

[Python] Numpy 를 통한 최대값, 최소값 , 통계함수 사용하기 — 나무늘보의 개발 블로그

노트북: 첫 번째 노트북

만든 날짜: 2020-10-30 오후 9:46

URL: <https://continuous-development.tistory.com/124?category=736681>

Python

[Python] Numpy 를 통한 최대값, 최소값 , 통계함수 사용하기

2020. 10. 13. 18:53 수정 삭제 공개

Numpy를 통한 다양한 함수 사용하기

- 최대/최소 : min, max, argmin, argmax
- 통계 sum, mean, median, std, var

아래와 같은 x 값이 있을 때 (만들어지는 방법은 다른 것을 가져온 것으로 신경 안 쓰셔도 됩니다.)

이 x 값에 대해 다양한 함수를 사용해서 값을 가져와 보자.

```
In [46]: print('최소값 인덱스:', x.argmin())
print('최대값 인덱스:', x.argmax())
print('최소값 :', x[x.argmin()])
print('최대값 :', x[x.argmax()])
print('중위수 값 :', np.median(x))
print('평균 값', np.mean(x))
print('합 값:', np.sum(x))
```

```
최소값 인덱스: 0
최대값 인덱스: 3
최소값 : 1
최대값 : 4
중위수 값 : 2.5
평균 값 2.5
합 값: 10
```

argmin - 최소값의 인덱스 값을 가져온다.

argmax - 최대값의 인덱스 값을 가져온다.

x [x.argmin] - x에 인덱스 값을 넣어줌으로써 최소값을 가져온다.

x [x.argmax] - x에 인덱스 값을 넣어줌으로써 최대값을 가져온다.

np.median(value) - 해당 value 의 중위값을 가져온다.

np.mean(value) - 해당 value 의 평균값을 가져온다.

np.sum(value) - 해당 value 의 합을 가져온다.

np.std(value) - 해당 value 의 표준편차를 가져온다.

np.var(value) - 해당 value 의 분산을 가져온다.

all / any 로 boolean 하기

all 은 and 조건으로 boolean 연산을 하고

any는 or 조건으로 boolean 연산을 한다.

```
In [51]: # all 은 and / any 는 or 0이다.
all_matrix = np.zeros((100,100), dtype=np.int)
aryInfo(all_matrix)
print(np.all(all_matrix==all_matrix))
```

```
type : <class 'numpy.ndarray'>
shape : (100, 100)
dimension : 2
dtype : int32
Array Data :
[[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]
True
```

```
In [59]: x_vector = np.array([1,2,3,2])
y_vector = np.array([2,2,3,2])
z_vector = np.array([6,4,4,5])

# all is and
print((x_vector>= y_vector).all())

# any is or 0이다.
print((x_vector>= y_vector).any())

False
True
```

axis를 통한 행열 연산(연산의 대상이 2차원 일 때)

- axis = 0 열 연산
- axis = 1 행 연산(생략될 경우 디폴트 값으로 0)

```
In [64]: x_matrix = np.arange(1,21,1).reshape(4,-1)
```

```
In [66]: aryInfo(x_matrix)

type : <class 'numpy.ndarray'>
shape : (4, 5)
dimension : 2
dtype : int32
Array Data :
[[ 1  2  3  4  5]
 [ 6  7  8  9 10]
 [11 12 13 14 15]
 [16 17 18 19 20]]
```

위의 형태의 x_matrix 가 있을 때

```
In [68]: # 열 별로 sum 을 한다.
x_matrix.sum(axis=0)

Out[68]: array([34, 38, 42, 46, 50])
```

sum을 하는 데 있어서 axis를 통해 방향을 정한다. 0 일 때는 열별로 sum을 한다.

```
In [69]: # 행 별로 sum 을 한다.
x_matrix.sum(axis=1)

Out[69]: array([15, 40, 65, 90])
```

1 일때는 행별로 sum을 한다.

예제)

```
In [74]: x_matrix = np.arange(1,31,1).reshape(5,-1)
```

```
In [75]: x_matrix
```

```
Out[75]: array([[ 1,  2,  3,  4,  5,  6],
               [ 7,  8,  9, 10, 11, 12],
               [13, 14, 15, 16, 17, 18],
               [19, 20, 21, 22, 23, 24],
               [25, 26, 27, 28, 29, 30]])
```

```
In [86]: # 전체의 최댓값
print(x_matrix.max())

# 각 행의 합
print(x_matrix.sum(axis=1))
print(np.sum(x_matrix,axis=1))

# 각 행의 최댓값
print(x_matrix.max(axis=1))
print(np.max(x_matrix,axis=1))

# 각 열의 평균
print(x_matrix.mean(axis=0))
print(np.mean(x_matrix,axis=1))

#각 열의 최댓값
print(x_matrix.max(axis=0))
print(np.sum(x_matrix,axis=1))
```

```
30
[ 21  57  93 129 165]
[ 21  57  93 129 165]
[  6  12  18  24  30]
[  6  12  18  24  30]
[13.  14.  15.  16.  17.  18.]
[ 3.5  9.5 15.5 21.5 27.5]
[25 26 27 28 29 30]
[ 21  57  93 129 165]
```

'Python' 카테고리의 다른 글

[Python] Numpy를 통한 난수생성, 카운팅, 통계함수 사용법

[Python] Numpy를 통한 정렬하기

[Python] Numpy 를 통한 최대값, 최소값 , 통계함수 사용하기

[Python] Numpy를 통한 배열 연산

[Python] Numpy의 배열 행 열 삭제

[Python] Numpy 배열 합치기(concatenate)

numpy 최대값

numpy 최소값

numpy 함수

numpy 행열 연산



나무늘보스

혼자 끄적끄적하는 블로그 입니다.