06 그래프 함수로 시각화 하기

목차

◆ 파이썬 함수로 기본적인 그래프 작성 방법 학습

matplotlib 그래프 종류 및 관련 옵션

● 그래프 종류

구분	주 요 내 용
막대 그래프	시간의 흐름 및 항목별 빈도 표현
선 그래프	두 데이터의 관계 표현
원 그래프	데이터의 분포 시각화
히스토그램	데이터 구간별 분포 파악
상자 수염 그래프	항목별 분포를 비교하며 이상치 파악

● 관련 패키지

패키지	특 징
matplotlib	다양한 형태의 **그래프 함수 및 서식 편집 기능** 제공
pandas	**데이터 프레임 내 데이터**를 요약, 전처리 후 그래프를 그릴 수 있는 함수 제공
seaborn	**색상 테마와 통계** 등을 추가하여 그래프를 그릴 수 있는 기능 제공

matplotlib 모듈 및 그래프 함수

● 그래프 함수

구분	주 요 내 용
plot	선 그래프
scatter	산점도
bar	수직 막대 그래프
barh	수평 막대 그래프
pie	원 그래프
hist	히스토그램
boxplot	상자 수염 그래프

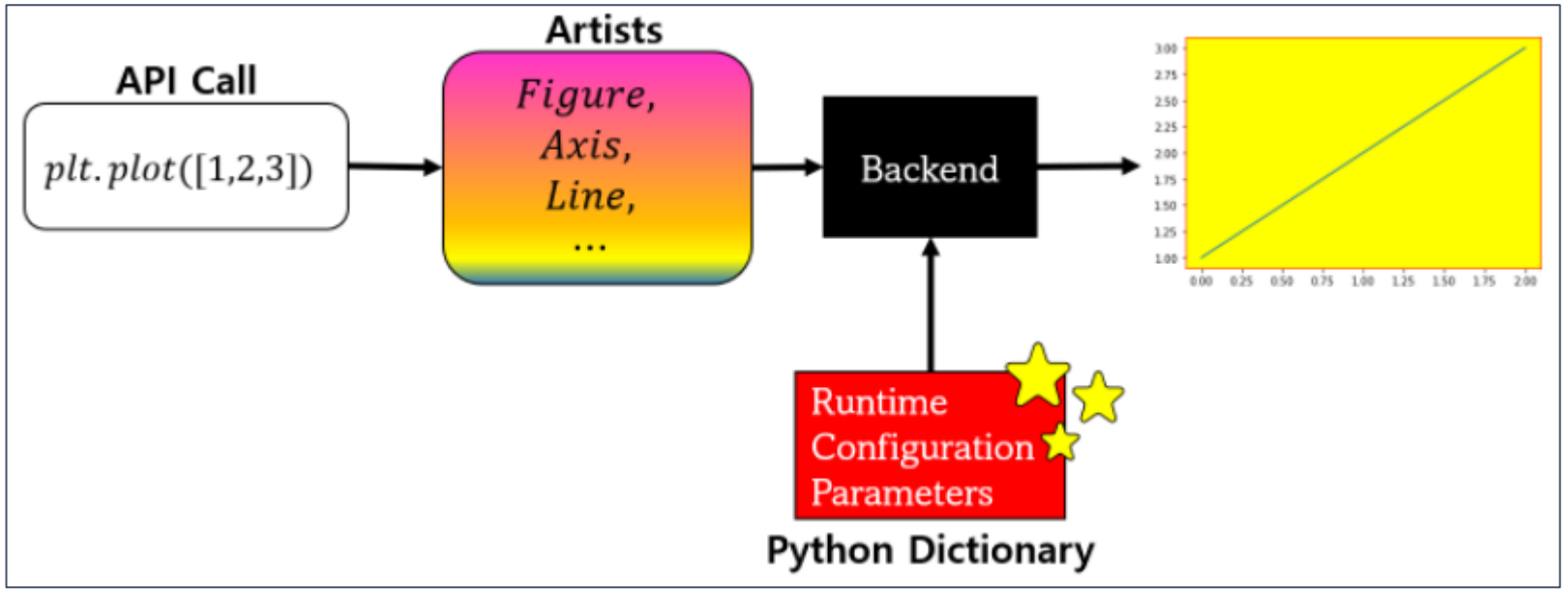
matplotlib 모듈 및 그래프 함수

● 패키지 설치하고 모듈 불러오기

```
!pip install matplotlib                          # matplotlib 패키지 설치하기
import matplotlib.pyplot as plt                  # matplotlib의 하위에 있는 pyplot를 plt라는 이름으로 불러오기
plt.rcParams[“font.family”] = "Malgun Gothic"    # 그래프의 제목이나 범례에 한글을 사용하기 위함
```

rcParams : Runtime Configuration Parameters

그래프의 폰트 사이즈, 크기, 바탕색 등 여러가지 성질들이 모두 **matplotlib.rcParams** 딕셔너리에 담겨있다.



matplotlib 에서 한글 깨짐 처리

◆ matplotlib 패키지 한글 깨짐 처리

```
import pandas as pd
import matplotlib.pyplot as plt
import platform

if platform.system() == 'Darwin':                    #맥
        plt.rc('font', family='AppleGothic')
elif platform.system() == 'Windows':                 #윈도우
        #plt.rcParams["font.family"] = "Malgun Gothic"
        plt.rc('font', family='Malgun Gothic')
elif platform.system() == 'Linux':                   #리눅스 (구글 코랩)
        #!wget "https://www.wfonts.com/download/data/2016/06/13/malgun-gothic/malgun.ttf"
        #!mv malgun.ttf /usr/share/fonts/truetype/
        #import matplotlib.font_manager as fm
        #fm._rebuild()
        plt.rc('font', family='Malgun Gothic')

plt.rcParams['axes.unicode_minus'] = False           # 그래프에서 마이너스 기호가 깨질 때
```

matplotlib 모듈 및 그래프 함수

- **rcParams 딕셔너리 내용보기**

```
print(plt.rcParams)
```

```
_internal.classic_mode: False
agg.path.chunksize: 0
animation.bitrate: -1
animation.codec: h264
animation.convert_args: []
animation.convert_path: convert
animation.embed_limit: 20.0
animation.ffmpeg_args: []
animation.ffmpeg_path: ffmpeg
animation.frame_format: png
animation.html: none
animation.writer: ffmpeg
axes.autolimit_mode: data
axes.axisbelow: line
axes.edgecolor: black
axes.facecolor: white
axes.formatter.limits: [-5, 6]
axes.formatter.min_exponent: 0
```

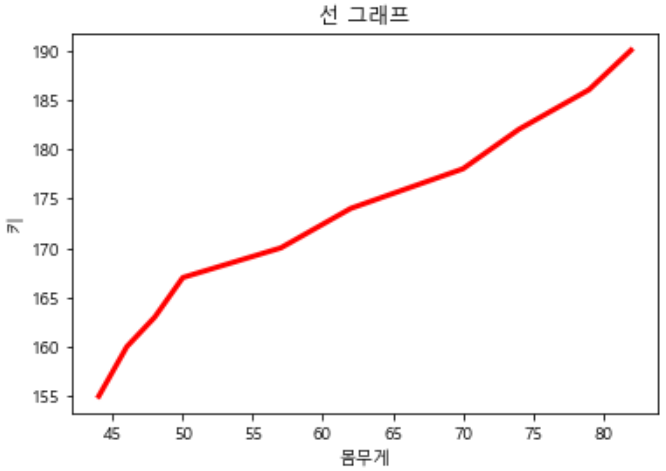
matplotlib 그래프 옵션

구분	정의	예시
xlim, ylim	x축, y축 **범위**	plt.xlim(-1, 1) # x축 범위를 -1에서 1까지 지정
grid	**격자** 눈금	plt.grid(True) # 격자 생성
legend	**범례** 위치 지정 1: 우측 위, 2: 좌측 위 3: 좌측 아래, 4: 우측 아래 6: 좌측 중앙, 7: 우측 중앙 8: 하측 중앙, 9: 상측 중앙	plt.legend(2) # 좌측 상단에 범례 위치
xlabel, ylabel	x축, y축 **레이블**	plt.xlabel("시간") # x축 제목을 시간으로 설정
title	그래프 **제목**	plt.title("월간매출") # 그래프 제목을 월간매출로 설정
xticks, yticks	x축, y축 **눈금** 조정	-

선 그래프

➢ x, y 축 두 변수의 관계를 선으로 연결하여 표현
➢ plt.plot(x, y, color, lw) # x = x축 데이터, y = y축 데이터, color = 컬러, lw = 라인 두께

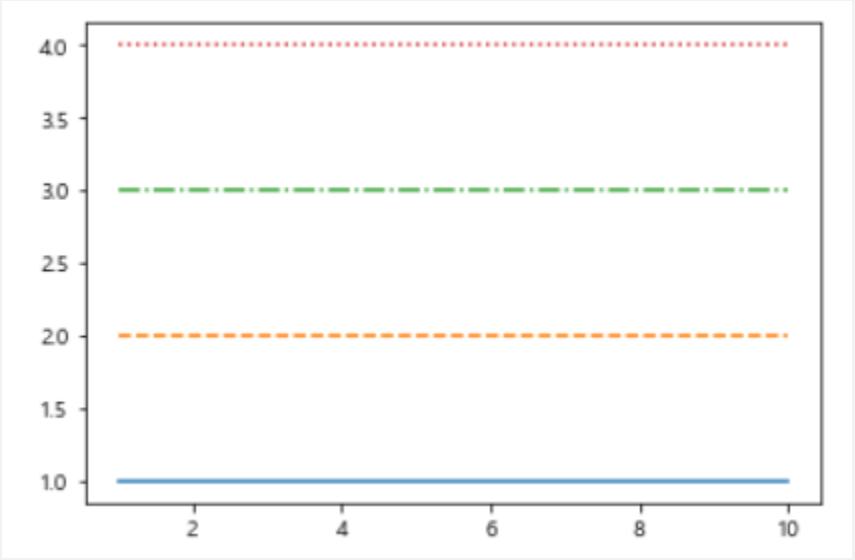
```
# 선 그래프 그리기
height = [155, 160, 163, 167, 170, 174, 178, 182, 186, 190]     # 키 데이터
weight = [44, 46, 48, 50, 57, 62, 70, 74, 79, 82]               # 몸무게 데이터
plt.title("선 그래프")                                           # 제목
plt.xlabel("몸무게")                                             # x축 제목
plt.ylabel("키")                                                 # y축 제목
plt.plot(weight, height, color = "red", lw = 3)                  # 그래프 그리기, 컬러 적색, 선 두께 3
plt.show()                                                       # 그래프 출력
```



선 종류 지정하기

➢ **plot() 함수 속성인 linestyle 을 지정하면 선 종류를 변경할 수 있다.**

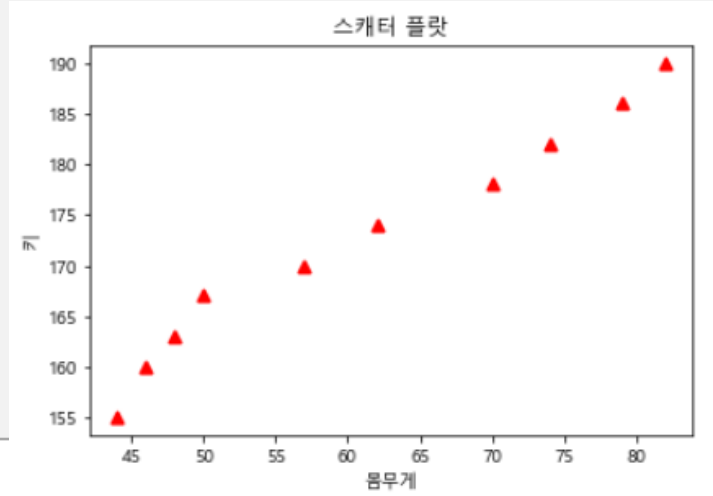
```
plt.plot([1,10], [1,1], linestyle = "solid")        # 실선(line)
plt.plot([1,10], [2,2], linestyle = "dashed")       # 파선(dash)
plt.plot([1,10], [3,3], linestyle = "dashdot")      # 쇄선(dashdot)
plt.plot([1,10], [4,4], linestyle = "dotted")       # 점선(dot)                    #
```



산점도 그래프

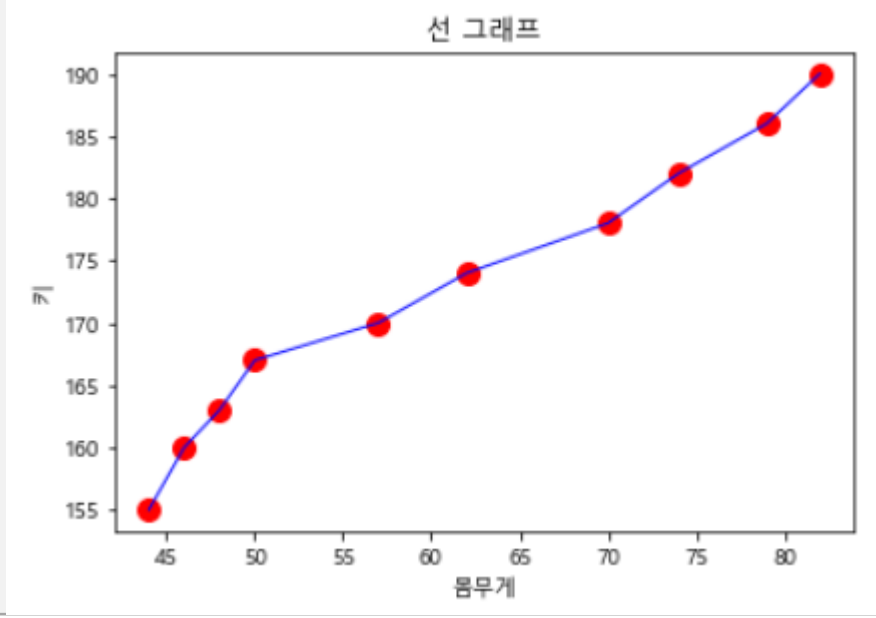
- 모든 데이터를 포인트로 표시해 관련성 여부를 시각적으로 판단할 수 있으므로 두 변수의 관계를 알아볼 때 유용
- plt.scatter(x, y, marker, color, lw) # marker = 사각형(s)/삼각형(^)/엑스(x)/원(o)/별표(*)

```
# 산점도 그리기
height = [155, 160, 163, 167, 170, 174, 178, 182, 186, 190]   # 키 데이터
weight = [44, 46, 48, 50, 57, 62, 70, 74, 79, 82]             # 몸무게 데이터
plt.title("스캐터 플랏")                                        # 제목
plt.xlabel("몸무게")                                            # x축 제목
plt.ylabel("키")                                                # y축 제목
plt.scatter(weight, height, marker="^", color="red", lw=2)     # 스캐터 플랏 그리기
plt.show()                                                     # 그래프 출력
```



산점도와 선 그래프 함께 그리기

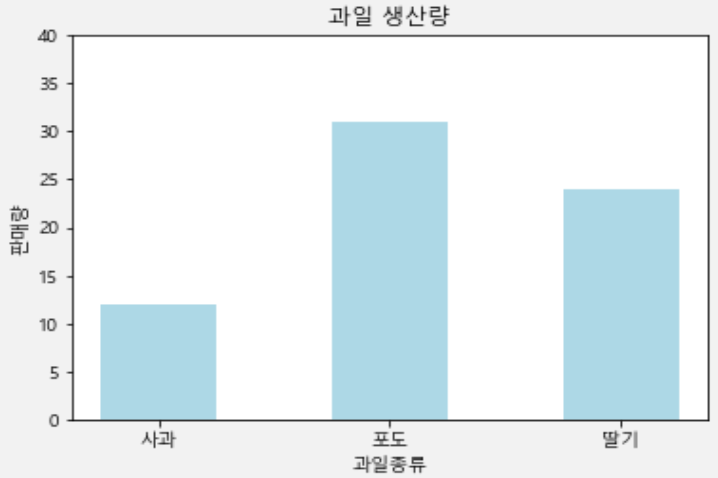
```
# 산점도와 선 그래프 함께 그리기
height = [155, 160, 163, 167, 170, 174, 178, 182, 186, 190]     # 키 데이터
weight= [44, 46, 48, 50, 57, 62, 70, 74, 79, 82]                # 몸무게 데이터
plt.title("선 그래프")                                           # 제목
plt.xlabel("몸무게")                                             # x축 제목
plt.ylabel("키")                                                 # y축 제목
plt.plot(weight, height, color = "blue", lw = 1)                 # 선 그래프, 청색, 선 두께 1
plt.scatter(weight, height, color = "red", s = 100)              # 산점도, 적색, 크기 100
plt.show()
```



막대 그래프

- 여러 범주의 데이터 빈도를 표현하고 비교
- plt.bar(x, y, width, color) # x = 범주, y = 그래프 높이, width = 그래프 폭, color = 색상

```
# 막대 그래프 그리기
x = [“사과”, “포도”, “딸기”]
y = [12, 31, 24]
plt.title("과일 생산량")
plt.bar(x, y, color = "lightblue", width = 0.5)
plt.xlabel("과일 종류")
plt.ylabel("판매량")
plt.ylim(0, 40)
plt.show()
```



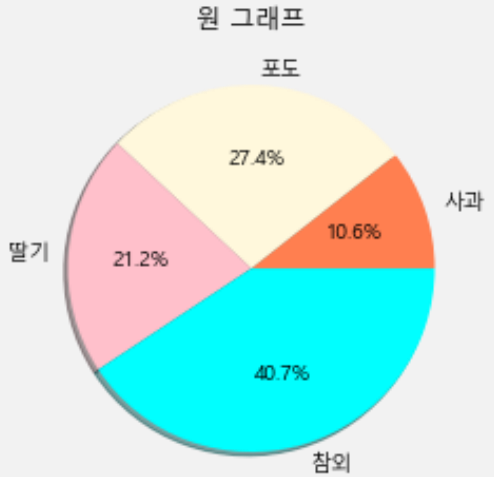

```
# 항목 데이터
# 빈도(크기) 데이터
# 그래프 제목
# 색상은 밝은 파랑, 그래프 폭은 0.5
# x축 제목
# y축 제목
# y축 범위 지정
# 그래프 출력
```

원 그래프

- 부분과 전체, 부분과 부분 간의 비율을 시각화
- plt.pie(y, labels, autopct, colors, shadow)

  # y=데이터, labels=항목, autopct=파이 조각 백분율, colors=파이 조각의 컬러, shadow=그림자 효과

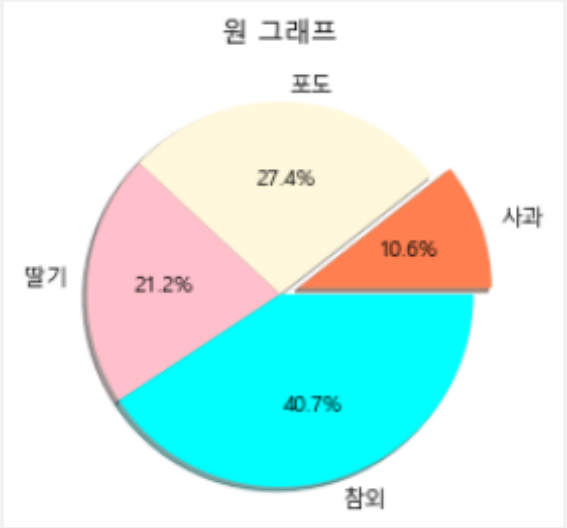
```
# 원 그래프 그리기
x = [“사과”, “포도”, “딸기”, “참외”]                                  # 항목 데이터
y = [12, 31, 24, 46]                                                  # 비율 데이터
colors = [“coral”, “cornsilk”, “pink”, “aqua”]                        # 항목별 컬러 지정
plt.title("원 그래프")                                                 # 파이 차트 제목
plt.pie(y, labels = x, autopct = “%1.1f%%”, colors = colors, shadow = True)   # 그래프 설정
plt.show()                                                            # 그래프 출력
```



원 그래프

- 원하는 조각만 분리하여 출력하기
- 속성의 explode = (0.1, 0, 0, 0) # 첫번째 값인 "사과" 조각이 0.1 크기만큼 분리

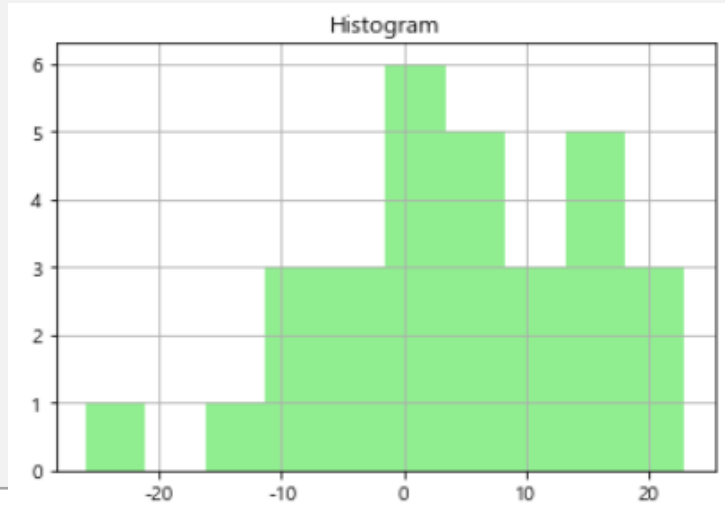
```
# 원 그래프 조각 분리하기
x = ["사과", "포도", "딸기", "참외"]                    # 항목 데이터
y = [12, 31, 24, 46]                                  # 비율 데이터
colors = ["coral", "cornsilk", "pink", "aqua"]        # 항목별 컬러 지정
plt.title("원 그래프")                                 # 파이 차트 제목
plt.pie(y, labels = x, autopct = "%1.1f%%", colors = colors, explode = (0.1, 0, 0, 0), shadow = True)
plt.show()                                            # 그래프 출력
```



히스토그램

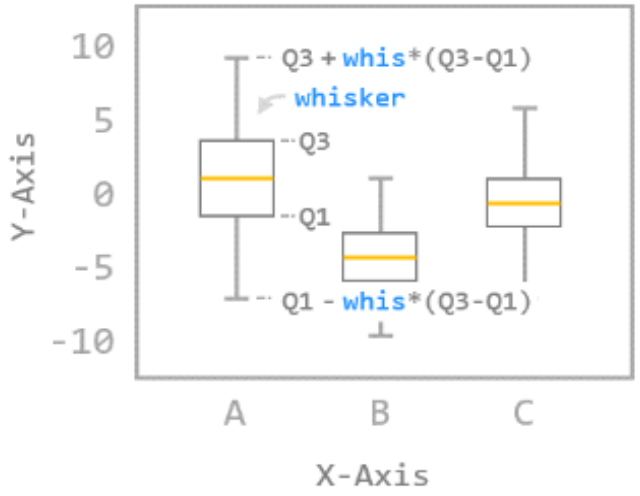
- 데이터의 관계를 표현하거나 데이터의 분포를 파악할 때. 계급을 가로축에, 계급별 빈도수를 세로축에 직사각형으로 표현
- plt.hist(x, bins, color) # x = 데이터, bins = 계급구간 수, color = 히스토그램 컬러

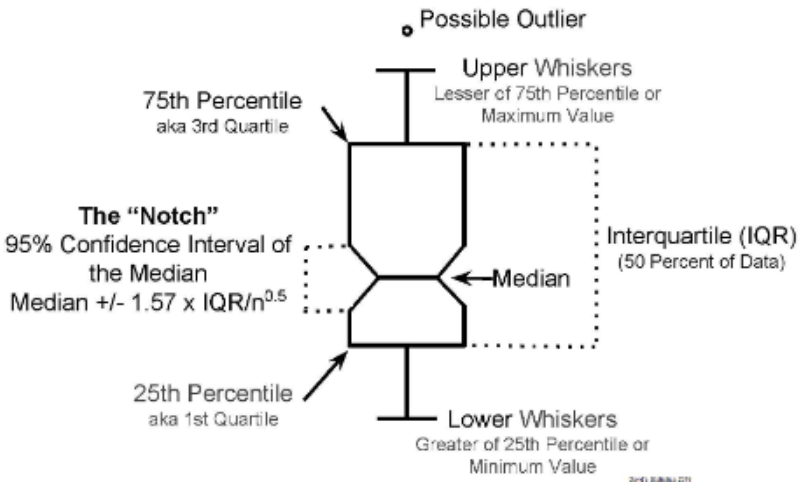
```
# 히스토그램 그리기
plt.rcParams['axes.unicode_minus'] = False          # 그래프에서 마이너스 기호가 깨질 때

x = [18, 4, 10, 22, 19, -10, 10, -2, -1, 4, 1, 15, 8, 1, 4, 3, 15, -2, 3, - 9, -26, 7, 9, -7, 23, -15, 0, -2, 15, 15]
plt.title("Histogram")                              # 그래프 제목 지정
plt.grid(True)                                      # 격자 출력
plt.hist(x, bins = 10, color = "lightgreen")        # 히스토그램 설정, 계급 구간을 10으로 지정
plt.show()                                          # 그래프 출력
```



상자 수염 그래프

- **Box plot 또는 Box-Whisker plot**
- **연속형 변수에 대해 분포 형태, 퍼짐 정도, 이상치 여부 등을 시각화.**
- **하나 혹은 여러 개의 그룹을 비교하는데 유용.**
- **연속형 변수에 대해 최소값(min), 제1사분위수(Q1), 중앙값(Q2, median), 제3사분위수(Q3), 최대값(max)의 요약 통계량을 활용하여 시각화**
- **박스의 위쪽 경계로부터 Q3 + whis*(Q3-Q1)까지, 박스의 아래쪽 경계로부터 Q1 - whis*(Q3-Q1)까지 수염 (Whisker)을 나타낸다.**
- **notch = True로 지정하면 중앙값 (Median)의 95% 신뢰 구간을 노치 형태로 표시한 Notched Boxplot을 나타낸다.**





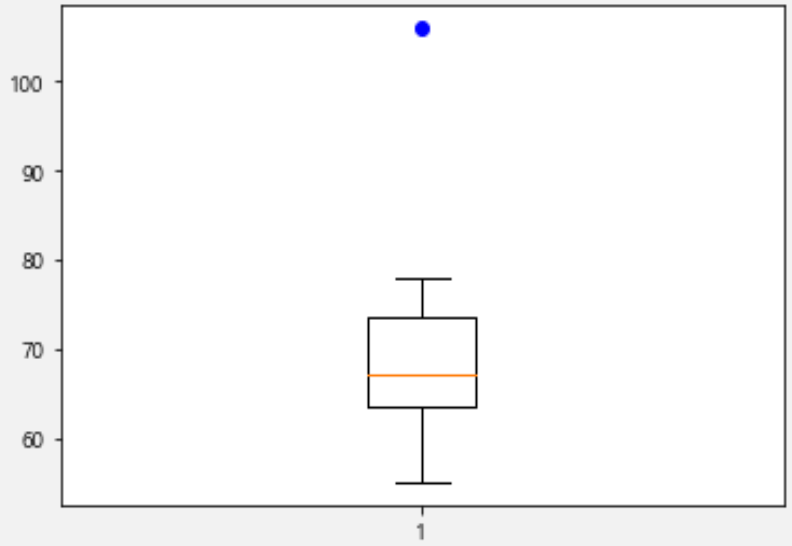
상자 수염 그래프

➢ plt.boxplot(x, sym, vert) # x=데이터, sym=이상치 표시 기호, vert=수직(1)/수평(0) 박스 플랏

- 이상치 : 상,하단 꼬리를 벗어나는 값
- sym="bo" # symbol=blue oval
- vert=True(수직) / False(수평)

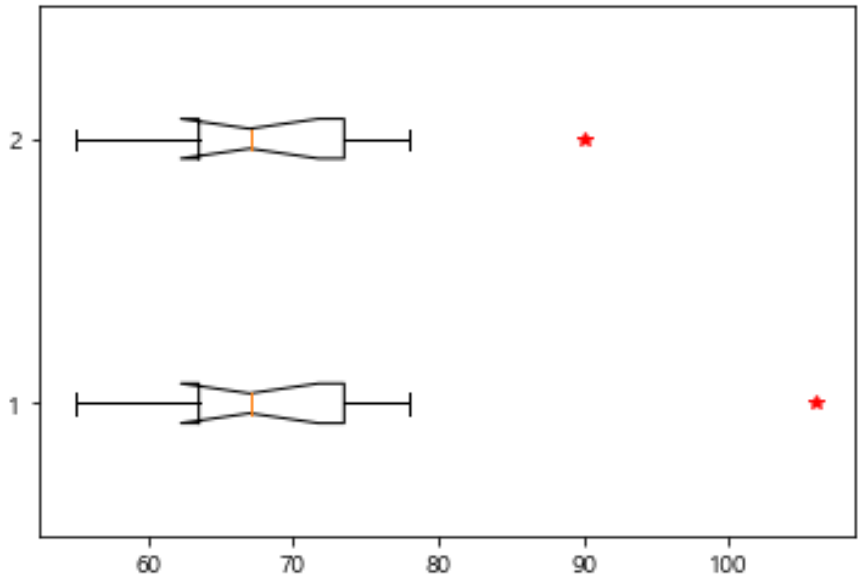
```
# 수직  플롯 그리기
x = [55, 60, 63, 67, 70, 74, 78, 66, 64, 73, 106]
plt.boxplot(x, sym = "bo", vert = 1)
plt.show()
```

그래프 출력



상자 수염 그래프

```
# 수평 박스 플롯 그리기
x = [55, 60, 63, 67, 70, 74, 78, 66, 64, 73, 106]
y = [55, 60, 63, 67, 70, 74, 78, 66, 64, 73, 90]
#plt.boxplot([x, y])
plt.boxplot([x, y], sym="r*", vert=0, notch=True)          # symbol=red star, 수평, notch
plt.show()
```



pandas 그래프 지정방법 및 종류

● 그래프 지정방법

➢ pandas는 'pd'로 줄여서 사용

```
pd.plot.그래프함수(x , y, ...)          # plot() 함수와 그래프 종류 추가
pd.plot(x, y, kind = "그래프종류", ...)  # plot() 함수 내 kind 옵션으로 그래프 종류 지정
```

● 그래프 종류

구분	주 요 내 용
bar	**수직 막대 그래프**
barh	수평 막대 그래프
scatter	산점도
pie	원 그래프
hist	히스토그램
boxplot	상자 수염 그래프

패키지 및 파일 불러오기

● 예제 파일: chapter06/그래프 실습2.xlsx

```
import pandas as pd                                    # pandas 불러오기
import matplotlib.pyplot as plt                        # matplotlib 불러오기
plt.rcParams['“font.family'”] = "Malgun Gothic"        # 그래프에서 한글 폰트 깨짐 방지

graph = pd.read_excel(r“chapter06/그래프 실습2.xlsx", sheet_name = "Sheet1")

graph.head(10)                                         # 총 30행으로 구성, 상위 10행만 출력
```

	반	성명	국어	영어	수학	사회	과학
0	1반	홍길동	82	80	94	23	64
1	2반	백일홍	76	63	76	84	56
2	3반	이삼상	75	74	86	90	61
3	1반	정말로	83	55	64	90	65
4	2반	한번도	82	95	66	75	60
5	3반	이철수	53	65	59	88	69
6	1반	김영자	66	71	32	84	57
7	2반	다니엘	81	54	95	71	48
8	3반	이미로	59	69	75	90	52
9	1반	신성삼	64	66	59	91	49

선 그래프

➢ **pd.plot(y, grid, title, color, ...) # y = y축 데이터, grid = 격자, title = 제목, color = 컬러**

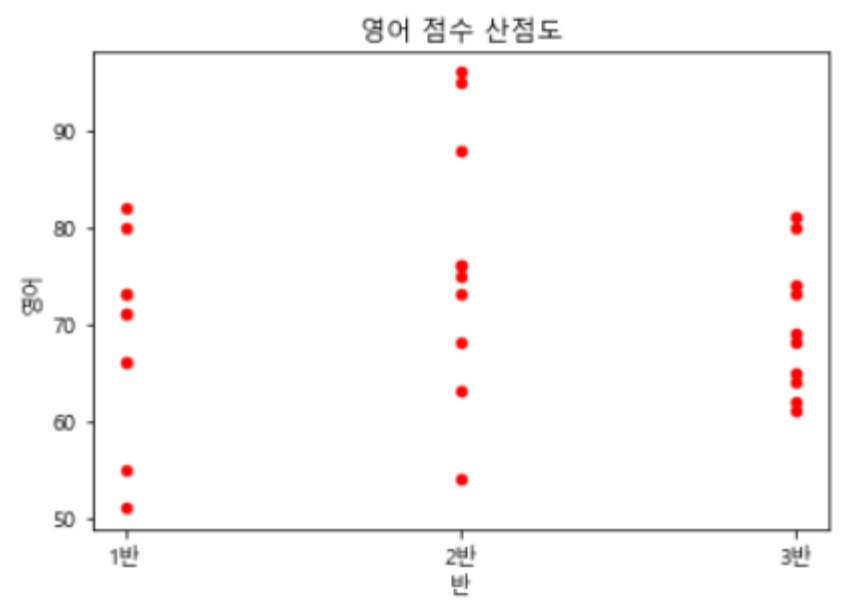
```
# 국어, 영어 열을 선택하여 선 그래프 그리기
graph.plot(y = ["국어", "영어"], grid = True, title = "선그래프", color = ["green","red"])
plt.show()
```

산점도 그래프

➢ pd.scatter(x, y, marker, grid, title, color, ...)

x축 데이터, y축 데이터, 표시형식, 격자, 제목, 컬러, ...

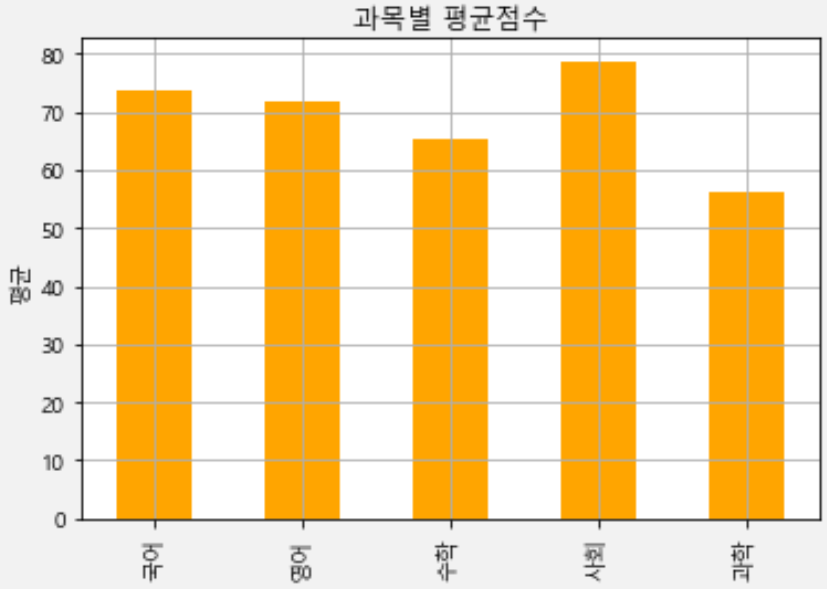
```
# 반별 영어 점수 산점도 그리기
graph.plot.scatter(x = "반", y = "영어", color = "red", title = "영어 점수 산점도")
plt.show()
```



막대 그래프 (수직 막대)

- 예제 파일의 과목별 모든 점수를 막대 그래프로 표현하면 5과목 30명이니 총 150개의 막대그래프.
  이 때는 **요약 정보**로 표현하는 것이 더 적합하다.
- pd.plot.bar(grid, title, color, ...)

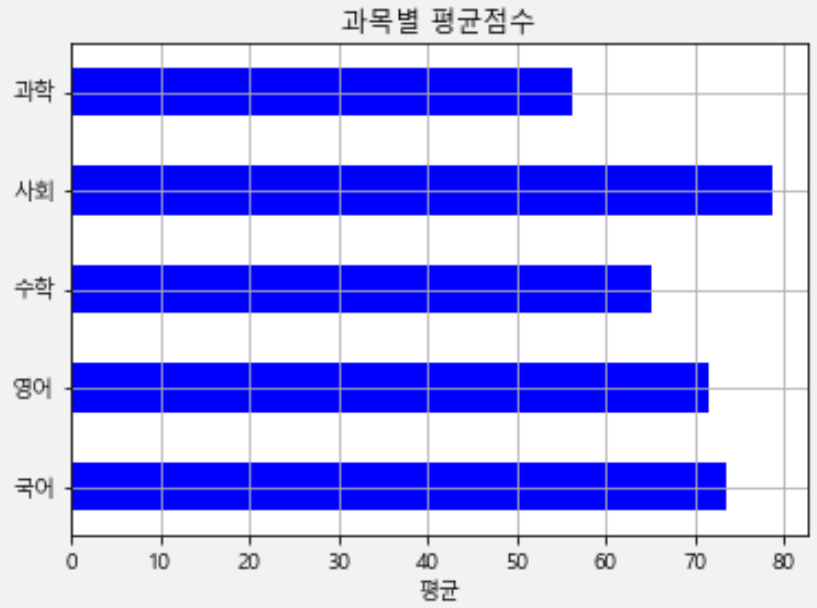
```
# 과목별 점수 데이터의 평균값을 막대 그래프로 그리기
# 2~6열(국어~과학) 평균을 구한 후 막대그래프 그리기
graph.iloc[ : , 2:7].mean().plot.bar(grid = True, title = "과목별 평균점수", color = "orange", ylabel = "평균")
plt.show()
```



막대 그래프 (수평 막대)

➢ **pd.plot.barh(grid, title, color, ...)**

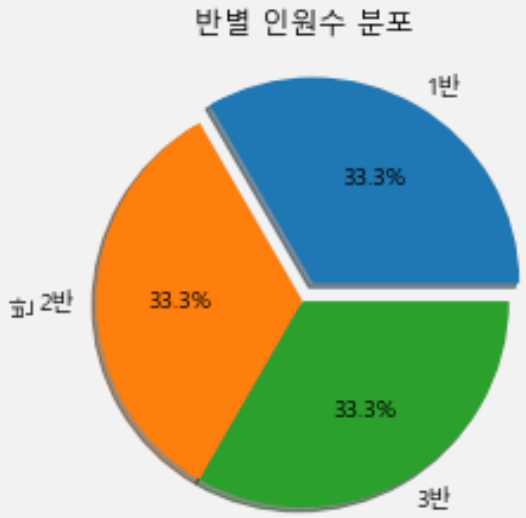
```
# 과목별 점수 데이터의 평균값을 수평 막대 그래프로 그리기
a = graph.iloc[ : , 2:7].mean().plot.barh(grid = True, color = "blue")
a.set_xlabel("평균")
a.set_title("과목별 평균점수")
plt.show()
```



원 그래프

➢ **pd.plot.pie(data, index)**
 # data = 시리즈 데이터명, index = 항목, ylabels = 항목명, autopct = 파이조각 백분율 표시,
explode = 파이조각 위치, shadow =그림자 여부

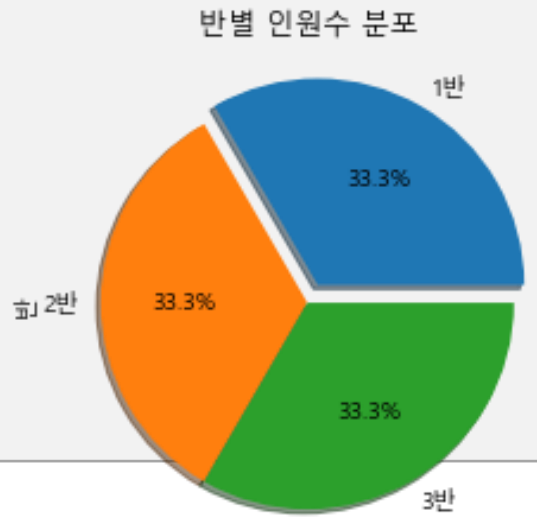
```
# 반별 인원수로 원 그래프 그리기
class_c = graph.groupby("반").size()                    # 반별 인원수 카운트해서 class_c에 저장
class_c.plot.pie(title = "반별 인원수 분포", ylabel = "반", autopct = “%1.1f%%”,
                 explode = (0.1, 0, 0), shadow = True)
plt.show()
```



원 그래프

- pandas에는 행과 열로 구성된 2차원 데이터 구조인 DataFrame과, 하나의 열로 구성된 1차원 데이터구조인 Series가 있다.
- pd.Series(data, index)

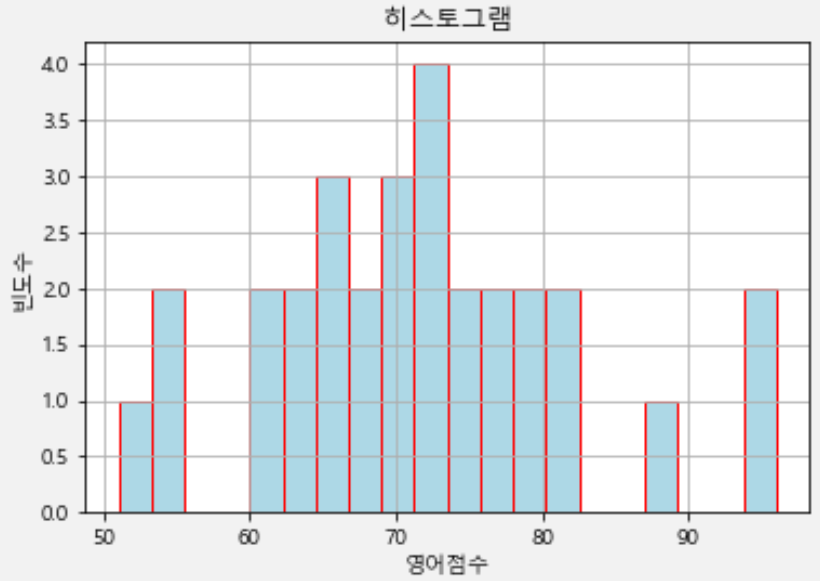
```
# Series를 이용해 과일 판매량으로 원 그래프 그리기
fruit = ["사과", "포도", "딸기", "참외"]                    # 과일 종류 데이터
sales = [12, 31, 24, 46]                                    # 판매량 데이터
df = pd.Series(sales, index = fruit)                        # pd.Series 자료형 생성

df.plot.pie(title="과일 판매량", ylabel = "과일", autopct = "%1.1f%%", explode = (0.1, 0, 0, 0),
            shadow = True)
plt.show()
```



히스토그램 (단일)

➢ pd[“열 이름”].plot.hist(bins, color, edgecolor, title,...) # bins=구간수, edgecolor=그래프외곽선컬러

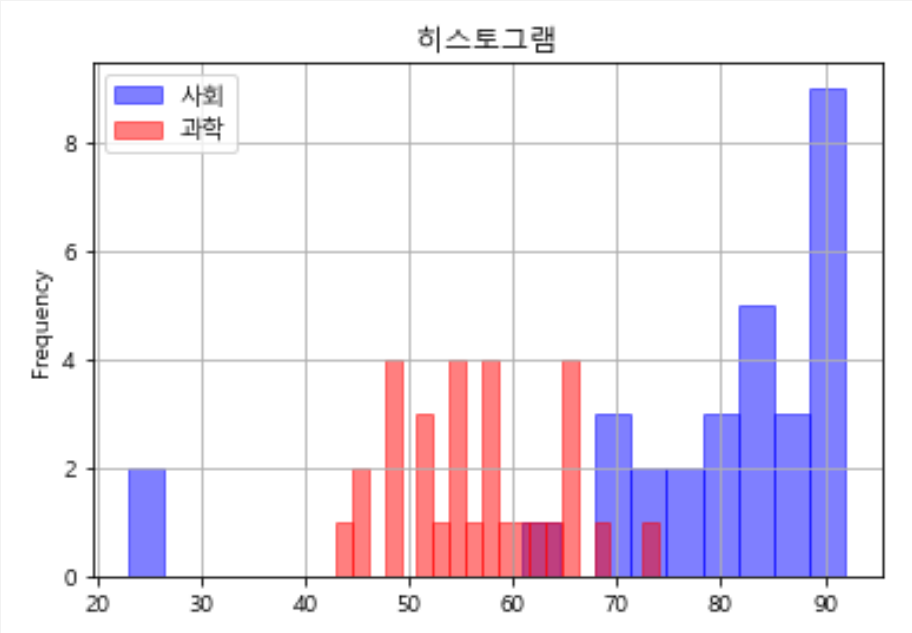
```
# 영어점수 히스토그램
a = graph["영어"].plot.hist( bins = 20, color = "lightblue", edgecolor = "red",
                             grid = True, title = "히스토그램")
a.set_xlabel("영어점수"), a.set_ylabel("빈도수")
plt.show()
```



히스토그램 (중첩)

➢ 중첩부분은 alpha값으로 투명도를 조절하여 같이 볼 수 있도록 표시

```
# 사회, 과학 점수 분포 시각화
graph["사회"].plot.hist( bins = 20, color = "blue", edgecolor = "blue",
                    alpha = 0.5, title = "히스토그램")
graph["과학"].plot.hist( bins = 20, color = "red", edgecolor = "red",
                    alpha = 0.5, grid = True)
plt.legend()
plt.show()
```



상자 수염 그래프

- 데이터 프레임의 수치형 열 값을 범주형 열 값으로 구분하여 그린다.
- 즉, 범주형 데이터 분포를 비교하며 이상치를 파악할 수 있다.
- pd.boxplot(column = [], by) # column = 수치형 열, by = 범주형 열

```
# 상자 수염 그래프
graph.boxplot(column = ["국어"], by = "반")
plt.show()
```