

Matplotlib(맷플롯립) 실습예제

- 데이터 시각화: 점이나 선, 막대 그래프 등의 시각적 이미지를 사용하여 데이터를 화면에 표시하는 기술
- 차트

1. 선 차트 그리기

```
In [ ]: import matplotlib.pyplot as plt

# 우리나라의 연간 1인당 국민소득을 각각 years, gdp에 저장
years = [1950, 1960, 1970, 1980, 1990, 2000, 2010]
gdp = [67.0, 80.0, 257.0, 1686.0, 6505, 11865.3, 22105.3]

# 선 그래프를 그린다. x축에는 years값, y축에는 gdp 값이 표시된다.
plt.plot(years, gdp, color='green', marker='o', linestyle='solid')

# 제목을 설정한다.
plt.title("1인당 GDP")

# y축에 레이블을 붙인다.
plt.ylabel("달러($)")
plt.savefig("gdp_per_capita.png", dpi=600) # png 이미지로 저장 가능
plt.show()
```

[차트에 한글이 보이도록 설정하기]

```
In [ ]: # 한글이 출력되도록 하려면 아래 코드를 먼저 실행해 주어야 함.

import platform
from matplotlib import font_manager, rc
import matplotlib

# '-' 부호가 제대로 표시되게 하는 설정
matplotlib.rcParams['axes.unicode_minus'] = False

# 운영체제 마다 한글이 보이게 하는 설정
# 윈도우
import platform
from matplotlib import font_manager, rc

if platform.system() == 'Windows':
    path = "c:\\Windows\\Fonts\\malgun.ttf"
    font_name = font_manager.FontProperties(fname=path).get_name()
    rc('font', family=font_name)
# 맥
elif platform.system() == 'Darwin':
    rc('font', family='AppleGothic')
# 리눅스
elif platform.system() == 'Linux':
    rc('font', family='NanumBarunGothic')
```

y = 2x 함수 그리기

```
In [ ]: # x값은 -10 ~ 10까지 정수로 설정, y는 2x 값으로 설정하기

x = [x for x in range(-10, 10)]
y = [2*t for t in x]          # 2*x를 원소로 가지는 y 함수

plt.plot(x, y, marker='o')    # 선 그래프에 동그라미 표식을 출력

plt.axis([-15, 15, -20, 20]) # 그림을 그릴 영역을 지정함
plt.show()
```

```
In [ ]: # x,y값을 아래처럼 리스트 생성해도 됨.

x=[]
y=[]
for i in range(-10,10) :
    x.append(i)
    y.append(2*i)
print(x,y)
```

차트 장식을 도와주는 다양한 기법들

```
In [ ]: # y1 = 2x, y2= x**2 + 5, y3= -x**2 - 5 함수 한꺼번에 그려보기

x = [x for x in range(-20, 20)] # -20에서 20사이의 수를 1의 간격으로 생성
y1 = [2*t for t in x]           # 2*x를 원소로 가지는 y1 함수
y2 = [t**2 + 5 for t in x]      # x**2 + 5를 원소로 가지는 y2 함수
y3 = [-t**2 - 5 for t in x]     # -x**2 - 5를 원소로 가지는 y3 함수

# 빨강색 점선, 녹색 실선과 세모기호, 파랑색 별표와 점선으로 각각의 함수를 표현
plt.plot(x, y1, 'r--', x, y2, 'g^-', x, y3, 'b-.*') # '색, 마크모양, 선모양 지정'

plt.axis([-30, 30, -30, 30])   # 그림을 그릴 영역을 지정함
plt.show()
```

하나의 차트에 여러개 차트 그리

```
In [ ]: x = [x for x in range(20)]    # 0에서 20까지의 정수를 생성
y = [x**2 for x in range(20)]        # 0에서 20까지의 정수 x에 대해 x 제곱값을 생성
z = [x**3 for x in range(20)]        # 0에서 20까지의 정수 x에 대해 x 세제곱값을 생성

plt.plot(x, x, label='x값')          # 각 선에 대한 레이블
plt.plot(x, y, label='x의 제곱값')
plt.plot(x, z, label='x의 세제곱값')

plt.xlabel('x label')                # x 축의 레이블
plt.ylabel('y label')                # y 축의 레이블
plt.title("My Plot")
plt.legend()
plt.show()
```

2. 막대형 차트 그리기

```
In [ ]: from matplotlib import pyplot as plt

# 1인당 국민소득
years = [1950, 1960, 1970, 1980, 1990, 2000, 2010]
gdp = [67.0, 80.0, 257.0, 1686.0, 6505, 11865.3, 22105.3]
```

```
plt.bar(range(len(years)), gdp)

plt.title("1인당 GDP") # 제목을 설정한다.
plt.ylabel("달러($)") # y축에 레이블을 붙인다.

# y축에 틱을 붙인다.
# plt.xticks(range(len(years)), years)
plt.show()
```

여러나라의 국민소득 추이를 다중 막대형 차트로 그리기

```
In [ ]: # 1인당 국민소득
years = [1965, 1975, 1985, 1995, 2005, 2015]
ko = [130, 650, 2450, 11600, 17790, 27250]
jp = [890, 5120, 11500, 42130, 40560, 38780]
ch = [100, 200, 290, 540, 1760, 7940]
```

```
In [ ]: x_range = range(len(years))
plt.bar(x_range, ko, width = 0.25)
plt.bar(x_range, jp, width = 0.25)
plt.bar(x_range, ch, width = 0.25)
```

```
In [ ]: # 공유하는 가로축 범위를 조금씩 차이를 두며 막대그래프 그리기 위해

import numpy as np

x_range = np.arange(len(years)) # 넘파이 배열로 만들어 값을 변경하기

plt.bar(x_range + 0.0, ko, width = 0.25, color='red', label='한국')
plt.bar(x_range + 0.3, jp, width = 0.25, label='일본')
plt.bar(x_range + 0.6, ch, width = 0.25, label='중국')

plt.legend() # 막대그래프에 라벨옵션 주면 범례 표시됨
plt.show()
```

3. 데이터를 점으로 표현하는 산포도(scatter) 그래프 그리기

```
In [ ]: import matplotlib.pyplot as plt
import numpy as np

xData = np.arange(20, 50)
yData = xData + 2 * np.random.randn(30) # xData에 randn() 함수로 잡음을 섞는다.
# 잡음은 정규분포로 만들어 질 것이다.

plt.scatter(xData, yData)
plt.title('실제 나이 vs 신체 나이')
plt.xlabel('실제 나이')
plt.ylabel('신체 나이')

plt.savefig("kkk.png", dpi=600);
plt.show()
```

4. 파이 차트(pie chart) 그리기

- 원형 비율로 나뉜 차트

```
In [ ]: times = [8, 14, 2]
        timelabels = ["Sleep", "Study", "Play"]

        # autopct로 백분율을 표시할 때 소수점 2번째 자리까지 표시하게 한다.
        # labels 매개 변수에 timelabels 리스트를 전달한다.
        plt.pie(times, labels = timelabels, autopct = "%.2f")
        plt.show()
```

5. 히스토그램(histogram) 그리기 - 빈도(frequency) 분석

```
In [ ]: # 예) 학생 8명이 1년 동안 읽은 책을 조사한 결과 빈도분석

        books = [ 1, 6, 2, 3, 1, 2, 0, 2 ]

        # 6개의 빈(bin)을 이용하여 books 안에 저장된 자료의 히스토그램 그리기
        plt.hist(books, bins = 6) # 6개 구간으로 그리기

        plt.xlabel("books")
        plt.ylabel("빈도수")
        plt.show()
```

겹쳐진 히스토그램도 그리자 : 다중 히스토그램

```
In [ ]: import numpy as np
        import matplotlib.pyplot as plt

        n_bins = 10
        x = np.random.randn(1000)
        y = np.random.randn(1000)

        plt.hist(x, n_bins, histtype='bar', color= "red");
        plt.hist(y, n_bins, histtype='bar', color= "blue", alpha=0.3);
        plt.show()
```

```
In [ ]: # [도전] 다음 코드에서 빈의 갯수를 5로 설정해 히스토그램을 그리면...

        x = [1, 2, 4, 4, 4, 6, 6, 8, 9, 9]
        plt.hist(x)
        plt.show()
```

```
In [ ]:
```

정규분포로 생성된 난수를 눈으로 확인하기

- 정규분포 난수 생성 식 = 평균 + 표준편차 * 표준 정규분포에서 추출한 난수
- 예) 평균10, 표준편차 2인 정규분포에서 난수 생성하려면,
- $10 + 2 * \text{np.random.randn}()$

```
In [ ]: # 표준 정규분포(평균0, 표준편차1)로 난수를 만개 생성하고,
        # 평균 10, 표준편차 2인 정규분포로 난수 만개 생성하고,
        # 평균 -6, 표준편차 3인 정규분포로 난수 만개 생성하여
        # 이들이 어떤 분포를 보이는지 히스토그램을 그려보자.

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

```

mu1, sigma1 = 10, 2
mu2, sigma2 = -6, 3

standard_Gauss = np.random.randn(10000)
Gauss1 = mu1 + sigma1 * np.random.randn(10000)
Gauss2 = mu2 + sigma2 * np.random.randn(10000)

plt.figure(figsize=(10,6))
plt.hist(standard_Gauss, bins=50, alpha=0.4)
plt.hist(Gauss1, bins=50, alpha=0.4)
plt.hist(Gauss2, bins=50, alpha=0.4)

plt.show()

```

6. 상자 차트(box chart) 그리기

- 데이터의 최대, 최소, 중간값, 사분위 수 등을 효율적으로 가시화 할 수 있는 차트

```

In [ ]: # 100개의 정규분포 난수를 이용하여 상자 차트 그리기

import numpy as np
import matplotlib.pyplot as plt

random_data = np.random.randn(100)

plt.boxplot(random_data)
plt.show()

```

여러 개의 상자 차트 그리기

```

In [ ]: import numpy as np
import matplotlib.pyplot as plt

data1 = [1, 2, 3, 4, 5]
data2 = [2, 3, 4, 5, 6]

plt.boxplot([ data1, data2 ]) # 리스트 단위로 돌리면 리스트별로 각각 상자가 나옴

```

```

In [ ]: # 넘파일 배열로 상자 차트를 그리면, 열 단위로 각각 상자가 나옴

plt.boxplot(np.array([ data1, data2]))

```

7. 한 화면에 여러 그래프 그리기 - subplots()

- subplots(2,2)는 전체 화면을 2행2열로 나눠서 4개의 차트를 그릴 수 있음
- subplots()의 반환값 중 하나인 ax로 인덱싱하여 각 영역을 구분함.

```

In [ ]: import matplotlib.pyplot as plt

# 갯수가 2 × 2개, 크기가 (5,5)인치 공간 확보
fig, ax = plt.subplots(2, 2, figsize=(5, 5))

ax[0, 0].plot(range(10), 'r') # row=0, col=0
ax[1, 0].plot(range(10), 'b') # row=1, col=0
ax[0, 1].plot(range(10), 'g') # row=0, col=1

```

```
ax[1, 1].plot(range(10), 'k') #row=1, col=1
plt.show()
```

서브플롯 이용하여 차트 만들기 예

```
In [ ]: import matplotlib.pyplot as plt
import numpy as np

np.random.seed(11111)

# 100개의 난수를 정규분포에 따라 생성 (2행 100열)
data = np.random.randn(2, 100)

print(data)

# 화면을 4개로 쪼갬
fig, axs = plt.subplots(2, 2, figsize=(5, 5))

axs[0, 0].hist(data[0])
axs[1, 0].scatter(data[0], data[1])
axs[0, 1].plot(data[0], data[1])
axs[1, 1].hist2d(data[0], data[1])

plt.show()
```