

#01. 준비과정

[1]패키지 참조

```
import numpy as np
```

#02. 기초 통계량 산출(최대, 최소 합계 평균 등)

[1] 기본 통계 기호

기호	LaTex	의미
Ω	<code>\Omega</code>	모집단
μ	<code>\mu</code>	모평균(모집단의 평균, 모수)
σ	<code>\sigma</code>	모표준편차(모집단의 표준편차)
σ^2	<code>\sigma^2</code>	모표준편차(모집단의 표준편차)
\bar{X}	<code>\bar{X}</code>	표본평균(표본집단의 평균)
s	<code>s</code>	표본표준편차(표본집단의 표준편차)
s^2	<code>s^2</code>	표본분산(표본집단의 표준분산)
n	<code>n</code>	표본크기(표본집단의 데이터 수)
\sum	<code>\sum</code>	합계

[1] 1차원 데이터의 경우

(1) 샘플 데이터

```
myarr = np.array([98, 72, 80, 64])
myarr
```

```
array([98, 72, 80, 64])
```

(2) 최대, 최소, 합계, 평균값

```
print("최대값:", np.max(myarr))
print("최소값:", np.min(myarr))
print("합계:", np.sum(myarr))
print("평균:", np.average(myarr))
```

최대값: 98
 최소값: 64
 합계: 314
 평균: 78.5

(3) 분산, 표준편차

구분	설명
편차	관측값(y) (=실제 값, 조사된 값)에서 평균을 뺀 값
분산	편차 제곱의 평균

편차

관측값 (y) (=실제 값, 조사된 값)에서 평균(\bar{y})을 뺀 값

$$d = y - \bar{y}$$

분산

편차 제곱의 평균

$$\bar{d^2}$$

```
print("분산:", np.var(myarr))
print("표준편차:", np.std(myarr))
```

분산: 158.75
 표준편차: 12.599603168354152

#02. 연산기능

[1] 연산자를 사용한 배열간의 연산

```
arr1 = np.array([10, 15, 20, 25, 30])
arr2 = np.array([2,3,4,5,6])
print(arr1)
print(arr2)
```

```
[10 15 20 25 30]
[2 3 4 5 6]
```

```
a = arr1 + arr2
a
```

```
array([12, 18, 24, 30, 36])
```

```
b = arr1 - arr2
b
```

```
array([ 8, 12, 16, 20, 24])
```

```
c = arr1 * arr2
c
```

```
array([ 20, 45, 80, 125, 180])
```

```
d = arr1/arr2
d
```

```
array([5., 5., 5., 5., 5.])
```

[2] Numpy 패키지의 함수 사용

```
arr1 = np.array([10, 15, 20, 25, 30])
arr2 = np.array([2,3,4,5,6])
```

```
print(arr1)
print(arr2)
```

```
[10 15 20 25 30]
[2 3 4 5 6]
```

```
a = np.add(arr1, arr2)
a
```

```
array([12, 18, 24, 30, 36])
```

```
b = np.subtract(arr1, arr2)
b
```

```
array([ 8, 12, 16, 20, 24])
```

```
c = np.multiply(arr1, arr2)
c
```

```
array([ 20, 45, 80, 125, 180])
```

```
d = np.divide(arr1, arr2)
d
```

```
array([5., 5., 5., 5., 5.])
```

#03. 기초 통계량 산출(최대, 최소, 합계, 평균 등)

[1] 1차원 데이터의 경우

(1) 샘플 데이터

```
myarr = np.array([98, 72, 80, 64])
```

```
myarr
```

```
array([98, 72, 80, 64])
```

(2) 최대, 최소, 합계, 평균값

```
print("최대값:", np.max(myarr))
print("최소값:", np.min(myarr))
print("합계:", np.sum(myarr))
print("평균:", np.average(myarr))
```

```
최대값: 98
최소값: 64
합계: 314
평균: 78.5
```

(3) 분산, 표준편차

편차(d)

관측값(y) (=실제값, 조사된 값)에서 관측값의 평균을 뺀 값

$$d = y - \bar{y}$$

관측치가 평균과 얼마만큼의 차이가 있는지 알아보기 위한 값이다. 편차의 총 합은 항상 0이다.

분산 편차 제곱의 평균 $\sigma^2 = \bar{d^2}$

각 데이터가 평균에서 얼마나 멀리 떨어져 있는가를 측정하는 값

편차는 총 합이 0이기 때문에 평균을 뺀 경우 큰 의미가 없다. 또한 분산은 관측치가 평균으로부터 얼마나 멀리 떨어져 있는가를 의미하는 값이기 때문에 양수인가 음수인가는 그다지 중요하지 않다.

따라서 편차의 합이 0이 되는 문제를 해결하기 위하여 편차의 음의 부호를 제거하는 방법이 필요한데 그 대표적인 값이 절대값과 제곱이다.

일반적으로 절대값은 기호 사용의 불편함과 함께 식의 계산 또는 변형이 어려워 식이 가지는 성질을 찾아내기가 어렵기 때문에 잘 사용하지 않고 제곱을 사용하여 편차²의 총 합을 변량의 개수로 나눈 값을 구한다.

```
print("분산:", np.var(myarr))
```

```
분산: 158.75
```

표준편차 분산만으로도 데이터의 산포도를 확인할 수 있지만 제곱을 했기 때문에 관측치보다 매우 큰 값이 된다. 그러므로 관측치와 단위를 같게 하기 위해 다시 제곱근을 구할 필요가 있는데 이 값이 표준편차 이다.

$\sigma =$

#04. 조건에 맞는 값 추출하기

[1] 원하는 위치의 값 추출

```
# 철수의 1학년~4학년까지의 평균 점수
grade = np.array([82, 77, 91, 88])
```

```
# -> 추출하고자 하는 원본과 같은 사이즈의 배열 생성
# -> 추출할 값은 True, 그렇지 않으면 False
bool_array = np.array([True, False, True, False])
bool_array
```

```
array([ True, False,  True, False])
```

```
# -> 조건에 맞는 항목만 1차 배열로 추출
result1 = grade[bool_array]
result1
```

```
array([82, 91])
```

[2] 조건에 맞는 원소 추출

80점 이상인 원소를 추출하시오

```
result2 = grade[grade >= 80]
result2
```

```
array([82, 91, 88])
```

80점 이상이고 90점 이하인 데이터만 추려냄

```
result3 = grade[np.logical_and(grade ≥ 80, grade ≤ 90)]  
result3
```

```
array([82, 88])
```

80점 미만이거나 90점 초과인 데이터만 추려냄

```
result4 = grade[np.logical_or(grade ≥ 80, grade ≤ 90)]  
result4
```

```
array([82, 77, 91, 88])
```