

[N차원 배열 다루기-III]

3. 복사(copy)와 뷰(view)

- (1) 모두 복사 안함(단순 할당): 배열 객체나 데이터의 사본을 만들지 않는다.
- (2) View: 동일한 데이터를 보는 새로운 객체 생성 / **view()**
- (3) copy: 배열 및 해당 데이터의 전체 복사본 생성 / **copy()**

4. 고급 인덱싱

- (1) 인덱스 배열로 인덱싱하기: 배열의 인덱싱을 단일 숫자가 아닌, 넘파이 배열을 이용
ex) `i = np.array[[1, 2], [3, 4]]`
- (2) 다차원 인덱싱: 각 차원에 대한 인덱스 배열은 동일해야 함
ex) `a = array[[1, 2, 3, 4],
 [5, 6, 7, 8]]`
- (3) **argmax**: 최대값이 있는 곳의 인덱스 반환
ex) `ind = data.argmax(axis=0)` # 각 열에서 큰값이 있는 index
- (4) 논리 배열을 이용한 인덱싱: 해당 조건을 입력하여 요소별로 True / False 반환
- (5) **ix_()** 함수: N개의 1차원 시퀀스를 입력받아 추출해서 각각 N차원 N개 반환

5. 선형대수 – 선형 연립 방정식 풀이(회귀도출)에 쓰이는 함수들

- 행렬곱 : **@ / np.dot()**
- 역행렬 : **np.linalg.inv(x)**
- 전치행렬 : **x.T, x.transpose()**
- $n \times n$ 단위행렬 : **np.eye(n)**
- 대각합 : **np.trace(x)**
- 연립방정식 해 풀기 : **np.linalg.solve(a, b)**