





- ❤ matplotlib 그래프에서 한글 폰트 사용하기 위한 설정
- ✓ 히스토그램 그리는 hist() 함수
- ★ 박스플롯 그리는 boxplot() 함수





- ☑ matplotlib 그래프에서 한글 폰트 사용하기 위한 설정을 수행할 수 있다.
- ☑ 함수 hist() 사용하여 그래프를 그릴 수 있다.
- ☑ 함수 boxplot()를 사용하여 그래프를 그릴 수 있다.

LESSON 01

한글 폰트 사용





⊸ 경고 무시하기

```
import warnings
# 경고 메시지를 무시하고 숨기거나
warnings.filterwarnings(action='ignore')
# 숨기기했던 경고 메시지를 다시 보이게
# warnings.filterwarnings(action='default')
```

```
C:\Users\PC\AppData\Roaming\Python\Python311\site-packages\ma
tplotlib_inline\config.py:68: Deprecation\Warning: InlineBacke
nd._figure_format_changed is deprecated in traitlets 4.1: use
@observe and @unobserve instead.
    def _figure_format_changed(self, name, old, new):
C:\Users\PC\AppData\Roaming\Python\Python311\site-packages\IP
ython\core\pylabtools.py:152: User\Warning: Glyph 52629 (\WN\{HA}
NGUL SYLLABLE CUG\}) missing from current font.
    fig.canvas.print_figure(bytes_io, **kw)
C:\Users\PC\AppData\Roaming\Python\Python311\site-packages\IP
ython\core\pylabtools.py:152: User\Warning: Glyph 49440 (\WN\{HA}
NGUL SYLLABLE SEON\}) missing from current font.
fig.canvas.print_figure(bytes_io, **kw)
```



⊸ 한글 폰트 사용하기(1)

❤️ 그래프의 제목, 범례, x축 및 y축 레이블에 한글로 내용을 작성하면 어떻게 될까요?

```
import matplotlib.pyplot as plt
plt.plot([-2, -1, 0, 1, 2, 3], [20, 30, 15, 40, 45, 35], label='선 1')
plt.plot([-2, -1, 0, 1, 2, 3], [25, 10, 35, 20, 15, 25], label='선 2')
plt.title('선 그래프 예제')
plt.xlabel('x축')
plt.ylabel('y축')
                                                              0 000 00
plt.legend()
                                       45
                                               П1
plt.show()
                                               □ 2
                                       40
                                       35
                                       30
                                     모
                                       25
                                       20
                                       15
                                       10
```

XΠ



- ⊸ 한글 폰트 사용하기(2)
- ❤️ Malgun Gothic은 "맑은 고딕" 확인
 - ☑ 지원 폰트 확인

```
U약 macOS 운영체제를 사용하고 있다면 "AppleGothic"이라고 쓰세요.

In [4]: import matplotlib as mpl from matplotlib import font_manager

In [6]: font_list = sorted([f.name for f in font_manager.fontManager.ttflist])
    'Malgun Gothic' in font_list

Out[6]: True
```

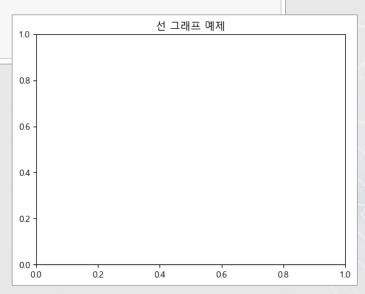
☑ 한글 폰트 사용



→ 한글 폰트 사용하기(3)

```
import matplotlib.pyplot as plt
# 다음 두 문장 중 하나 사용, rc: runtime configuration
plt.rc('font', family='Malgun Gothic')
plt.rcParams['font.family'] = 'Malgun Gothic'
```

plt.title('선 그래프 예제') plt.show()

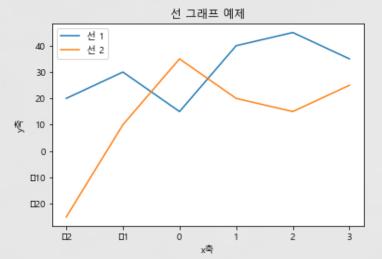




⊸ 한글 폰트 사용하기(4)

```
import matplotlib.pyplot as plt

plt.rc('font', family='Malgun Gothic')
plt.plot([-2, -1, 0, 1, 2, 3], [20, 30, 15, 40, 45, 35], label='선 1')
plt.plot([-2, -1, 0, 1, 2, 3], [-25, 10, 35, 20, 15, 25], label='선 2')
plt.title('선 그래프 예제')
plt.xlabel('x축')
plt.ylabel('y축')
plt.legend()
plt.show()
```



한글로 작성했던 내용들은 잘 표시되지만, 마이너스 부호가 제대로 표시되지 않습니다.



→ 한글 폰트 사용하기(5)

❤️ 한글 폰트 사용시 마이너스 부호 표현하기

```
import matplotlib.pyplot as plt

plt.rc('font', family='Malgun Gothic')
plt.rcParams['font.family'] = 'Malgun Gothic'
plt.rc('axes', unicode_minus=False)
plt.rcParams['axes.unicode_minus'] = False
```



⊸ 한글 폰트 사용하기(6)

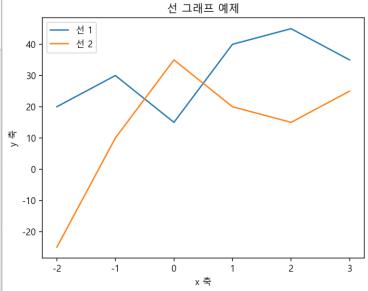
```
import matplotlib.pyplot as plt

plt.rc('font', family='Malgun Gothic')
plt.rcParams['axes.unicode_minus'] = False
plt.plot([-2, -1, 0, 1, 2, 3], [20, 30, 15, 40, 45, 35], label='선 1')
plt.plot([-2, -1, 0, 1, 2, 3], [-25, 10, 35, 20, 15, 25], label='선 2')
plt.title('선 그래프 예제')
plt.xlabel('x축')
plt.ylabel('y축')
plt.legend()
plt.show()

40

40

41
```





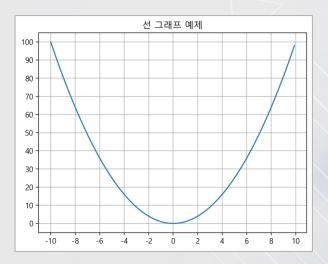
⊸ x, y축 눈금과 내부 그리드 선(1)

```
import numpy as np
In [54]:
         x = np.arange(-10, 10, 0.1)
         plt.plot(x, x**2)
         plt.xticks(np.arange(-10, 11, 2)) # X 左글
         plt.yticks(np.arange(0, 101, 20)) # Y 左글
         # plt.grid(True)
         plt.rc('font', family='Malgun Gothic')
         plt.title('선 그래프 예제')
                                                                           선 그래프 예제
Out[54]: Text(0.5, 1.0, '선 그래프 예제')
                                                    100 -
                                                     90
                                                     80
                                                     70
                                                     60
                                                     50
                                                     40 -
                                                     30
                                                     20
                                                     10 -
                                                         -10
                                                                                              6 8
```



→ x, y축 눈금과 내부 그리드 선(2)

```
import numpy as np
import matplotlib.pyplot as plt
plt.rc('font', family='Malgun Gothic')
plt.rcParams['axes.unicode_minus'] = False
x = np.arange(-10, 10, 0.1)
plt.plot(x, x**2)
plt.xticks(np.arange(-10, 11, 2)) # X 눈금
plt.yticks(np.arange(0, 101, 10)) # Y 눈금
plt.grid(True)
plt.title('선 그래프 예제')
plt.show()
```



LESSON 02

hist()와 boxplot() 그리기를



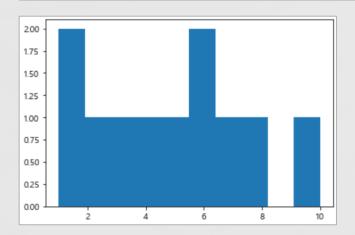


→ 히스토그램(Histogram)

- ❤️ 데이터의 분포 상태를 직사각형 모양의 도수 분포 그래프
 - ☑ 데이터의 빈도에 따라 직사각형의 높이(빈도 수)가 결정
 - ✓ hist() 함수 사용

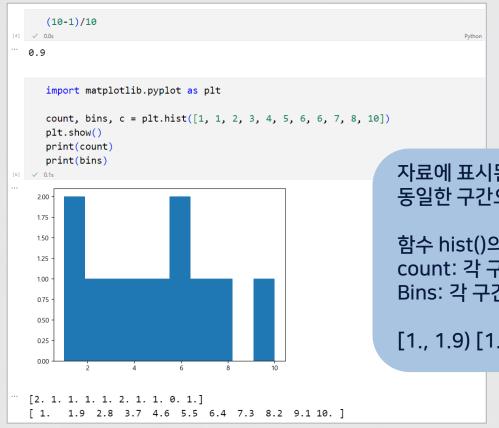
```
import matplotlib.pyplot as plt
plt.hist([1, 1, 2, 3, 4, 5, 6, 6, 7, 8, 10])
plt.show()
```

자료에 표시된 최대값에서부터 최대값까지 10개의 동일한 구간으로 나누어 각각의 구간의 빈도를 표시





⊸ 반환 값 의미



자료에 표시된 최대값에서부터 최대값까지 10개의 동일한 구간으로 나누어 각각의 구간의 빈도를 표시

함수 hist()의 반환값에 모두 저장 count: 각 구간의 빈도 수 배열 Bins: 각 구간을 나누는 수 배열

[1., 1.9) [1.9, 2.8) ··· [8.2, 9.1), [9.1, 10.]



→ 매개변수 bins



√ 0.1s

🌶 구간을 표시하는 배열을 지정

✓ np.arange(0.5, 11.5)

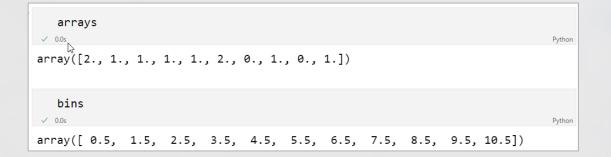
```
import matplotlib.pyplot as plt import numpy as np

data = [1, 1, 2, 3, 4, 5, 6, 6, 8, 10] arrays, bins, patches = plt.hist(data, bins=np.arange(0.5, 11.5), rwidth=.9) plt.xticks(np.arange(11)) # x축 눈금 모두 표시 plt.show()
```

2.00 -

1.50

1.25



Python

조양미래대학교 인공지능소프트웨어학과

→ 주사위 시뮬레이션

✓ Step 1) 주사위를 굴린다.

1과 6사이의 랜덤 숫자를 만든다.

☑ Step 2) 나온 결과를 기록한다.

리스트에 저장한다.

☑ Step 3) Steps 1-2의 과정을 n번 반복한다.

for 반복문

✓ Step 4) 주사위의 눈이 나온 횟수를 히스토그램으로 그린다.



- → 함수 randint()
- ≥ 1과 6사이의 랜덤 숫자 만들기

import random

print(random.randint(1, 6))



⊸ 상자 그림(Boxplot)

→ 주사위 시뮬레이션 5회 수행하기

```
import random as rd

dice = []
for i in range(5):
    dice.append(rd.randint(1, 6))
print(dice)

dice = [rd.randint(1, 6) for i in range(5)]
print(dice)
```

Python

```
[3, 3, 2, 4, 4] [2, 3, 5, 1, 3]
```

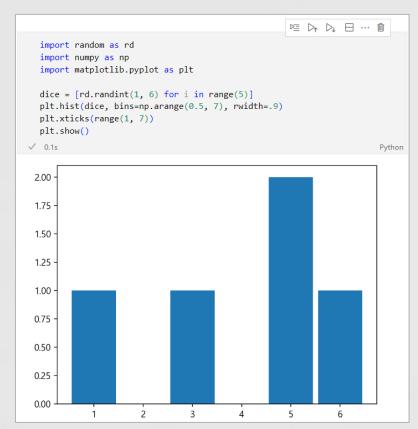
매 수행마다 값이 달라집니다.



⊸ 주사위 n회 모의 실험



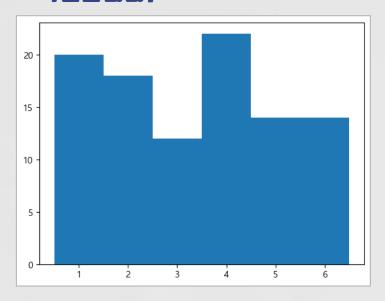
◇ 주사위 시뮬레이션 5회 수행 결과를 히스토그램으로 시각화하기

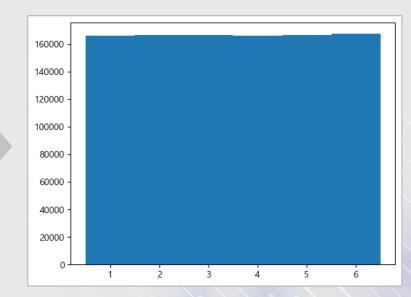


[[hist()와 boxplot() 그리기



⊸ 회수를 늘리면 확률에 근접





✓ 주사위를 던지는 횟수가 늘어날 수록 주사위의 특정 숫자가 나오는 횟수가 전체의 1/6에 가까워 짐을 알 수 있습니다.

큰 수의 법칙



⊸ 상자 그림(Boxplot)

● 데이터에서 얻어낸 최대값, 최소값, 상위 1/4, 2/4 (중앙), 3/4에 위치한 값을 보여주는 그래프

```
최대값
   import matplotlib.pyplot as plt
                                                                  800
   import random
   result = [random.randint(1, 1000) for i in range(13)]
                                                                  600
                                                                       3/4값
  print(sorted(result))
  plt.boxplot(result)
                                                                                                  2/4값
                                                                  400
  plt.show()

√ 0.0s

                                                                  200 -
                                                                        1/4값
[21, 34, 96, 165, 211, 264, 426, 513, 527, 539, 562, 858, 869]
                                                                                                 최소값
```

[전문] hist()와 boxplot() 그리기



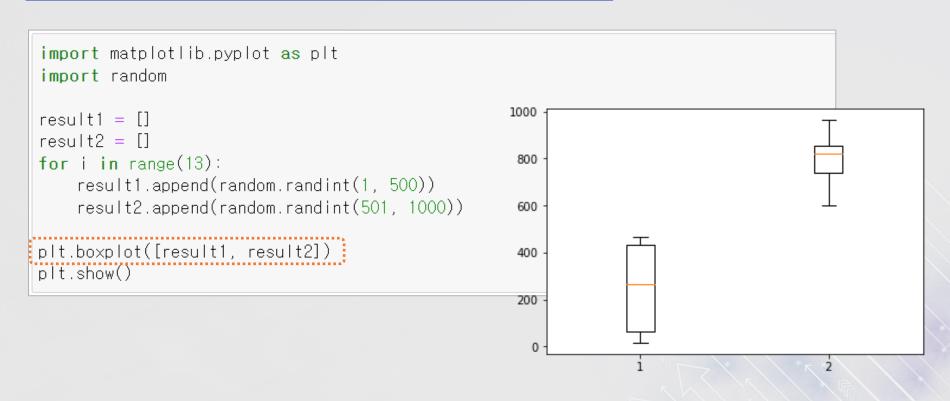
→ 하나의 그래프에서 두 데이터의 상자 그림 그리기(1)



[전문] hist()와 boxplot() 그리기



→ 하나의 그래프에서 두 데이터의 상자 그림 그리기(2)



SUMMARY

학습정긴





•••

🌣 한글 폰트 설정

```
plt.rc('font', family='Malgun Gothic')
plt.rcParams['font.family'] = 'Malgun Gothic'
plt.rc('axes', unicode_minus=False)
plt.rcParams['axes.unicode_minus'] = False
```

♥ 경고 무시 설정

```
import warnings
```

```
# 경고 메시지를 무시하고 숨기거나 warnings.filterwarnings(action='ignore')
```

숨기기했던 경고 메시지를 다시 보이게 warnings.filterwarnings(action='default')







•••

- ♡ 도수 분포표: plt.hist()
 - >> 매개변수 bins=n 또는 리스트나 배열
 - >> 반환 값: 빈도수, 구간 표시 배열
- 학스플롯: plt.boxplot()
 - >> 자료의 분포를 이상치, 최소값, 1/4값, 중앙 값(2/4), 3/4값, 최대값을 박스로 표시



