

Numpy, Pnadas

Numpy

Scalar

방향 x, 실수 공간에서 크기를 나타내는 값 (상수로 생각하면 편함)

```
np.array([0])  
>> array([0])
```

Vector

n차원 공간에서 방향과 크기를 갖는 단위

```
x1 = np.array([1,2,3])  
x1  
>> array([1,2,3])  
  
# shape = 차원  
x1.shape  
>> (3,) # 3차원
```

Matrix

행과 열로 이루어진 구조

```
A = np.array([[1,2,3],  
              [4,5,6]])  
>> array([[1, 2, 3],  
          [4, 5, 6]])
```

Functions

np.shape = 행렬의 차원을 확인

np.reshape = 행렬의 차원을 변경

⇒ (2,8)차원은 (4,4)차원으로 변환 가능 (3,5)차원을 가진 행렬로는 변환 X

```
A = np.array([[1, 2, 3, 4, 5, 6, 7, 8], [8, 7, 6, 5, 4, 3, 2, 1]])
>> array([[1, 2, 3, 4, 5, 6, 7, 8],
          [8, 7, 6, 5, 4, 3, 2, 1]])

np.reshape(A, (4, 4)) # 4x4차원으로 변경
>> array([[1, 2, 3, 4],
          [5, 6, 7, 8],
          [8, 7, 6, 5],
          [4, 3, 2, 1]])
```

np.concatenate() = 행렬을 특정방향으로 이어 붙이기

```
sample_arr = np.reshape(A, (4, 2))
>> array([[0, 1],
          [2, 3],
          [4, 5],
          [6, 7]])

# 행 방향으로 붙이기
np.concatenate([sample_arr, sample_arr, sample_arr], axis=0).shape
>> (12, 2)

# 열 방향으로 붙이기
np.concatenate([sample_arr, sample_arr, sample_arr], axis=1).shape
>> (4, 6)
```

Pandas

Series

라벨(colname)을 가진 1차원 배열

```
pd.series(np.array([1,2,3,4,5,6,7,8]), name = '변수명')
>> 0    1
    1    2
    2    3
    3    4
    4    5
    5    6
    6    7
```

```
7      8
Name: 변수 이름, dtype: int32
```

DataFrame

라벨(colname)을 가진 2차원 배열

```
DF = pd.DataFrame([[1, 2, 3], [4, 5, 6], [7, 8, 9]], columns=['Var1', 'Var2', 'Var3'])
>> Var1 Var2 Var3
0 1 2 3
1 4 5 6
2 7 8 9
```

Functions

DataFrame.shape = 데이터프레임의 차원을 확인

차원을 변경하는 reshape 함수 x

차원을 변경할 때는 numpy로 변환 후 np.reshape 사용하여 변경

```
df.shape # 차원확인
df.head() # 값 일부분 반환
df.info() # 객체가 가진 변수들의 정보 보여줌
df.describe() # 기초통계량
df.describe(include = 'all') # 범주형까지 기초통계량
df.value() # numpy배열로 반환

# concat
pd.concat([a, b], axis = 0) # 행방향
pd.concat([a, b], axis = 1) # 열방향
```

DataFrame.loc

DataFrame.loc[행조건, 열조건]

DataFrame.iloc

DataFrame.iloc[행 인덱스, 열 인덱스]

```
DF.iloc[[0,2,4,6,8], [1,3,5,7,9]]
# 0,2,4,6,8번째 행 1,3,5,7,9번째 열
```