

numpy 를 이용한 데이터 생성 및 통계

- 수열과 그래프 표현
- 기본 통계
- random 함수들

베어드교양대학
강의선
백마관 203호
02-828-7264
iami86@ssu.ac.kr

수열 생성

- **np.arange(a, b, c)**
 - a : 시작값
 - b : 종료값
 - c : 증가값
 - 실수 값 지원
 - list가 아니라 array 형태로 생성
- **np.linspace(a, b, c)**
 - a : 시작값
 - b : 종료값
 - c : 구분수를 통한 Array 생성 (등급의 개수)
 - 100으로 하면 100개의 구간으로 나누는 수열 생성

range()와 arange()의 차이점

- 공통점
 - (시작값, 종료값, 증감값) 동일
- 차이점
 - range() : 시작값, 종료값, 증감값에 실수 지원 안함
 - arange() : 시작값, 종료값, 증감값에 실수 지원

```
1 list(range(1, 10, 1))  
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
1 list(range(1.5, 10.5, 2.1))
```

```
TypeError                                Traceback (most recent call last)  
<ipython-input-11-7f0a3141493f> in <module>  
----> 1 list(range(1.5, 10.5, 2.1))
```

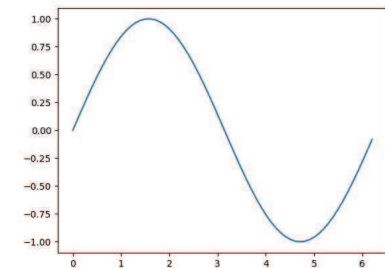
```
TypeError: 'float' object cannot be interpreted as an integer
```

```
1 list(np.arange(1, 10, 1))  
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
1 list(np.arange(1.5, 10.5, 2.1))  
[1.5, 3.6, 5.7, 7.800000000000001, 9.9]
```

arange()를 이용한 Sin 함수 그리기

```
import numpy as np  
import matplotlib.pyplot as plt  
  
xlist = np.arange(0, 3.14*2, 0.1)  
ylist = np.sin( xlist )  
  
plt.plot(xlist, ylist)  
plt.show()
```



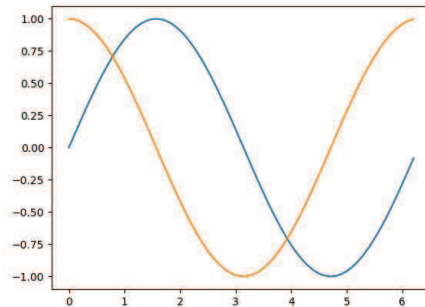
Sin, Cos 함수 그리기

```
import numpy as np
import matplotlib.pyplot as plt

x = np.arange(0, 3.14*2, 0.1)
y1 = np.sin(x)
y2 = np.cos(x)

plt.plot(x, y1)
plt.plot(x, y2)

plt.show()
```



5

Sin, Cos 자세히 그리기

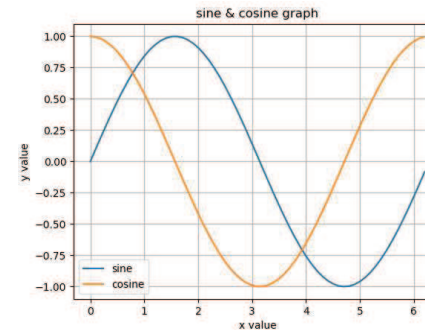
```
import numpy as np
import matplotlib.pyplot as plt

x = np.arange(0, 3.14*2, 0.1)
y1 = np.sin(x)
y2 = np.cos(x)

plt.plot(x, y1, label='sine')
plt.plot(x, y2, label='cosine')

plt.title('sine & cosine graph')
plt.xlabel('x value')
plt.ylabel('y value')
plt.grid(True)
plt.legend()

plt.show()
```



6

np.linspace() 함수

np.linspace(a, b, c)

- a : 시작값 #정수, 실수
- b : 종료값 #종료값을 포함한 정수, 실수
- c : 구분수를 통한 Array 생성 (등급의 개수) #정수
✓100으로 하면 100개의 구간으로 나누는 수열 생성

ex) 1부터 10까지의 범위를 10등분하기

```
In [5]: import numpy as np
```

```
In [10]: a = np.linspace(1, 10, 10)
```

```
In [11]: a
```

```
Out[11]: array([ 1.,  2.,  3.,  4.,  5.,  6.,  7.,  8.,  9., 10.])
```

7

np.linspace 함수 이해하기

0부터 25까지의 범위를 30등분하기

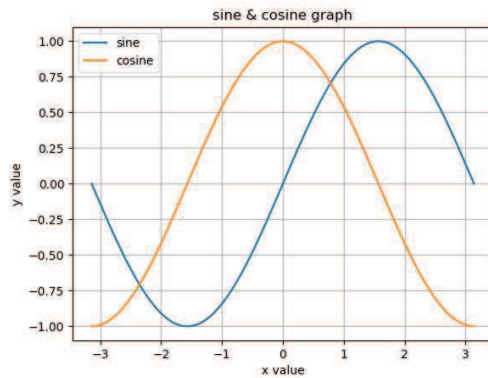
```
In [14]: a = np.linspace(0, 25, 30)
```

```
In [15]: a
```

```
Out[15]: array([ 0.,  0.86206897,  1.72413793,  2.58620689,  3.44827586,
  4.31034483,  5.17241379,  6.03448276,  6.89655172,  7.75862069,
  8.62068966,  9.48275862, 10.34482759, 11.20689655, 12.06896552,
 12.93103448, 13.79310345, 14.65517241, 15.51724138, 16.37931034,
 17.24137931, 18.10344828, 18.96551724, 19.82758621, 20.68965517,
 21.55172414, 22.4137931 , 23.27586207, 24.13793103, 25.])
```

8

np.linspace() 활용



```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(-np.pi, np.pi, 100)
y1 = np.sin(x)
y2 = np.cos(x)

plt.plot(x, y1, label='sine')
plt.plot(x, y2, label='cosine')

plt.title('sine & cosine graph')
plt.xlabel('x value')
plt.ylabel('y value')
plt.legend()
plt.grid(True)
plt.show()
```

9

기본 통계

- 10명의 학생들의 성적을 입력 받으시오.
- 입력 받은 성적에 대하여 아래의 통계, 수학 계산을 수행하시오.

- 합 (Sum)
- 평균 (Average)
- 분산 (Variation)
- 표준편차 (Standard Deviation)
- 중간 값 (Median)
- 최대 값 (Max)
- 최소 값 (Min)

1번 성적입력 -> 1
2번 성적입력 -> 2
3번 성적입력 -> 3
4번 성적입력 -> 4
5번 성적입력 -> 5

입력된 전체점수
[1, 2, 3, 4, 5]

합계 : 15
평균 : 3.00
분산 : 2.00
편차 : 1.41

10

numpy 활용하지 않는 경우 : 1단계

```
scores = []
for i in range(5):
    val = int(input('%d번 성적입력 -> ' % (i+1)))
    scores.append(val)

print('\n입력된 전체점수')
print(scores, '\n')
```

```
# 합계, 평균 구하기
sumR = sum(scores)
aveR = sumR / len(scores)

print('합계 : %10d' % sumR)
print('평균 : %10.2f' % aveR)
```

11

numpy 활용하지 않는 경우 : 2단계

```
scores = []
for i in range(5):
    val = int(input('%d번 성적입력 -> ' % (i+1)))
    scores.append(val)

print('\n입력된 전체점수')
print(scores, '\n')
```

```
# 합계, 평균 구하기
sumR = sum(scores)
aveR = sumR / len(scores)

# 분산, 표준편차
r = 0
for i in range(5):
    r += (scores[i] - aveR)**2
varR = r / len(scores)
stdR = varR**(1/2)

print('합계 : %10d' % sumR)
print('평균 : %10.2f' % aveR)
print('분산 : %10.2f' % varR)
print('편차 : %10.2f' % stdR)
```

12

numpy 활용하는 경우

```
import numpy as np

scores = []
for i in range(5):
    val = int(input('%d번 성적입력 -> ' % (i+1) ))
    scores.append( val )

print('\n입력된 전체점수')
print( scores )
```

```
sumR = np.sum(scores)
aveR = np.mean(scores)
varR = np.var(scores)
stdR = np.std(scores)

print('합계 : %10d' % sumR)
print('평균 : %10.2f' % aveR)
print('분산 : %10.2f' % varR)
print('편차 : %10.2f' % stdR)
```

13

random 함수들 : 난수 생성

- **np.random.rand(a, b)**
 - 0.0~1.0사이의 실수형 난수
 - (a, b) : 난수로 이루어진 2차원 array 생성
b 생략시 난수로 이루어진 1차원 array 생성
- **np.random.randint(a, b, size=(x,y))**
 - a부터 b까지의 정수형 난수 생성
 - size(x,y) : 2차원 array 로 생성
- **np.random.normal(a, b, c)**
 - 정규분포를 갖는 난수 생성
 - a : 평균
 - b : 표준편차
 - c : 생성할 개수

14

random 함수들 : 난수 생성

- **np.random.rand(a, b)**
 - 0.0~1.0사이의 실수형 난수 생성
 - (a, b) : 난수로 이루어진 2차원 array 생성
b 생략시 난수로 이루어진 1차원 array 생성. 즉 a 개수만큼 생성

```
1 np.random.rand(5)
```

```
array([0.04740552, 0.37827322, 0.58528326, 0.33715437, 0.87254352])
```

```
1 np.random.rand(2,3)
```

```
array([[0.78218873, 0.62574872, 0.2548894 ],
       [0.79800068, 0.97700598, 0.93269461]])
```

15

random 함수들 : 난수 생성

- **np.random.rand(a, b)**
 - 특정 범위내에서 난수 생성하기

```
1 a = 10
2 b = 20
3 (b-a)*np.random.rand(5)+a
```

```
array([18.01764315, 12.61751212, 19.05152115, 11.27588819, 17.89533915])
```

16

random 함수들 : 난수 생성

▪ np.random.randint(a, b, size=(x,y))

- a부터 b-1까지의 정수형 난수 생성
- size(x,y) : 난수 생성 개수
2차원 array 로 생성

```
1 np.random.randint(2, 10, size=10)
```

```
array([9, 6, 7, 4, 4, 5, 5, 4, 5, 7])
```

```
1 np.random.randint(1, 11, size=(4,7))
```

```
array([[ 2,  1,  4,  6,  4,  7,  3],  
       [ 5,  4,  4,  2,  9, 10,  8],  
       [ 8,  5, 10,  4,  2,  4,  7],  
       [ 4,  2,  7,  1,  1,  3,  5]])
```

17

random 함수들 : 난수 발생 생성

▪ np.random.normal(a, b, c)

- 정규분포를 갖는 난수 생성
- a : 평균
- b : 표준편차
- c : 생성할 개수

```
1 np.random.normal(55, 5, 10)
```

```
array([46.51276787, 53.99194182, 53.777356 , 59.0822788 , 55.63037258,  
       59.82335208, 61.96516865, 51.35143241, 56.63152502, 53.43816751])
```

18

수고하셨습니다.

19

19