5. 벡터 공간(Vector Space)

1. 벡터 공간

   벡터 공간은 수학에서 벡터와 스칼라 연산이 가능한 공간을 의미합니다. 벡터 공간은 아래 두 가지 연산이 정의되어야 합니다.

   - 벡터 덧셈 : 벡터 공간 내 두 벡터를 더하면 여전히 벡터 공간 내의 벡터가 됩니다.
   - 스칼라 곱셈 : 벡터와 스칼라(숫자)를 곱하면 여전히 벡터 공간 내의 벡터가 됩니다.

   예시

   $$V = R^2 = (x, y) \mid x, y \in R$$

   위는 2차원 실수 벡터 공간으로, 모든 2차원 실수 벡터의 집합입니다.

   벡터 공간의 공리(기본 규칙)

   - 벡터 공간에는 벡터의 덧셈과 스칼라 곱셈이 존재하며, 이 두 연산은 연속적인 규칙(교환법칙, 결합법칙, 분배법칙 등)을 만족합니다.
     - 교환법칙(Commutative Law) : 두 벡터를 더하거나, 스칼라를 곱할 때 순서에 상관없이 같은 결과가 나온다는 의미입니다.

       벡터 덧셈에서의 교환법칙

       $$u + v = v + u$$

       즉, 두 벡터를 더할 때 그 순서는 중요하지 않다는 것입니다.

       예시

       $$(1, 2) + (3, 4) = (3, 4) + (1, 2) = (4, 6)$$

       벡터의 덧셈에서는 벡터의 순서가 바뀌어도 결과가 같다는 의미입니다.
     - 결합법칙 (Associative Law)

       결합법칙은 벡터들의 덧셈이나 스칼라 곱에서 괄호의 위치에 상관없이 결과가 동일하다는 법칙입니다.

       벡터 덧셈에서의 결합법칙

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

세 벡터를 더할 때, 어느 두 벡터를 먼저 더하든 상관없이 결과가 같습니다.

예시

$$((1,2)+(3,4))+(5,6)=(1,2)+((3,4)+(5,6))$$
$$(4,6)+(5,6)=(1,2)+(8,10)=(9,12)$$

스칼라 곱에서의 결합법칙

스칼라 곱에서도 결합법칙이 성립합니다.

$$a(bv)=(ab)v$$

여기서 a와 b는 스칼라(숫자)이고, v는 벡터입니다. 스칼라 곱에서도 어떤 순서로 곱하든 결과는 같습니다.

- 분배법칙 (Distributive Law)

분배법칙은 벡터에 대한 스칼라 곱 또는 벡터들의 덧셈에 대한 분배를 의미합니다.

스칼라의 벡터 덧셈에 대한 분배법칙

$$a(u+v)=au+av$$

즉, 스칼라 a를 벡터 덧셈에 분배할 수 있습니다.

예시

$$2\times((1,2)+(3,4))=2\times(1,2)+2\times(3,4)$$
$$2\times(4,6)=(2,4)+(6,8)=(8,12)$$

스칼라 덧셈에 대한 분배법칙

$$(a+b)v=av+bv$$

스칼라 a와 b가 벡터 v에 각각 분배됩니다.

예시

$$(2+3)\times(1,2)=2\times(1,2)+3\times(1,2)$$
$$5\times(1,2)=(2,4)+(3,6)=(5,10)$$

```
import numpy as np

# 벡터와 스칼라 정의
u = np.array([1, 2])
v = np.array([3, 4])
w = np.array([5, 6])
a = 2
b = 3

# 교환법칙
print("교환법칙: u + v == v + u :", np.array_equal(u + v,

# 결합법칙
print("결합법칙: (u + v) + w == u + (v + w) :", np.array_

# 분배법칙 (스칼라에 대한 분배)
print("분배법칙 1: a * (u + v) == a * u + a * v :", np.ar

# 분배법칙 (스칼라 덧셈에 대한 분배)
print("분배법칙 2: (a + b) * u == a * u + b * u :", np.ar
```

2. 선형 독립성(Linear Independence)

   벡터들이 선형 독립이라는 것은 그 벡터들을 선형 결합하여 0벡터를 만들기 위해서는 각 벡터의 스칼라 계수들이 모두 0이어야 한다는 의미입니다.

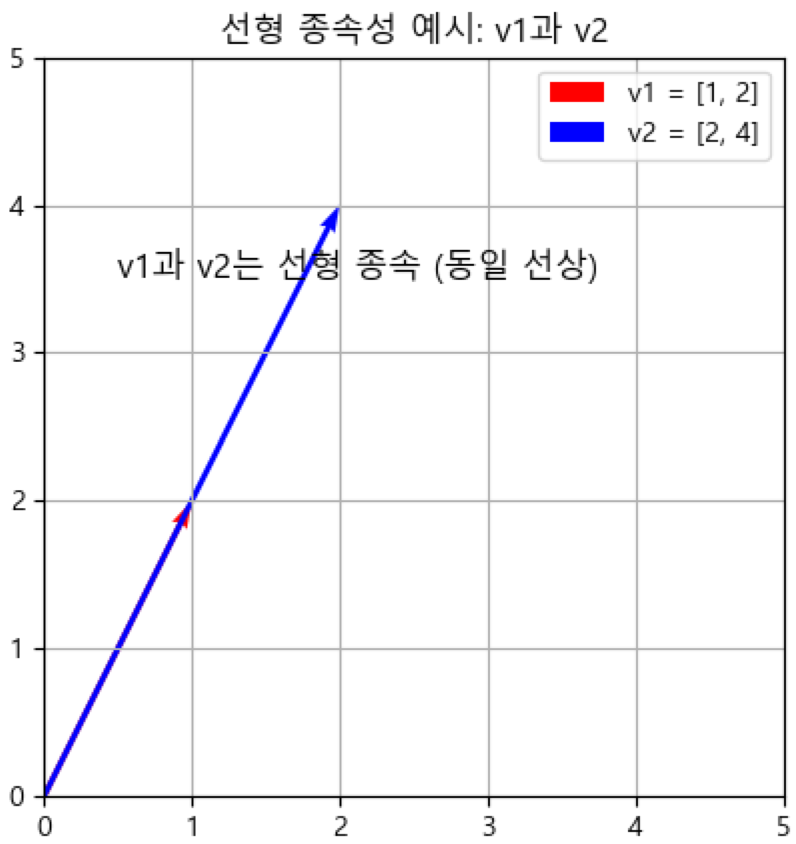
   수학적 정의

   벡터 $v_1, v2, ..., vn$이라면

$$c_1v_1 + c_2v_2 + ... + c_nv_n = 0$$

   일 때, 모든 $c_1 = c_2 = ... = c_n = 0$이어야 합니다. 즉, 0벡터를 만들기 위한 유일한 방법은 모든 계수가 0일 때 입니다.

   반대로 선형 종속이라면 : 한 벡터가 다른 벡터들의 선형 결합으로 표현될 수 있습니다.



3. 기저 (Basis)

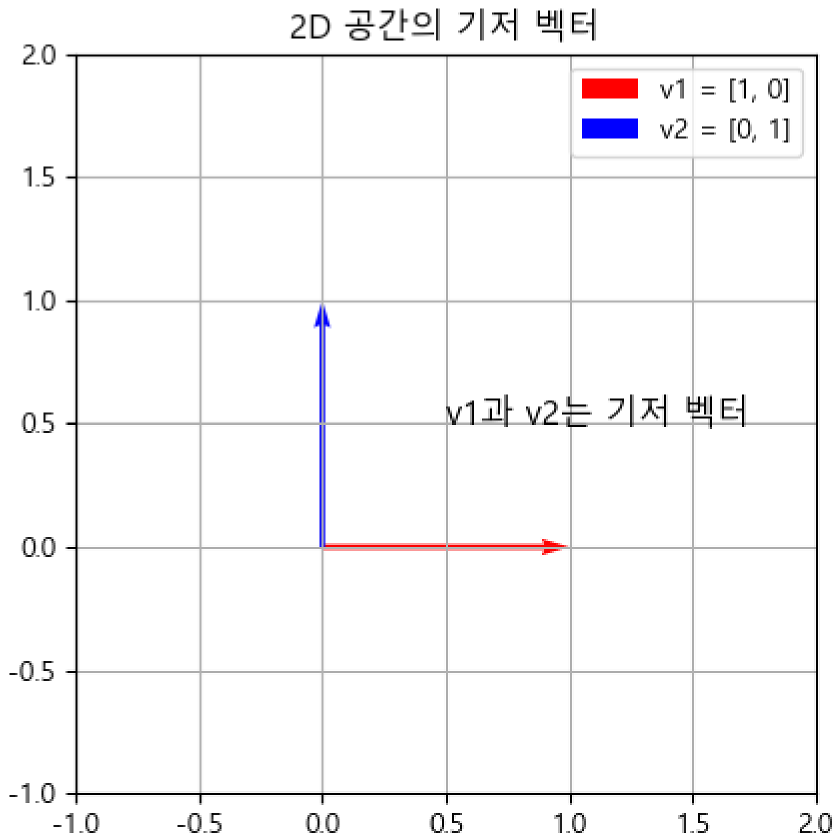
   기저는 벡터 공간 내 모든 벡터를 표현할 수 있는 벡터들의 집합입니다. 이 벡터들은 선형 독립이어야 하며, 기저 벡터들의 선형 결합으로 벡터 공간 내의 모든 벡터를 표현할 수 있습니다.

   예시

   2차원 공간에서 기저는 다음과 같이 나타날 수 있습니다.

$$v_1 = (1, 0), v_2 = (0, 1)$$

   여기서 v_1과 v_2는 서로 선형 독립적이며, 이들을 사용하여 2차원 공간 내의 모든 벡터를 표현할 수 있습니다.




```
import numpy as np

# 두 개의 벡터 정의
v1 = np.array([1, 0])
v2 = np.array([0, 1])

# 벡터들을 행렬로 결합
matrix = np.vstack([v1, v2])

# 선형 독립성을 확인하기 위해 행렬의 랭크 계산
rank = np.linalg.matrix_rank(matrix)

print(f"행렬의 랭크: {rank}")

# 기저인지 확인: 랭크가 벡터의 개수와 같으면 선형 독립
if rank == matrix.shape[0]:
    print("이 벡터들은 선형 독립적이며, 이들은 기저를 이룹니다.")
```

```
else:
    print("이 벡터들은 선형 종속적입니다.")
```