 3, 1, 0]
doc2: [1, 2, 0, 3]

Dot product: (21) + (32) + (10) + (03) = 2 + 6 + 0 + 0 = 8
sqrt(2^2 + 3^2 + 1^2 + 0^2) = sqrt(14) ≈ 3.7417

Cosine similarity: 8 / (3.7417 * 3.7417) ≈ 0.6806

The cosine similarity of approximately 0.6806 indicates a moderate level of similarity between the two documents.

They share some common words ("apple" and "banana") with similar frequencies, but also have unique words ("cherry" in doc1 and "date" in doc2), which prevents them from being perfectly similar.

Python Code

Norm

```
v_norm1 = la.norm(v1)
print(v_norm1)
v_norm2 = np.sqrt(v1@v1)
print(v_norm2)
```

```
4.123105625617661
4.123105625617661
```

Dot product

$V \cdot W = V@W$

```
vd1 = np.dot(v1,v2)
vd2 = v1@v2
vd3 = v1.dot(v2)
print(vd1)
print(vd2)
print(vd3)
```

```
8
8
8
```

Matrices

- A matrix is a two-dimensional array of values.
  - Name a matrix with an uppercase italic letter, such as A.
  - Examples:

$$A = \begin{bmatrix} \sqrt{5} & -1 & 0 \\ 2 & \pi & 1/2 \end{bmatrix} \qquad C = [7]$$

$$E = \begin{bmatrix} 1 & 12 & 5 \end{bmatrix}$$

$$B = \begin{bmatrix} 5.1 & 1.2 & -1 \\ 6.9 & 0 & 4.4 \\ -7.3 & 9 & 8.5 \end{bmatrix} \qquad F = \begin{bmatrix} -6 \\ 14 \end{bmatrix}$$

- NumPy allows us to do matrix arithmetic.

Special Matrices

- A zero matrix has all zeros as elements.
  - Example:

$$O = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

- An identity matrix is a square matrix with ones along the diagonal and zeros elsewhere.
  - Example:

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Matrix

Identity matrix (I) $\Rightarrow \begin{bmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{bmatrix}$

$I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ $I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

$V = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$

$$I_3 V = \underset{3 \times 3}{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}} \underset{3 \times 1}{\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

3x1 vector

(nxn) matrix (square) A B (nxn) matrix

if $AB = I$, $BA = I$ B is an inverse matrix of A

A^{-1} : Inverse matrix $AA^{-1} = I$ $A^{-1}A = I$

Python Code

Matrix

```
A=np.array([[1,3,4],[2,3,5]])
print(A)
print(".shape: show nxm matrix shape: ", A.shape)
```

```
[[1 3 4]
 [2 3 5]]
.shape: show nxm matrix shape:  (2, 3)
```

```
I= np.eye(3)
I1 = np.identity(3)
print(I)
print(I1)
```

```
[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]
[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]
```

```
O = np.zeros([3,3])
print(O)
```

```
[[0. 0. 0.]
 [0. 0. 0.]
 [0. 0. 0.]]
```

Matrix Arithmetic

- Addition

  - You can only add matrices that have the same shape (same numbers of rows and columns).
  - The sum matrix will have the same shape.
  - Add the matrices element-by-element.

  - Example: $A = \begin{bmatrix} 1 & 4 & 0 \\ -2 & 6 & 5 \end{bmatrix} \quad B = \begin{bmatrix} -3 & 1 & -1 \\ 3 & 0 & 2 \end{bmatrix}$

$$A + B = \begin{bmatrix} -2 & 5 & -1 \\ 1 & 6 & 7 \end{bmatrix}$$

  - For all matrices, $A + B = B + A$ and $A + O = O + A = A$

Matrix Arithmetic

- Scalar multiplication

  - To multiply a matrix by a constant, multiply each element of the matrix by the constant.

  - Example:

$$A = \begin{bmatrix} 1 & 4 & 0 \\ -2 & 6 & 5 \end{bmatrix}$$

$$2A = \begin{bmatrix} 2 & 8 & 0 \\ -4 & 12 & 10 \end{bmatrix}$$

Matrix Multiplication

If A is an m x n matrix and B is an n x r matrix, then the product $C = AB$ is an m x r matrix.

The (i, j) element of the product c_{ij} is

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{in}b_{nj}$$

- In other words, the c_{ij} element of the product is the dot product of the i^{th} row of A and the j^{th} column of B.

- If A is an n x n square matrix and I is the n x n identity matrix, then $AI = IA = A$.

Example: Matrix Multiplication

- Ann and Bert each shop for fruit, but for different numbers of each fruit. There are two fruit stores with different prices. How much would each person spend if they shopped at these two stores?
  - Numbers of fruit

	Apples	Grapefruit	Oranges
Ann	6	3	10
Bert	4	8	5

  - Fruit prices

	Store A	Store B
Apples	\$0.10	\$0.15
Grapefruit	\$0.40	\$0.30
Oranges	\$0.10	\$0.20

Example: Matrix Multiplication

	Apples	Grapefruit	Oranges
Ann	6	3	10
Bert	4	8	5

	Store A	Store B
Apples	\$0.10	\$0.15
Grapefruit	\$0.40	\$0.30
Oranges	\$0.10	\$0.20

- Ann at Store A: 6(0.10) + 3(0.40) + 10(0.10) = \$2.80
- Ann at Store B: 6(0.15) + 3(0.30) + 10(0.20) = \$3.80
- Bert at Store A: 4(0.10) + 8(0.40) + 5(0.10) = \$4.10
- Bert at Store B: 4(0.15) + 8(0.30) + 5(0.20) = \$4.00

- Equivalent to:

$$\begin{bmatrix} 6 & 3 & 10 \\ 4 & 8 & 5 \end{bmatrix} \begin{bmatrix} 0.10 & 0.15 \\ 0.40 & 0.30 \\ 0.10 & 0.20 \end{bmatrix} = \begin{bmatrix} 2.80 & 3.80 \\ 4.10 & 4.00 \end{bmatrix}$$

Python Code

Transpose

```
print(A.T)
```

```
[[1 2]
 [3 3]
 [4 5]]
```

Matrix Multiplication

```
v1 = np.array([2, 2, 3])
v2 = np.array([-1, 2, 2])
v3 = v1 + v2
```

```
A = np.vstack((v1,v2,v3))
B = np.array([[1, 4,-1],[3,2,7],[2,5,5]])
print(A)
print(B)
C = A@B   # original matrix multiplication
print(C)
D= A*B    # elementwise multiplication
print(D)
```

A@B

(CNN with filter) ~ DATA 255

```
[[ 2  2  3]
 [-1  2  2]
 [ 1  4  5]]
[[ 1  4 -1]
 [ 3  2  7]
 [ 2  5  5]]
[[14 27 27]
 [ 9 10 25]
 [23 37 52]]
[[ 2  8 -3]
 [-3  4 14]
 [ 2 20 25]]
```

More Vector and Matrix Facts

- Miscellaneous facts
  - Matrix transpose
  - QR factorization
  - See:

3_LinearAlgebra.ipynb

4_MatrixInverse.ipynb

Linear Equations

- A linear equation is one where the variables have “simple” forms.
  - No exponents (other than the implied 1)
  - The graph is always a straight line.

- You can see examples from linear regression:

$$y = mx + b$$

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k$$

Linear Equations

- Example: A linear equation in n variables $x_1, x_2, \ldots, x_n$:

$$\boxed{a_1x_1 + a_2x_2 + \cdots + a_nx_n = b}$$

- The coefficients $a_1, a_2, \ldots, a_n$ and the constant term b are constants.

- When the number of variables is at most three, we can use the variable names x, y, and z in place of the x_i.
    - Example: $3x - 3y + 2z = 16$

Augmented Matrix

- We can solve a system of linear equations with an augmented matrix.

  - Example: Solve

$$\begin{aligned} x - y - z &= 2 \quad \text{a} \\ 3x - 3y + 2z &= 16 \quad \text{b} \\ 2x - y + z &= 9 \quad \text{c} \end{aligned} \qquad \left[\begin{array}{ccc|c} 1 & -1 & -1 & 2 \\ 3 & -3 & 2 & 16 \\ 2 & -1 & 1 & 9 \end{array}\right]$$

  - Begin by eliminating x from equations b and c.

Solve with an Augmented Matrix

$$\begin{aligned} x - y - z &= 2 \quad \text{a} \\ 3x - 3y + 2z &= 16 \quad \text{b} \\ 2x - y + z &= 9 \quad \text{c} \end{aligned}$$

$$\left[\begin{array}{ccc|c} 1 & -1 & -1 & 2 \\ 3 & -3 & 2 & 16 \\ 2 & -1 & 1 & 9 \end{array}\right]$$

Subtract 3 times equation a from equation b:

$$\begin{aligned} x - y - z &= 2 \quad \text{a} \\ 5z &= 10 \quad \text{b} \\ 2x - y + z &= 9 \quad \text{c} \end{aligned}$$

Subtract 3 times the first row from the second row:

$$\left[\begin{array}{ccc|c} 1 & -1 & -1 & 2 \\ 0 & 0 & 5 & 10 \\ 2 & -1 & 1 & 9 \end{array}\right]$$

Subtract 2 times equation a from equation c:

$$\begin{aligned} x - y - z &= 2 \quad \text{a} \\ 5z &= 10 \quad \text{b} \\ y + 3z &= 5 \quad \text{c} \end{aligned}$$

Subtract 2 times the first row from the third row:

$$\left[\begin{array}{ccc|c} 1 & -1 & -1 & 2 \\ 0 & 0 & 5 & 10 \\ 0 & 1 & 3 & 5 \end{array}\right]$$

Solve with an Augmented Matrix

$$\begin{aligned} x - y - z &= 2 \quad \text{a} \\ 5z &= 10 \quad \text{b} \\ y + 3z &= 5 \quad \text{c} \end{aligned}$$

$$\left[\begin{array}{ccc|c} 1 & -1 & -1 & 2 \\ 0 & 0 & 5 & 10 \\ 0 & 1 & 3 & 5 \end{array}\right]$$

Interchange equations b and c:

$$\begin{aligned} x - y - z &= 2 \quad \text{a} \\ y + 3z &= 5 \quad \text{b} \\ 5z &= 10 \quad \text{c} \end{aligned}$$

Interchange the second and third rows:

$$\left[\begin{array}{ccc|c} 1 & -1 & -1 & 2 \\ 0 & 1 & 3 & 5 \\ 0 & 0 & 5 & 10 \end{array}\right]$$

- Now as before, we can solve for x, y, and z using back substitution.

Gaussian Elimination

- A system of linear equations is ready to be solved by back substitution if the corresponding augmented matrix is in row echelon form.

- Gaussian elimination: The process of reducing a matrix to its row echelon form in order to solve a system of linear equations.

  - Gauss-Jordan elimination puts a matrix into reduced row echelon form: It's in row echelon form and each leading entry is a 1 (called a leading 1), and each column containing a leading 1 has zeros elsewhere.

Solution

- Solving a system of linear equations by hand is a real pain!

- Fortunately, Python has a numpy function:
  `np.linalg.solve()`.

It solves equations of the form **Ax = B** for the vector **x**, where **A** is the coefficient matrix and **B** is the outcome vector.

Returns:
The function returns the vector (or matrix) **x** that satisfies **Ax = B**.

4_Matrix_Solve.ipynb

Solution Using Python

$$\left[\begin{array}{ccc|c} 8 & 25 & 16 & 397{,}000 \\ 25 & 87 & 55 & 1{,}281{,}100 \\ 16 & 55 & 36 & 817{,}700 \end{array}\right]$$

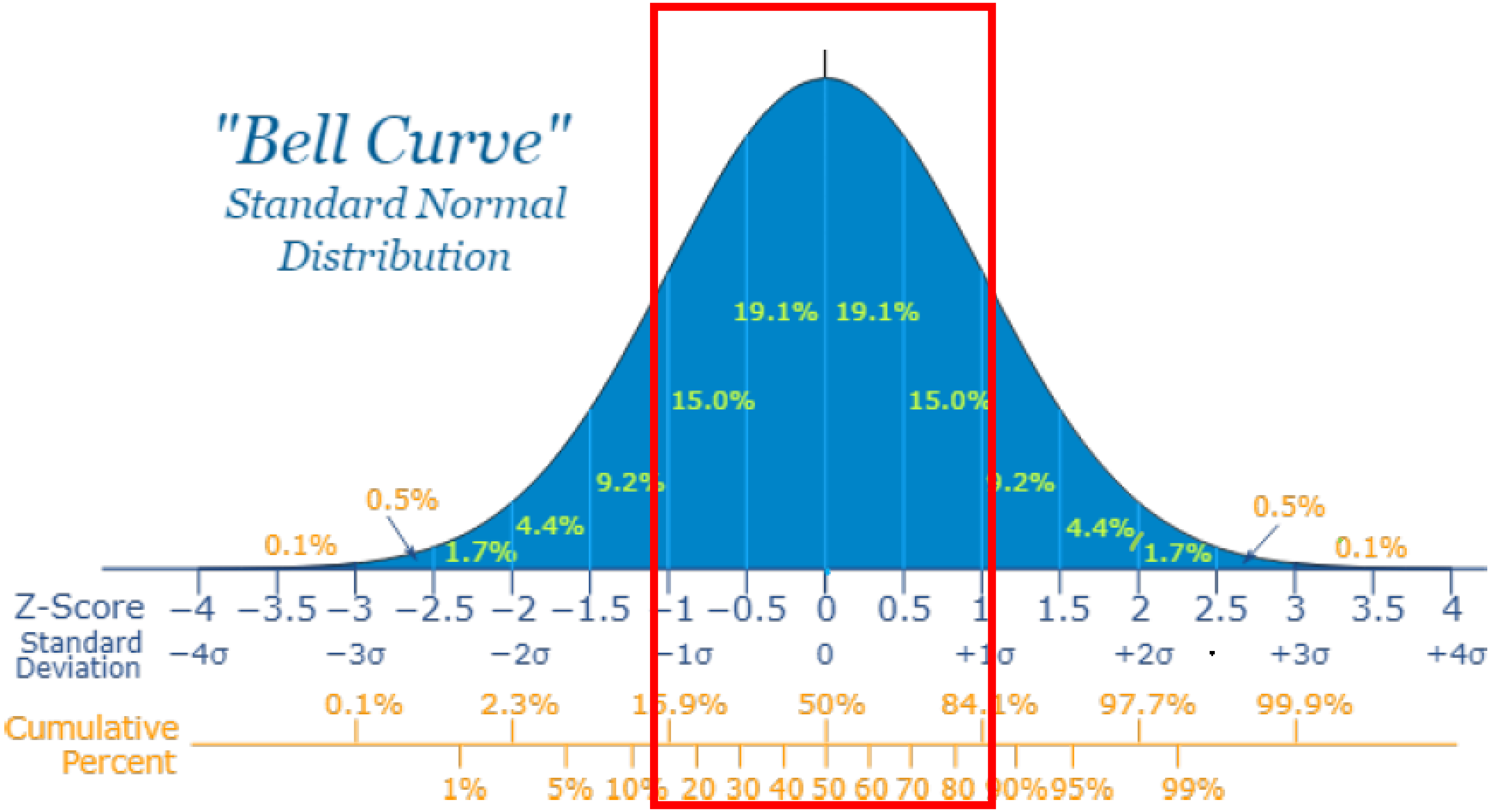
```python
import numpy as np

A = np.array([[ 8, 25, 16],
              [25, 87, 55],
              [16, 55, 36]])

b = np.array([397_000, 1_281_100, 817_700])

solution = np.linalg.solve(A, b)

print(f'β0 = {solution[0]:9,.2f}')
print(f'β1 = {solution[1]:9,.2f}')
print(f'β2 = {solution[2]:9,.2f}')
```

```
β0 = 35,191.67
β1 = 4,133.33
β2 = 758.33
```

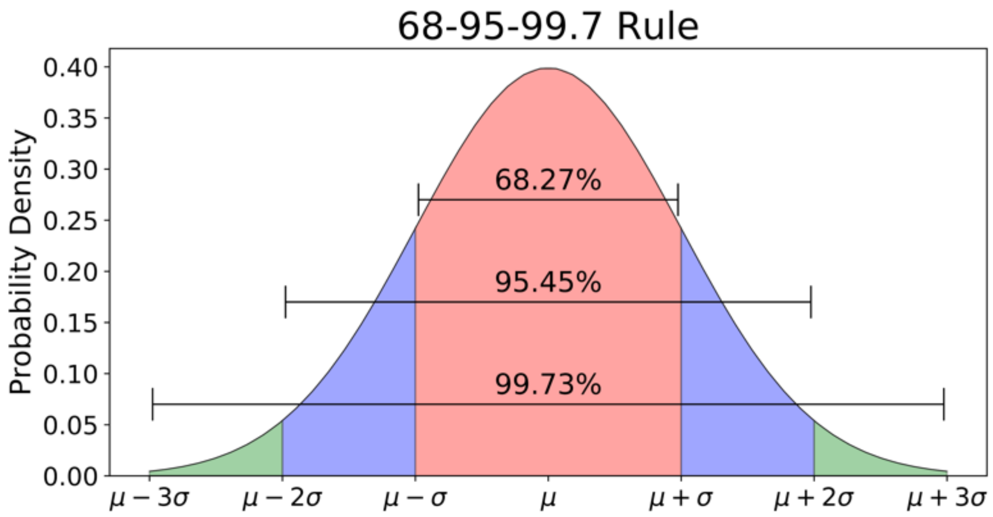
Midterm

- Time: 6:00 pm -7:40 pm (100 min).
- Closed book/lecture slides.
- Bring your computer (with a cam) to access Canvas Lockdown.
- Communication with someone is not allowed
  - Your video/voice will be recorded.

Standard Normal Distribution



Properties of the Normal Distribution



The total area must equal 1.

- The probability that a normal random value is within one standard deviation σ of the mean μ is 68.27%.
- The probability is 95.45% that it's within two standard deviations of the mean.
- The probability is 99.74% that it's within three standard deviations of the mean.

Have you heard of 6-Sigma (6σ) for process improvement? That's 99.99966%