

15장 데이터 시각화

15.1 matplotlib로 그래프 그리기

- matplotlib는 파이썬에서 데이터를 효과적으로 시각화하기 위해 만든 라이브러리이다.
 - <https://matplotlib.org/>

15.1.1 선 그래프

(1) 기본적인 선 그래프 그리기

- matplotlib에서는 다음과 같은 형식으로 2차원 선 그래프를 그린다.
 - `plt.plot([x,] y [,fmt])`: x/y축의 좌표, `fmtt(format string)`으로 다양한 그래프 옵션

[ch15_matplotlib/ex01_line.py]

```
01 import matplotlib.pyplot as plt
02
03 data1 = [10, 14, 19, 20, 25]
04
05 plt.plot(data1)
06 plt.show()
07
08
09 import numpy as np
10
11 x = np.arange(-4.5, 5, 0.5) # 배열 x 생성. 범위: [-4.5, 5), 0.5씩 증가
12 y = 2*x**2 # 수식을 이용해 배열 x에 대응하는 배열 y 생성
13 print([x,y])
14 ...
15 [array([-4.5, -4. , -3.5, -3. , -2.5, -2. , -1.5, -1. , -0.5,  0. ,  0.5,
16         1. ,  1.5,  2. ,  2.5,  3. ,  3.5,  4. ,  4.5]), array([40.5, 32. , 24.5, 18. , 12.5,  8. ,
17         4.5,  2. ,  0.5,  0. ,  0.5,
18         2. ,  4.5,  8. , 12.5, 18. , 24.5, 32. , 40.5])]
19 ...
20
21 plt.plot(x,y)
22 plt.show()
23
24
25
26
```

(2) 여러 그래프 그리기

- 하나의 그래프 창에 여러 개의 데이터를 효과적으로 시각화한다.

[ch15_matplotlib/ex02_lines.py]

```
01 import matplotlib.pyplot as plt
02 import numpy as np
03
04 x = np.arange(-4.5, 5, 0.5)
05 y1 = 2*x**2
06 y2 = 5*x + 30
```

```

07 y3 = 4*x**2 + 10
08
09
10 plt.plot(x, y1)
11 plt.plot(x, y2)
12 plt.plot(x, y3)
13 plt.show()
14
15 plt.plot(x, y1, x, y2, x, y3)
16 plt.show()
17
18 plt.plot(x, y1) # 처음 그리기 함수를 수행하면 그래프 창이 자동으로 생성됨
19 plt.figure() # 새로운 그래프 창을 생성함
20
21 plt.plot(x, y2) # 새롭게 생성된 그래프 창에 그래프를 그림
22 plt.show()
23
24
25 # 데이터 생성
26 x = np.arange(-5, 5, 0.1)
27 y1 = x**2 - 2
28 y2 = 20*np.cos(x)**2 # NumPy에서 cos()는 np.cos()으로 입력
29
30 plt.figure(1) # 1번 그래프 창을 생성함
31 plt.plot(x, y1) # 지정된 그래프 창에 그래프를 그림
32
33 plt.figure(2) # 2번 그래프 창을 생성함
34 plt.plot(x, y2) # 지정된 그래프 창에 그래프를 그림
35
36 plt.figure(1) # 이미 생성된 1번 그래프 창을 지정함
37 plt.plot(x, y2) # 지정된 그래프 창에 그래프를 그림
38
39 plt.figure(2) # 이미 생성된 2번 그래프 창을 지정함
40 plt.clf() # 2번 그래프 창에 그려진 모든 그래프를 지움
41 plt.plot(x, y1) # 지정된 그래프 창에 그래프를 그림
42
43 plt.show()
44
45
46 # 데이터 생성
47 x = np.arange(0, 10, 0.1)
48 y1 = 0.3*(x-5)**2 + 1
49 y2 = -1.5*x + 3
50 y3 = np.sin(x)**2 # NumPy에서 sin()은 np.sin()으로 입력
51 y4 = 10*np.exp(-x) + 1 # NumPy에서 exp()는 np.exp()로 입력
52
53 # 2 × 2 행렬로 이뤄진 하위 그래프에서 p에 따라 위치를 지정
54 plt.subplot(2, 2, 1) # p는 1
55 plt.plot(x, y1)
56
57 plt.subplot(2, 2, 2) # p는 2
58 plt.plot(x, y2)
59
60 plt.subplot(2, 2, 3) # p는 3
61 plt.plot(x, y3)
62
63 plt.subplot(2, 2, 4) # p는 4
64 plt.plot(x, y4)
65
66 plt.show()

```

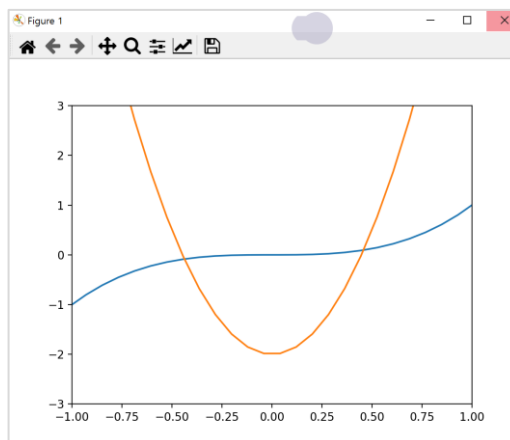
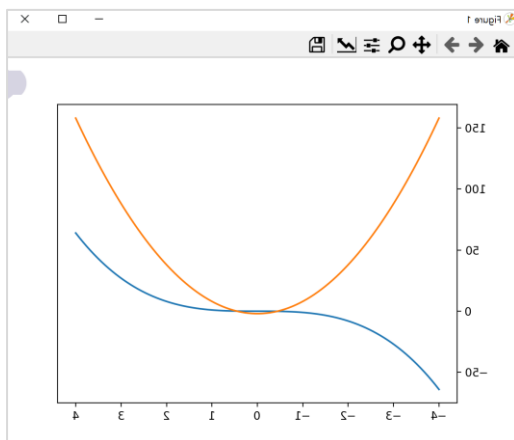
(3) 그래프의 출력 범위 지정하기

■ 출력되는 그래프의 x축과 y축의 좌표 범위를 지정해 전체 그래프 중 관심 영역만 그래프로 그릴 수 있다.

- `plt.xlim(xmin, xmax)` # x축의 좌표 범위 지정(`xmin ~ xmax`)
- `plt.ylim(ymin, ymax)` # y축의 좌표 범위 지정(`ymin ~ ymax`)
- `[xmin, xmax] = plt.xlim()` # x축의 좌표 범위 가져오기
- `[ymin, ymax] = plt.ylim()` # y축의 좌표 범위 가져오기
- 11~14: `xlim()`와 `ylim()`로 축의 범위를 지정해 그래프의 관심 영역을 확대한다.

[ch15_matplotlib/ex03_lim.py]

```
01 import matplotlib.pyplot as plt
02 import numpy as np
03
04 x = np.linspace(-4, 4, 100) # [-4, 4] 범위에서 100개의 값 생성
05 y1 = x**3
06 y2 = 10*x**2 - 2
07
08 plt.plot(x, y1, x, y2)
09 plt.show()
10
11 plt.plot(x, y1, x, y2)
12 plt.xlim(-1, 1)
13 plt.ylim(-3, 3)
14 plt.show()
```



15.1.2 그래프 꾸미기

(1) 출력 형식 지정

■ `plot()`에서 `fmt` 옵션을 이용하면 그래프의 컬러, 선의 스타일, 마커를 지정할 수 있다.

- 형식: `fmt = '[color][line_style][marker]'`

[ch15_matplotlib/ex04_line_style.py]

```
01 import matplotlib.pyplot as plt
02
03 import numpy as np
04 x = np.arange(0, 5, 1)
```

```

05 y1 = x
06 y2 = x + 1
07 y3 = x + 2
08 y4 = x + 3
09
10 plt.plot(x, y1, x, y2, x, y3, x, y4)
11 plt.show()
12
13 plt.plot(x, y1, 'm', x, y2, 'y', x, y3, 'k', x, y4, 'c')
14 plt.show()
15
16 plt.plot(x, y1, '-', x, y2, '--', x, y3, ':', x, y4, '-.')
17 plt.show()
18
19 plt.plot(x, y1, 'o', x, y2, '^', x, y3, 's', x, y4, 'd')
20 plt.show()
21
22 plt.plot(x, y1, '>--r', x, y2, 's-g', x, y3, 'd:b', x, y4, '-.Xc')
23 plt.show()

```

(2) 라벨, 제목, 격자, 범례, 문자열 표시

- 그래프의 x축과 y축의 라벨, 그리고 제목, 격자, 범례, 문자열을 추가할 수 있다.

[ch15_matplotlib/ex05_label.py]

```

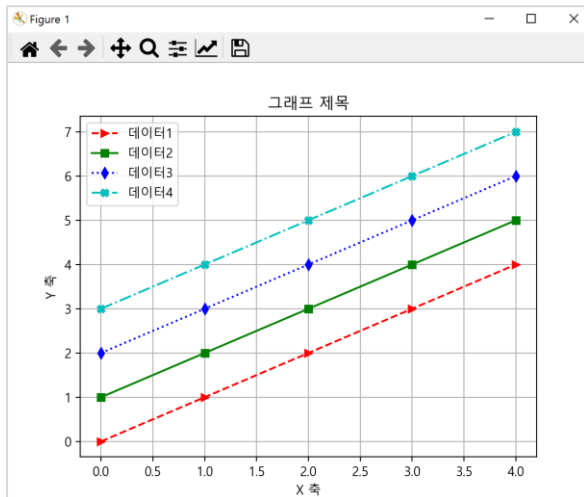
01 import matplotlib
02 import matplotlib.pyplot as plt
03 import numpy as np
04
05 x = np.arange(-4.5, 5, 0.5)
06 y = 2*x**3
07 plt.plot(x, y)
08 plt.xlabel('X-axis')
09 plt.ylabel('Y-axis')
10 plt.show()
11
12 plt.plot(x, y)
13 plt.xlabel('X-axis')
14 plt.ylabel('Y-axis')
15 plt.title('Graph title')
16 plt.show()
17
18 plt.plot(x, y)
19 plt.xlabel('X-axis')
20 plt.ylabel('Y-axis')
21 plt.title('Graph title')
22 plt.grid(True) # 'plt.grid()'도 가능
23 plt.show()
24
25 x = np.arange(0, 5, 1)
26 y1 = x
27 y2 = x + 1
28 y3 = x + 2
29 y4 = x + 3
30 plt.plot(x, y1, '>--r', x, y2, 's-g', x, y3, 'd:b', x, y4, '-.Xc')
31 plt.legend(['data1', 'data2', 'data3', 'data4'])
32 plt.show()
33
34 plt.plot(x, y1, '>--r', x, y2, 's-g', x, y3, 'd:b', x, y4, '-.Xc')
35 plt.legend(['data1', 'data2', 'data3', 'data4'], loc='lower right')

```

```

36 plt.show()
37
38 matplotlib.rcParams['font.family'] = 'Malgun Gothic' # '맑은 고딕'으로 설정
39 matplotlib.rcParams['axes.unicode_minus'] = False
40
41 plt.plot(x, y1, '>--r', x, y2, 's-g', x, y3, 'd:b', x, y4, '-.Xc')
42 plt.legend(['데이터1', '데이터2', '데이터3', '데이터4'], loc='best')
43 plt.xlabel('X 축')
44 plt.ylabel('Y 축')
45 plt.title('그래프 제목')
46 plt.grid(True)
47 plt.show()

```



15.1.3 산점도

- 산점도(scatter 그래프)는 두 개의 요소로 이뤄진 데이터 집합의 관계(예를 들면, 키와 몸무게와의 관계, 기온과 아이스크림 판매량과의 관계, 공부 시간과 시험 점수와의 관계)를 시각화하는데 유용하다.

[ch15_matplotlib/ex06_scatter.py]

```

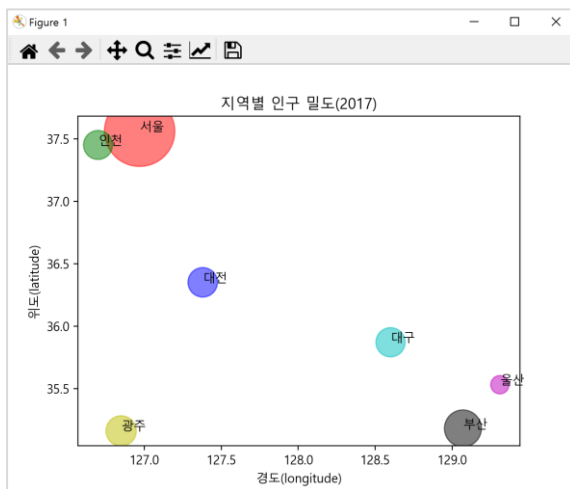
01 import matplotlib.pyplot as plt
02 import numpy as np
03
04 height = [165, 177, 160, 180, 185, 155, 172] # 키 데이터
05 weight = [62, 67, 55, 74, 90, 43, 64] # 몸무게 데이터
06
07 plt.scatter(height, weight)
08 plt.xlabel('Height(m)')
09 plt.ylabel('Weight(Kg)')
10 plt.title('Height & Weight')
11 plt.grid(True)
12 plt.scatter(height, weight, s=500, c='r') # 마커 크기는 500, 컬러는 붉은색(red)
13 plt.show()
14
15
16 size = 100 * np.arange(1, 8) # 데이터별로 마커의 크기 지정
17 colors = ['r', 'g', 'b', 'c', 'm', 'k', 'y'] # 데이터별로 마커의 컬러 지정
18
19 plt.scatter(height, weight, s=size, c=colors)
20 plt.show()

```

```

21
22
23 import matplotlib
24 matplotlib.rcParams['font.family'] = 'Malgun Gothic' # '맑은 고딕'으로 설정
25 matplotlib.rcParams['axes.unicode_minus'] = False
26
27 city = ['서울', '인천', '대전', '대구', '울산', '부산', '광주']
28
29 # 위도(latitude)와 경도(longitude)
30 lat = [37.56, 37.45, 36.35, 35.87, 35.53, 35.18, 35.16]
31 lon = [126.97, 126.70, 127.38, 128.60, 129.31, 129.07, 126.85]
32
33 # 인구 밀도(명/km^2): 2017년 통계청 자료
34 pop_den = [16154, 2751, 2839, 2790, 1099, 4454, 2995]
35
36 size = np.array(pop_den) * 0.2 # 마커의 크기 지정
37 colors = ['r', 'g', 'b', 'c', 'm', 'k', 'y'] # 마커의 컬러 지정
38
39 plt.scatter(lon, lat, s=size, c=colors, alpha=0.5)
40 plt.xlabel('경도(longitude)')
41 plt.ylabel('위도(latitude)')
42 plt.title('지역별 인구 밀도(2017)')
43
44 for x, y, name in zip(lon, lat, city):
45     plt.text(x, y, name) # 위도 경도에 맞게 도시 이름 출력
46
47 plt.show()

```



15.1.4 막대 그래프

- 막대(bar) 그래프는 값을 막대의 높이로 나타내므로 여러 항목의 수량이 많고 적음을 한눈에 알아볼 수 있다.

[ch15_matplotlib/ex07_bar.py]

```

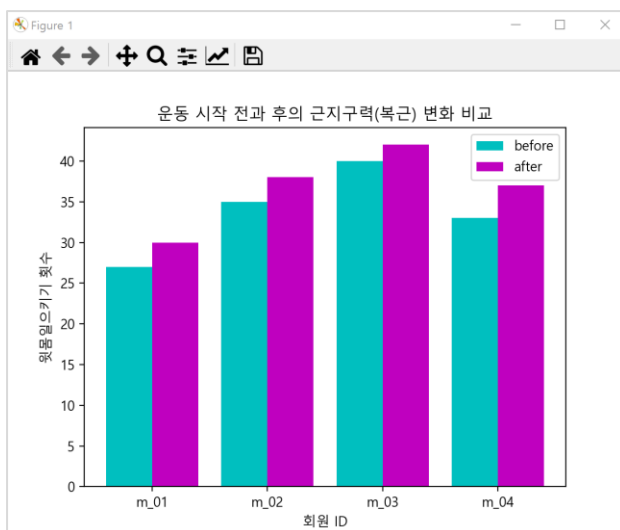
01 import matplotlib
02 import numpy as np
03 import matplotlib.pyplot as plt
04 member_IDs = ['m_01', 'm_02', 'm_03', 'm_04'] # 회원 ID
05 before_ex = [27, 35, 40, 33] # 운동 시작 전
06 after_ex = [30, 38, 42, 37] # 운동 한 달 후
07

```

```

08
09 n_data = len(member_IDs)    # 회원이 네 명이므로 전체 데이터 수는 4
10 index = np.arange(n_data)    # NumPy를 이용해 배열 생성 (0, 1, 2, 3)
11 plt.bar(index, before_ex)    # bar(x,y)에서 x=index, height=before_ex 로 지정
12 plt.show()
13
14
15 plt.bar(index, before_ex, tick_label=member_IDs)
16 plt.show()
17
18
19 colors = ['r', 'g', 'b', 'm']
20 plt.bar(index, before_ex, color=colors, tick_label=member_IDs)
21 plt.show()
22
23
24 plt.bar(index, before_ex, tick_label=member_IDs, width=0.6)
25 plt.show()
26
27
28 colors = ['r', 'g', 'b', 'm']
29 plt.barh(index, before_ex, color=colors, tick_label=member_IDs)
30 plt.show()
31
32
33 matplotlib.rcParams['font.family'] = 'Malgun Gothic'    # '맑은 고딕'으로 설정
34 matplotlib.rcParams['axes.unicode_minus'] = False
35
36 barWidth = 0.4
37 plt.bar(index, before_ex, color='c', align='edge',
38         width=barWidth, label='before')
39 plt.bar(index + barWidth, after_ex, color='m',
40         align='edge', width=barWidth, label='after')
41
42 plt.xticks(index + barWidth, member_IDs)
43 plt.legend()
44 plt.xlabel('회원 ID')
45 plt.ylabel('윗몸일으키기 횟수')
46 plt.title('운동 시작 전과 후의 근지구력(복근) 변화 비교')
47 plt.show()

```

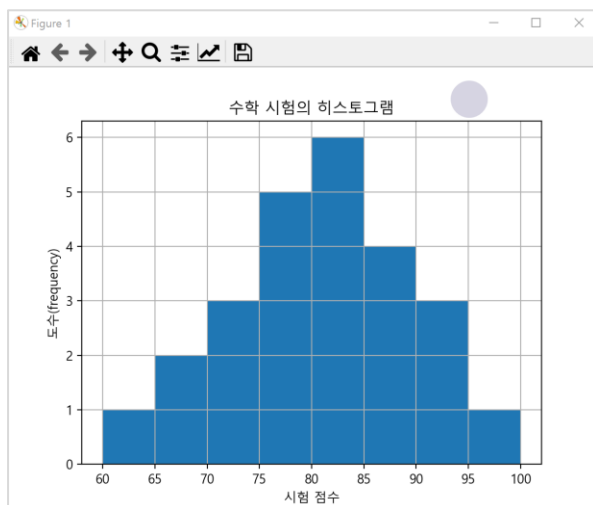


15.1.5 히스토그램

- 히스토그램(histogram)은 데이터를 정해진 간격으로 나눈 후 그 간격 안에 들어간 데이터 개수를 막대로 표시한 그래프로, 데이터가 어떤 분포를 갖는지를 볼 때 주로 이용한다.

[ch15_matplotlib/ex08_histogram.py]

```
01 import matplotlib
02 import matplotlib.pyplot as plt
03 matplotlib.rcParams['font.family'] = 'Malgun Gothic' # '맑은 고딕'으로 설정
04 matplotlib.rcParams['axes.unicode_minus'] = False
05
06 math = [76, 82, 84, 83, 90, 86, 85, 92, 72, 71, 100, 87,
07         81, 76, 94, 78, 81, 60, 79, 69, 74, 87, 82, 68, 79]
08 plt.hist(math)
09
10 plt.hist(math, bins=8)
11 plt.show()
12
13
14 plt.hist(math, bins=8)
15 plt.xlabel('시험 점수')
16 plt.ylabel('도수(frequency)')
17 plt.title('수학 시험의 히스토그램')
18 plt.grid()
19 plt.show()
```



15.1.6 파이 그래프

- 파이(pie) 그래프는 원 안에 데이터의 각 항목이 차지하는 비율만큼 부채꼴의 크기를 갖는 영역으로 이뤄진 그래프이다.

[ch15_matplotlib/ex09_pie.py]

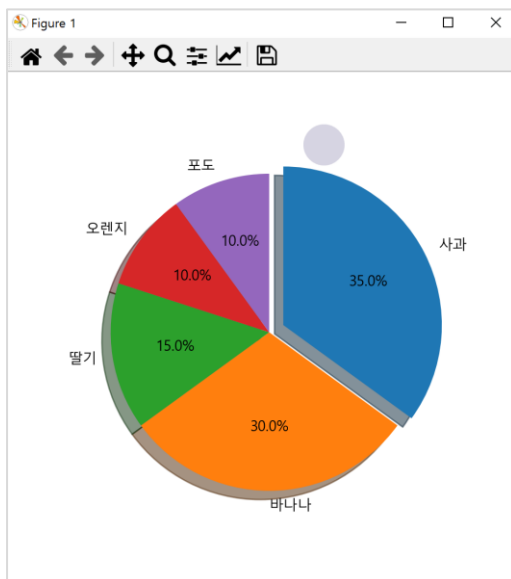
```
01 import matplotlib
02 import matplotlib.pyplot as plt
03 matplotlib.rcParams['font.family'] = 'Malgun Gothic' # '맑은 고딕'으로 설정
04 matplotlib.rcParams['axes.unicode_minus'] = False
05
06 fruit = ['사과', '바나나', '딸기', '오렌지', '포도']
```



```

07 result = [7, 6, 3, 2, 2]
08
09 plt.pie(result)
10 plt.show()
11
12
13 plt.figure(figsize=(5,5))
14 plt.pie(result)
15 plt.show()
16
17
18 plt.figure(figsize=(5,5))
19 plt.pie(result, labels= fruit, autopct='%.1f%%')
20 plt.show()
21
22
23 plt.figure(figsize=(5,5))
24 plt.pie(result, labels= fruit, autopct='%.1f%%', startangle=90, counterclock = False)
25 plt.show()
26
27
28 explode_value = (0.1, 0, 0, 0, 0)
29
30 plt.figure(figsize=(5,5))
31 plt.pie(result, labels= fruit, autopct='%.1f%%', startangle=90, counterclock = False,
32 explode=explode_value, shadow=True)
33 plt.show()

```



15.1.7 그래프 저장하기

- matplotlib를 이용해 생성한 그래프는 화면으로 출력할 수 있을 뿐만 아니라 이미지 파일로 저장할 수도 있다. 그래프를 파일로 저장하려면 다음 방법을 이용한다.
 - `plt.savefig(file_name, [,dpi=dpi_n(기본은 72)])`

[ch15_matplotlib/ex10_savefig.py]

```

01 import matplotlib
02 import matplotlib as mpl

```

```

03 import matplotlib.pyplot as plt
04 import numpy as np
05
06 matplotlib.rcParams['font.family'] = 'Malgun Gothic' # '맑은 고딕'으로 설정
07 matplotlib.rcParams['axes.unicode_minus'] = False
08
09 mpl.rcParams['figure.figsize']
10 mpl.rcParams['figure.dpi']
11
12
13 x = np.arange(0, 5, 1)
14 y1 = x
15 y2 = x + 1
16 y3 = x + 2
17 y4 = x + 3
18
19 plt.plot(x, y1, x, y2, x, y3, x, y4)
20
21 plt.grid(True)
22 plt.xlabel('x')
23 plt.ylabel('y')
24 plt.title('Saving a figure')
25
26 # 그래프를 이미지 파일로 저장. dpi는 100으로 설정
27 plt.savefig('d:\dev\workspace\python\ch15_matplotlib\saveFigTest1.png', dpi=100)
28 plt.show()
29
30
31 fruit = ['사과', '바나나', '딸기', '오렌지', '포도']
32 result = [7, 6, 3, 2, 2]
33 explode_value = (0.1, 0, 0, 0, 0)
34
35 plt.figure(figsize=(5, 5)) # 그래프의 크기를 지정
36 plt.pie(result, labels=fruit, autopct='%1f%%', startangle=90,
37         counterclock=False, explode=explode_value, shadow=True)
38
39 # 그래프를 이미지 파일로 저장. dpi는 200으로 설정
40 plt.savefig('d:\dev\workspace\python\ch15_matplotlib\saveFigTest2.png', dpi=200)
41 plt.show()

```

15.2 pandas로 그래프 그리기

15.2.1 pandas의 그래프 구조

15.2.2 pandas의 선 그래프

- pandas로 생성한 데이터를 plot()을 이용해 그래프로 그린다.

[ch15_matplotlib/ex11_pandas_line.py]

```

01 import pandas as pd
02 import matplotlib.pyplot as plt
03
04 s1 = pd.Series([1,2,3,4,5,6,7,8,9,10])
05 print(s1)
06 '''
07 0      1
08 1      2
09 2      3

```

```

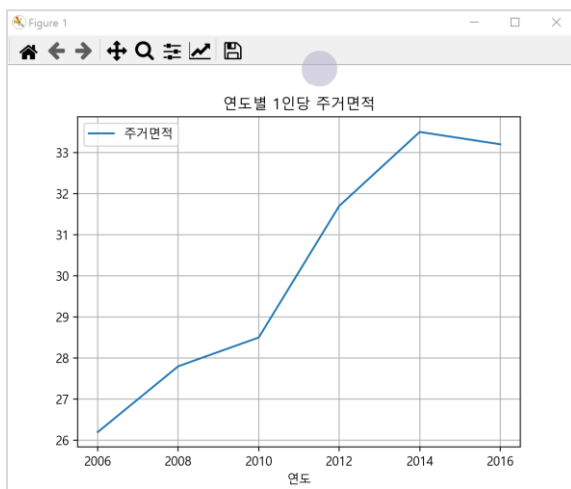
10 3    4
11 4    5
12 5    6
13 6    7
14 7    8
15 8    9
16 9   10
17 dtype: int64
18 '''
19 s1.plot()
20 plt.show()
21
22
23 s2 = pd.Series([1,2,3,4,5,6,7,8,9,10], index = pd.date_range('2019-01-01', periods=10))
24 print(s2)
25 '''
26 2019-01-01    1
27 2019-01-02    2
28 2019-01-03    3
29 2019-01-04    4
30 2019-01-05    5
31 2019-01-06    6
32 2019-01-07    7
33 2019-01-08    8
34 2019-01-09    9
35 2019-01-10   10
36 Freq: D, dtype: int64
37 '''
38 s2.plot()
39 plt.show()
40
41
42 s2.plot(grid=True)
43 plt.show()
44
45
46 df_rain = pd.read_csv('D:\dev\workspace\python\ch14_numpy\sea_rain1.csv', index_col="연도" )
47 print(df_rain)
48 '''
49          동해      남해      서해      전체
50 연도
51 1996  17.4629  17.2288  14.4360  15.9067
52 1997  17.4116  17.4092  14.8248  16.1526
53 1998  17.5944  18.0110  15.2512  16.6044
54 1999  18.1495  18.3175  14.8979  16.6284
55 2000  17.9288  18.1766  15.0504  16.6178
56 '''
57
58
59 import matplotlib
60 matplotlib.rcParams['font.family'] = 'Malgun Gothic'# '맑은 고딕'으로 설정
61 matplotlib.rcParams['axes.unicode_minus'] = False
62
63 df_rain.plot()
64 plt.show()
65
66
67 rain_plot = df_rain.plot(grid = True, style = ['r--*', 'g-o', 'b:*', 'm-.p'])
68 rain_plot.set_xlabel("연도")
69 rain_plot.set_ylabel("강수량")
70 rain_plot.set_title("연간 강수량")
71 plt.show()
72
73
74 year = [2006, 2008, 2010, 2012, 2014, 2016] # 연도

```

```

75 area = [26.2, 27.8, 28.5, 31.7, 33.5, 33.2] # 1인당 주거면적
76 table = {'연도':year, '주거면적':area}
77 df_area = pd.DataFrame(table, columns=['연도', '주거면적'])
78 print(df_area)
79 '''
80     연도  주거면적
81 0  2006   26.2
82 1  2008   27.8
83 2  2010   28.5
84 3  2012   31.7
85 4  2014   33.5
86 5  2016   33.2
87 '''
88
89 df_area.plot(x='연도', y='주거면적', grid = True, title = '연도별 1인당 주거면적')
90 plt.show()

```



15.2.3 pandas의 산점도

- 산점도는 두 데이터의 관계를 추정할 때 사용할 수 있다.

[ch15_matplotlib/ex12_pandas_scatter.py]

```

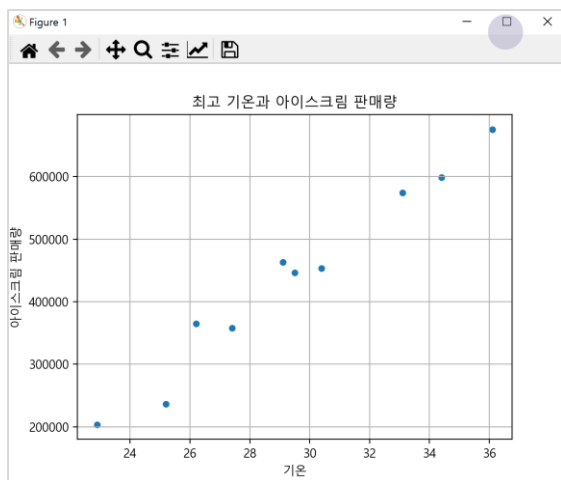
01 import matplotlib
02 import matplotlib.pyplot as plt
03 import pandas as pd
04
05 matplotlib.rcParams['font.family'] = 'Malgun Gothic' # '맑은 고딕'으로 설정
06 matplotlib.rcParams['axes.unicode_minus'] = False
07
08 temperature = [25.2, 27.4, 22.9, 26.2, 29.5, 33.1, 30.4, 36.1, 34.4, 29.1]
09 Ice_cream_sales = [236500, 357500, 203500, 365200,
10                   446600, 574200, 453200, 675400, 598400, 463100]
11
12 dict_data = {'기온': temperature, '아이스크림 판매량': Ice_cream_sales}
13 df_ice_cream = pd.DataFrame(dict_data, columns=['기온', '아이스크림 판매량'])
14
15 print(df_ice_cream)
16 '''
17     기온  아이스크림 판매량
18 0  25.2    236500
19 1  27.4    357500

```

```

20 2 22.9 203500
21 3 26.2 365200
22 4 29.5 446600
23 5 33.1 574200
24 6 30.4 453200
25 7 36.1 675400
26 8 34.4 598400
27 9 29.1 463100
28 '''
29
30 df_ice_cream.plot.scatter(x='기온', y='아이스크림 판매량',
31                           grid=True, title='최고 기온과 아이스크림 판매량')
32 plt.show()

```



15.2.4 pandas의 막대 그래프

■ 이번에는 막대 그래프에 대해 살펴본다.

[ch15_matplotlib/ex13_pandas_bar.py]

```

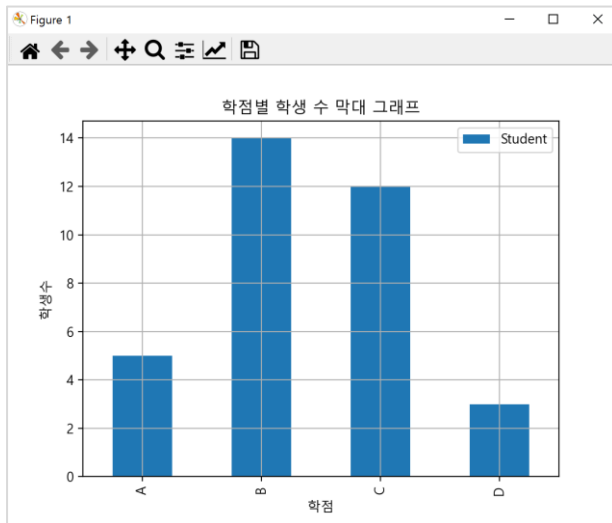
01 import matplotlib
02 import matplotlib.pyplot as plt
03 import pandas as pd
04
05 matplotlib.rcParams['font.family'] = 'Malgun Gothic' # '맑은 고딕'으로 설정
06 matplotlib.rcParams['axes.unicode_minus'] = False
07
08 grade_num = [5, 14, 12, 3]
09 students = ['A', 'B', 'C', 'D']
10
11 df_grade = pd.DataFrame(grade_num, index=students, columns=['Student'])
12 print(df_grade)
13 '''
14      Student
15 A          5
16 B         14
17 C         12
18 D          3
19 '''
20
21 grade_bar = df_grade.plot.bar(grid=True)
22 grade_bar.set_xlabel("학점")

```

```

23 grade_bar.set_ylabel("학생수")
24 grade_bar.set_title("학점별 학생 수 막대 그래프")
25 plt.show()

```



15.2.5 pandas의 히스토그램

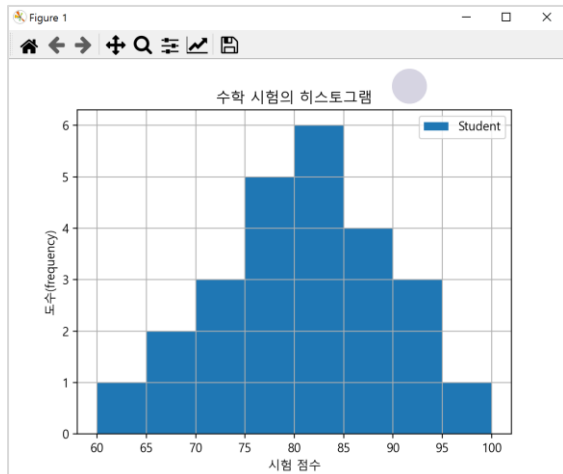
- pandas에서 히스토그램을 그리려면 `plot(kind='hist',[, bin=num])` 혹은 `plot.hist([bin=num])`를 이용한다.

[ch15_matplotlib/ex14_pandas_hist.py]

```

01 import matplotlib
02 import matplotlib.pyplot as plt
03 import pandas as pd
04
05 matplotlib.rcParams['font.family'] = 'Malgun Gothic' # '맑은 고딕'으로 설정
06 matplotlib.rcParams['axes.unicode_minus'] = False
07
08 math = [76,82,84,83,90,86,85,92,72,71,100,87,81,76,94,78,81,60,79,69,74,87,82,68,79]
09
10 df_math = pd.DataFrame(math, columns = ['Student'])
11
12 math_hist = df_math.plot.hist(bins=8, grid = True)
13 math_hist.set_xlabel("시험 점수")
14 math_hist.set_ylabel("도수(frequency)")
15 math_hist.set_title("수학 시험의 히스토그램")
16
17 plt.show()

```



15.2.6 pandas의 파이 그래프

- pandas에서 파이 그래프를 그리려면 `plot(kind='pie')` 혹은 `plot.pie()`를 이용한다.

[ch15_matplotlib/ex15_pandas_pie.py]

```
01 import matplotlib
02 import matplotlib.pyplot as plt
03 import pandas as pd
04
05 matplotlib.rcParams['font.family'] = 'Malgun Gothic' # '맑은 고딕'으로 설정
06 matplotlib.rcParams['axes.unicode_minus'] = False
07
08 fruit = ['사과', '바나나', '딸기', '오렌지', '포도']
09 result = [7, 6, 3, 2, 2]
10
11 df_fruit = pd.Series(result, index=fruit, name='선택한 학생수')
12 print(df_fruit)
13 '''
14 사과      7
15 바나나    6
16 딸기      3
17 오렌지    2
18 포도      2
19 Name: 선택한 학생수, dtype: int64
20 '''
21
22 df_fruit.plot.pie()
23 plt.show()
24
25
26 explode_value = (0.1, 0, 0, 0, 0)
27 fruit_pie = df_fruit.plot.pie(figsize=(5, 5), autopct='%1.1f%%', startangle=90,
28                               counterclock=False, explode=explode_value, shadow=True, table=True)
29 fruit_pie.set_ylabel("") # 불필요한 y축 라벨 제거
30 fruit_pie.set_title("과일 선호도 조사 결과")
31
32 # 그래프를 이미지 파일로 저장. dpi는 200으로 설정
33 plt.savefig('d:\dev\workspace\python\ch15_matplotlib\saveFigTest3.png', dpi=200)
34 plt.show()
```

