

<7장> 시각화 라이브러리

학습 목표

- 데이터 시각화 필요성을 이해하고 사용한다.
- 데이터 시각화에 효율적인 맷플롯립(Matplotlib) 라이브러리를 이해하고 사용한다.
- 맷플롯립(Matplotlib) 라이브러리로 표현한 그래프에 시본(Seaborn) 라이브러리를 활용하여 손쉽게 입체적이고 실용적인 그래프로 표현한다.

목차

01 데이터 시각화 이해

02 맷플롯립 라이브러리

03 시본 라이브러리

01

데이터 시각화 이해

1. 데이터 시각화 이해

■ 데이터 시각화 장점

- 많은 양의 데이터를 시각적으로 요약

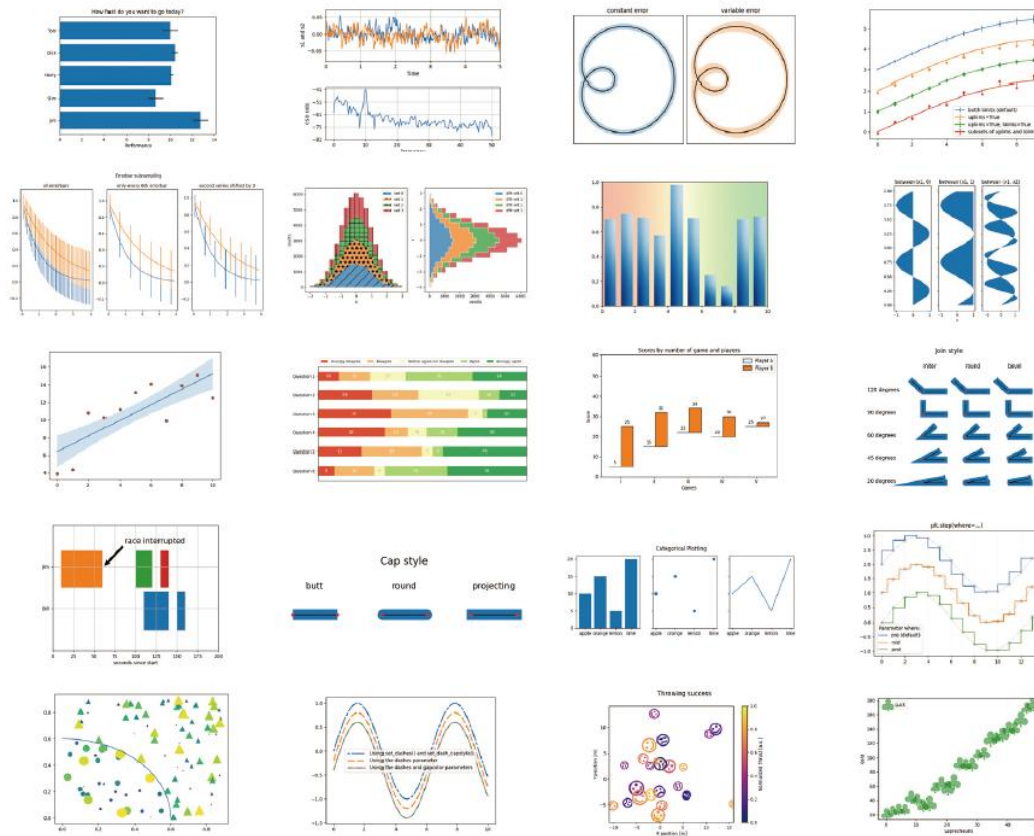


그림 7-1. 데이터 시각화 유형 예

1. 데이터 시각화 이해

■ 시각을 통한 데이터 인사이트 도출

- 많은 양의 데이터를 시각적으로 요약
 - 시각화 요소인 도형의 형태, 크기, 위치, 색 정보 등으로 시각적 패턴을 찾는다.
 - 정확한 전달을 위해 시각화 형태를 잘 선택해야 한다.
 - 데이터 시각화의 목적을 명확히 해야 한다.
- 더 정확한 데이터 분석 결과 도출
 - 데이터 분석 결과를 전달하기 위한 목적으로도 사용된다.
 - 시각화를 활용하여 데이터를 분석하는 시각적 분석으로도 활용된다.

1. 데이터 시각화 이해

■ 데이터 시각화 활용 사례

- 시각화 대시보드를 통한 공동의 데이터 활용
 - 데이터 시각화를 공유하는 방식 중 하나

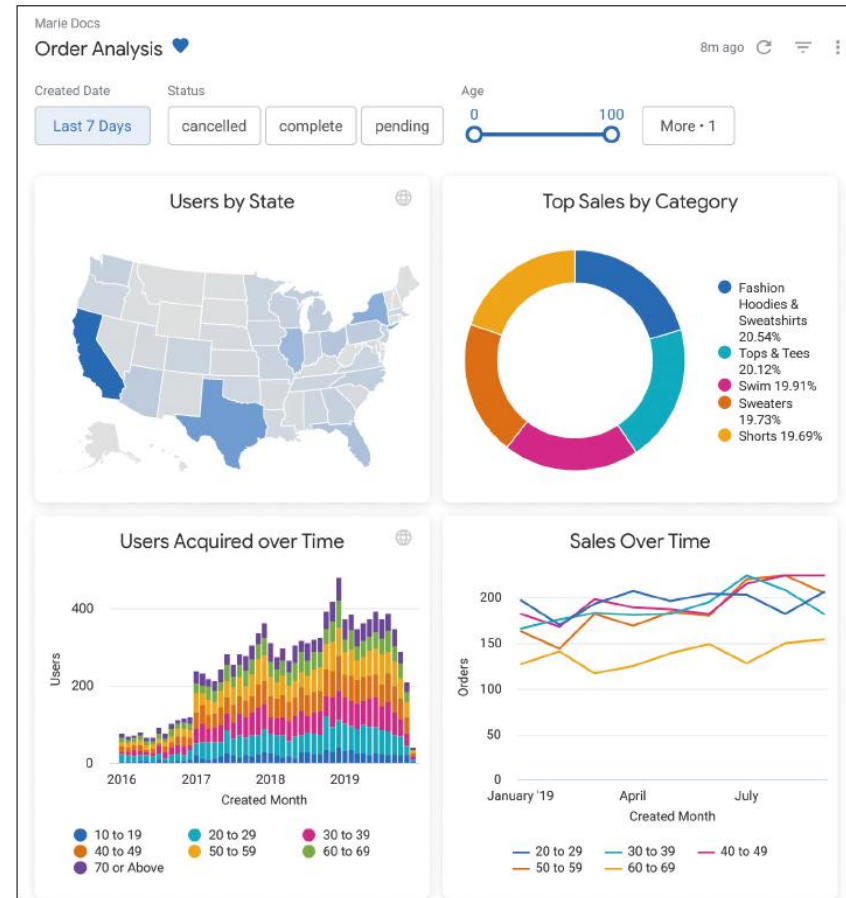


그림 7-2. 데이터 시각화 대시보드 예

1. 데이터 시각화 이해

■ 데이터 시각화 활용 사례

■ 다양한 분야에서의 시각화 활용

- 데이터 시각화는 데이터가 존재하는 모든 분야에서 활용
- 공공데이터 포털 시각화 서비스, 지자체별 공공데이터를 개방하는 다수의 사이트 대표 사례

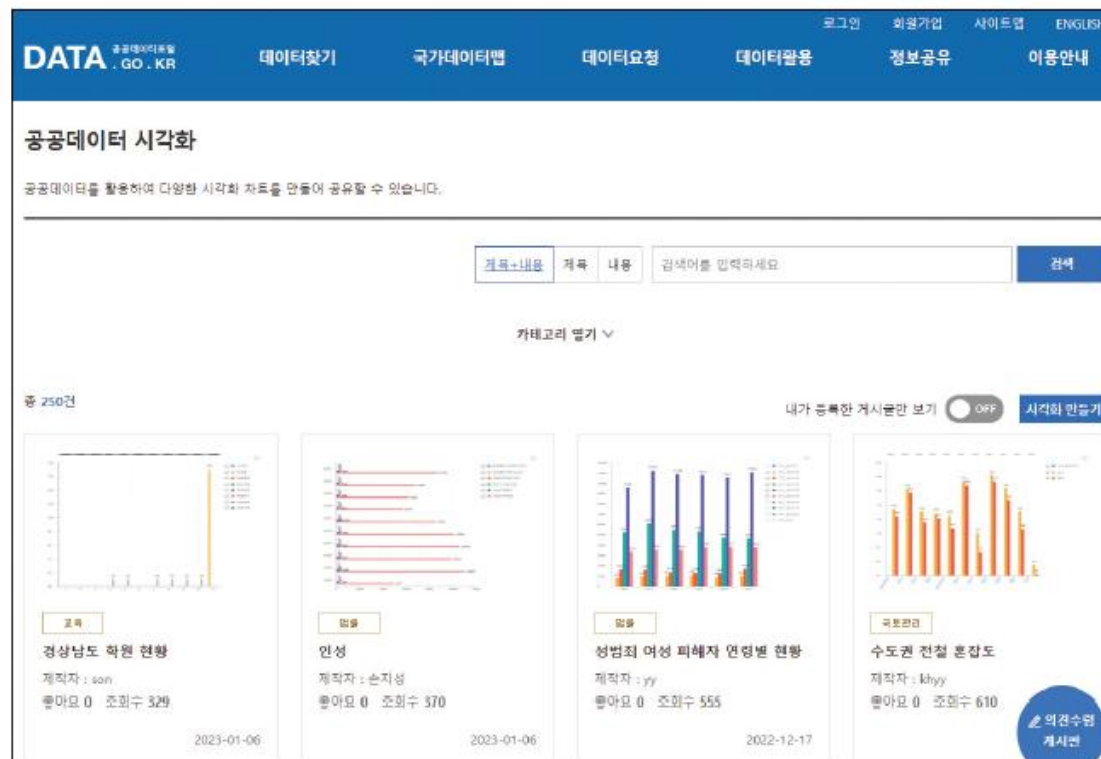


그림 7-3. 공공데이터 포털의 시각화 화면 예

1. 데이터 시각화 이해

■ 파이썬으로 하는 데이터 시각화

- 시각화 라이브러리
 - 데이터 분석 목적에 따라 선택적으로 사용

표 7-1. 시각화 라이브러리

시각화 라이브러리	특징
맷플롯립(Matplotlib)	파이썬에서 가장 많이 사용하며, 데이터를 차트나 플롯으로 시각화하는 라이브러리이다. 판다스의 데이터프레임을 바로 시각화할 때도 내부적으로 맷플롯립을 사용한다. 맷플롯립은 데이터 분석 이전의 데이터 이해를 위한 시각화 또는 데이터 분석 이후의 결과를 시각화하기 위해 사용된다.
시본(Seaborn)	맷플롯립을 기반으로 색 테마, 차트 기능 등을 추가해 주는 라이브러리이다. 맷플롯립과 함께 사용하며 히트맵, 카운트 플롯 등을 제공한다.
폴리움(Folium)	지도 데이터를 이용하여 위치 정보를 시각화하는 라이브러리이다. 자바스크립트 기반으로 상호작용의 그래프를 그릴 수 있다.
파이차트(Pyecharts)	바이두(Baidu)에서 데이터 시각화를 위해 만든 파이썬 버전의 라이브러리이다. 다양한 그래프들이 내장되어 있고 자바스크립트 기반으로 상호작용의 그래프를 그릴 수 있다.
플로트나인(Plotnine)	R의 ggplot2에 기반을 두어 그래프를 그려 주는 라이브러리이다. R 시각화 경험에 있다면 편리하게 사용할 수 있다.
플로틀리(Plotly)	상호작용 그래프를 그려 주는 라이브러리이다. R, 스칼라, 파이썬, 자바스크립트, 매트랩 등에서 사용할 수 있다. 사용이 쉽고 세련된 도구이며 온라인과 오프라인 버전이 따로 존재한다.
보케(Bekeh)	맷플롯립의 대화형 버전으로, 다양한 기능을 제공하고 디자인이 훌륭하며 플롯(축)들 간의 링크가 가능하다. 반면 시본에 비교하여 문법이 복잡하다는 단점이 있다.

1. 데이터 시각화 이해

■ 파이썬으로 하는 데이터 시각화

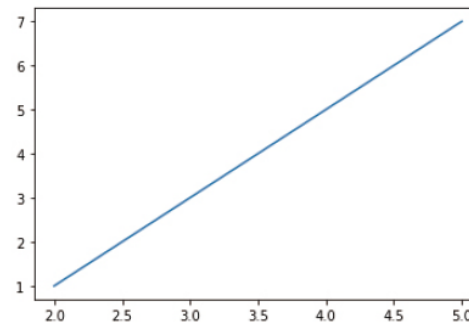
■ 시각화 단계

- 1단계 : 시각화 라이브러리 불러오기
- 2단계 : x축, y축에 표시할 데이터 정하기
- 3단계 : plot() 함수에 데이터 입력하기
- 4단계 : 그래프 보여 주기

기본적인 시각화 문법

1	import matplotlib.pyplot as plt	#1단계
2	xdata=[2, 3, 4, 5]	#2단계
3	ydata=[1, 3, 5, 7]	
4	plt.plot(xdata, ydata)	#3단계
5	plt.show()	#4단계

〈실행결과〉



02

맷플롯립 라이브러리

2. 맷플롯립 라이브러리

■ 맷플롯립

- 그래프를 그릴 때 가장 많이 사용하는 파이썬 라이브러리 중 하나
- 다양한 유형의 그래프를 간편하게 그릴 수 있음.

Lines, bars and markers

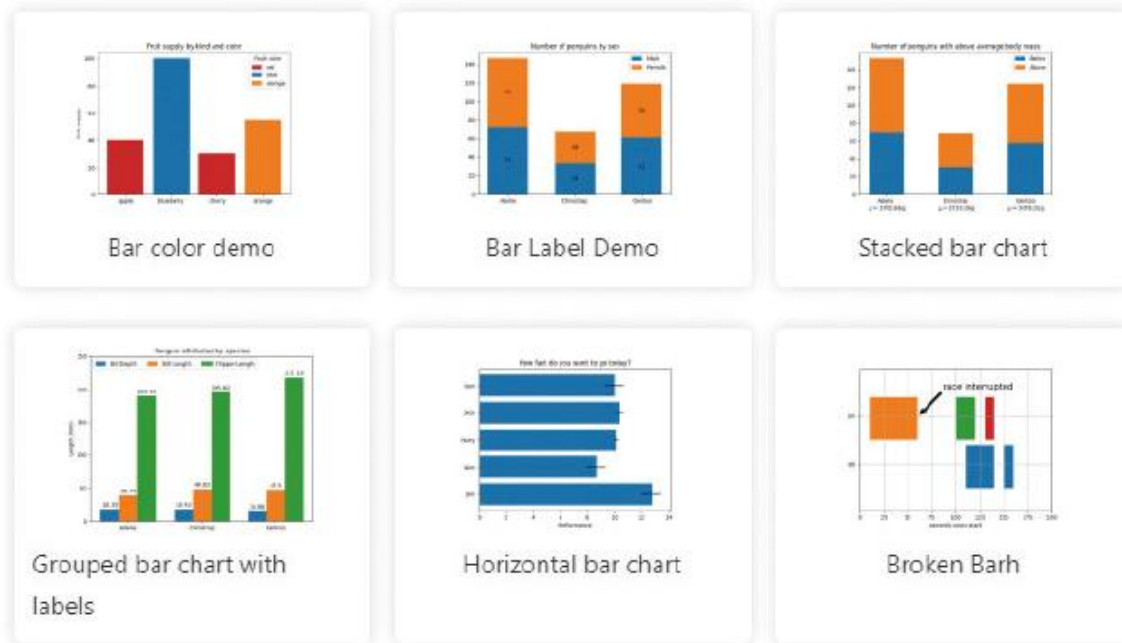


그림 7-4. 맷플롯립 시각화 예

2. 맷플롯립 라이브러리

■ 시각화 옵션

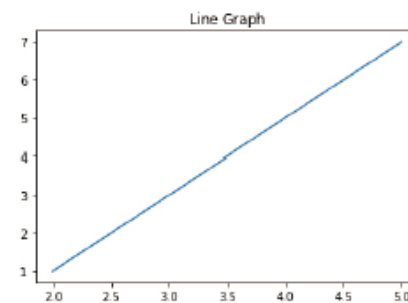
- 제목
 - 그래프 제목은 title() 함수를 사용하여 표현

제목 지정

〈코드〉

```
1 import matplotlib.pyplot as plt
2 plt.title('Line Graph')
3 xdata=[2, 3, 4, 5]
4 ydata=[1, 3, 5, 7]
5 plt.plot(xdata, ydata)
6 plt.show()
```

〈실행결과〉



2. 맷플롯립 라이브러리

■ 시각화 옵션

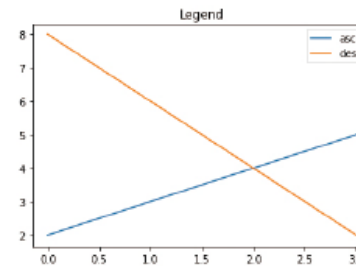
- 범례
 - 두 개 이상의 데이터를 표현할 때 사용

범례 지정

〈코드〉

```
1 import matplotlib.pyplot as plt
2 plt.title('Legend')
3 data1=[2, 3, 4, 5]
4 data2=[8, 6, 4, 2]
5 plt.plot(data1, label='asc')
6 plt.plot(data2, label='desc')
7 plt.legend()
8 plt.show()
```

〈실행결과〉



2. 맷플롯립 라이브러리

■ 시각화 옵션

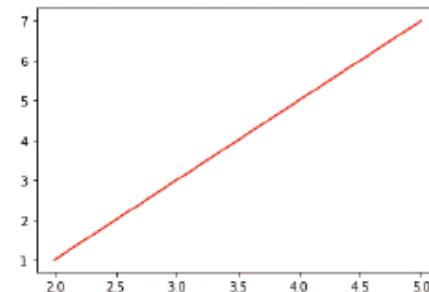
- 색상
 - 색상은 plot() 함수에 color 속성을 추가하여 표현

색상 지정

〈코드〉

```
1 import matplotlib.pyplot as plt
2 xdata=[2, 3, 4, 5]
3 ydata=[1, 3, 5, 7]
4 plt.plot(xdata, ydata, color='red')
5 plt.show()
```

〈실행결과〉



2. 맷플롯립 라이브러리

■ 시각화 옵션

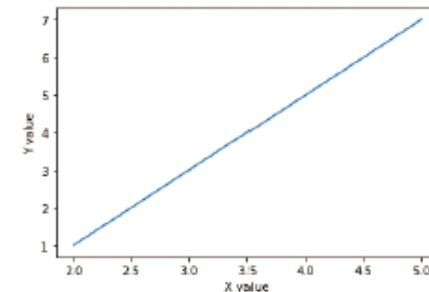
- x축 및 y축 이름
 - x축 이름은 xlabel() 함수를 사용하고 y축 이름은 ylabel() 함수를 사용하여 표현

x축 및 y축 이름 지정

〈코드〉

```
1 import matplotlib.pyplot as plt
2 xdata=[2, 3, 4, 5]
3 ydata=[1, 3, 5, 7]
4 plt.plot(xdata, ydata)
5 plt.xlabel('X value')
6 plt.ylabel('Y value')
7 plt.show()
```

〈실행결과〉



2. 맷플롯립 라이브러리

■ 시각화 옵션

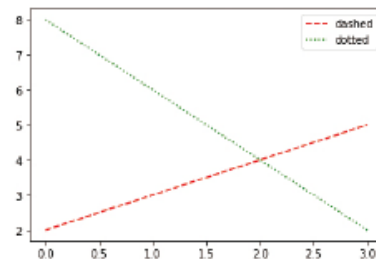
■ 그래프 선 모양

- 실선은 '-', 파선은 '--', 점쇄선은 '-.', 점선은 ':' 기호로 설정

그래프 선 모양 지정

```
1 import matplotlib.pyplot as plt
2 data1=[2, 3, 4, 5]
3 data2=[8, 6, 4, 2]
4 plt.plot(data1, color='r', label='dashed', linestyle='--')
5 plt.plot(data2, color='g', label='dotted', linestyle=':')
6 plt.legend()
7 plt.show()
```

〈실행결과〉



2. 맷플롯립 라이브러리

■ 시각화 옵션

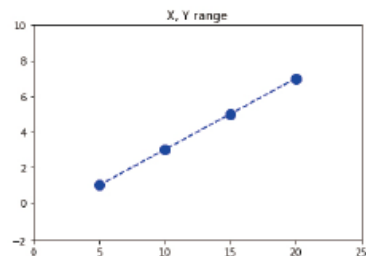
■ 그림 범위 지정

- `xlim()` 함수와 `ylim()` 함수를 사용하여 그림의 범위를 수동으로 조정

x축 및 y축 이름 지정

```
1 import matplotlib.pyplot as plt
2 plt.title('X, Y range')
3 xdata=[5, 10, 15, 20]
4 ydata=[1, 3, 5, 7]
5 plt.plot(xdata, ydata, color='b', linestyle='--', marker='o', markersize='10')
6 plt.xlim(0,25)
7 plt.ylim(-2,10)
8 plt.show()
```

〈실행결과〉



2. 맷플롯립 라이브러리

■ 시각화 옵션

■ 마커 및 색상 속성

- plot() 함수에서 marker 속성을 사용하면 선의 점 모양을 다양하게 변경할 수 있다.
마커와 색상속성은 다음과 같다.



마커 속성	마커 의미
+	더하기
.	점
.	점
o	원형
v	역삼각형
^	삼각형
s	사각형

색상 속성	색상 의미
b	파랑색
g	녹색
r	빨강색
c	청록색
m	마젠타색
y	노랑색
k	검정색
w	흰색

2. 맷플롯립 라이브러리

■ 시각화 옵션

■ 내장 시각화 옵션

- 내장 그래프 도구를 이용하여 기본적인 시각화를 할 수 있다.

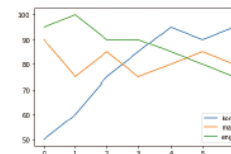
표 7-2. 그래프 속성값

속성값	그래프	속성값	그래프
line	선 그래프	kde	커널 밀도 그래프
bar	수직 막대 그래프	area	면적 그래프
barh	수평 막대 그래프	pie	파이 그래프
hist	히스토그램	scatter	산점도 그래프
box	상자수염 그래프	hexbin	고밀도 산점도 그래프

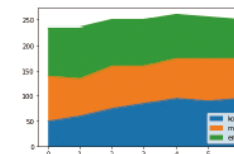
```

1 import pandas as pd
2 my_score=[[50,90,95],[60,75,100],[75,85,90],[85,75,90],[95,80,85],[90,85,80],[95,80,75]]
3 subject=['kor', 'math', 'eng']
4 df=pd.DataFrame(my_score, columns=subject)
5 df.plot(kind='line')    df.plot(kind='area')    df.plot(kind='box')
  
```

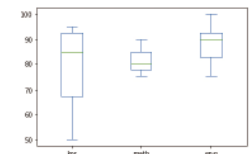
〈실행결과〉



〈실행결과〉



〈실행결과〉



2. 맷플롯립 라이브러리

■ 선 그래프

■ 데이터 셋 정보

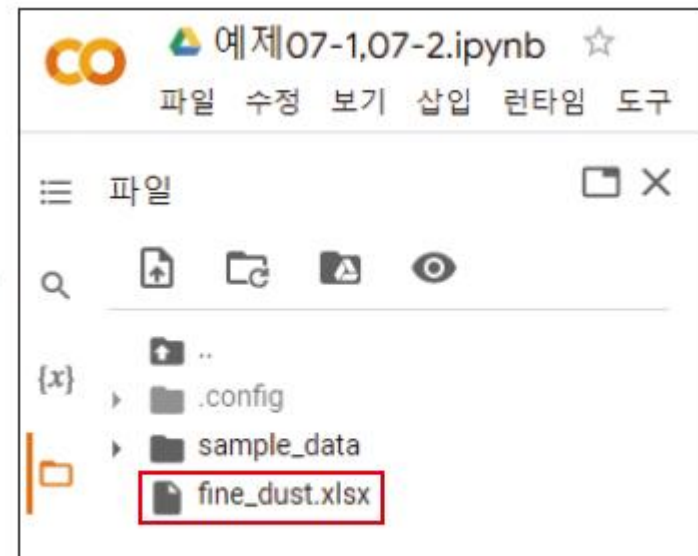
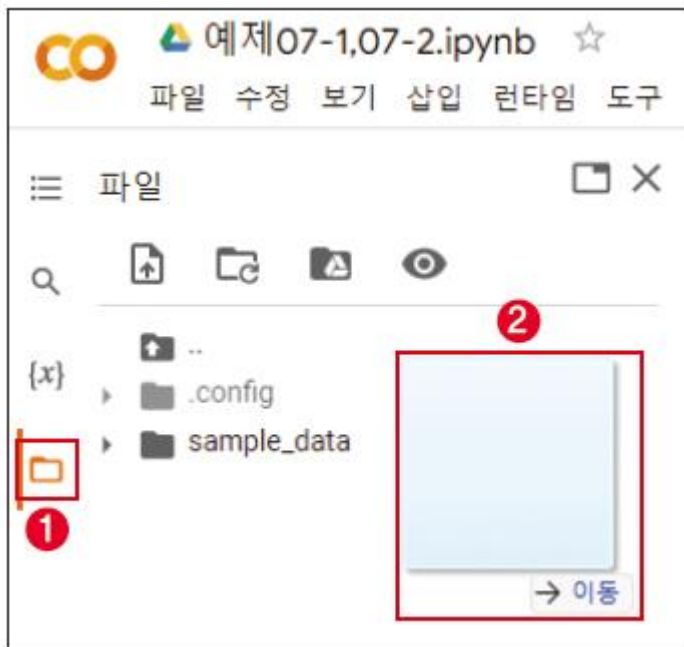
- 에어코리아(<https://www.airkorea.or.kr/>)에서 2001년 ~ 2019년까지의 미세먼지 데이터 'fine_dust.xlsx' 파일을 다운로드

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	area	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019
2	Seoul	71	76	69	61	58	60	61	55	54	49	47	41	45	46	45	48	44	40	42
3	Gyeonggi	71	74	68	67	65	68	66	60	60	58	56	49	54	54	53	53	51	44	46
4	Incheon	52	57	61	62	61	68	64	57	60	55	55	47	49	49	53	49	46	40	43
5	Busan	60	69	55	60	58	59	57	51	49	49	47	43	49	48	46	44	44	41	36
6	Daegu	67	71	59	0	55	54	53	57	48	51	47	42	45	45	46	43	42	39	39
7	Gwangju	57	52	36	46	49	55	52	50	46	45	43	38	42	41	43	40	40	41	42
8	Daejeon	48	53	43	49	48	49	49	45	43	44	44	39	42	41	46	44	45	44	42
9	Ulsan	0	54	40	50	50	52	53	54	49	48	49	46	47	46	46	43	43	40	37
10	Sejong	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	46	47	40	44
11	Gangwon	49	58	57	59	62	58	55	56	53	51	50	47	50	51	50	47	44	38	37
12	Chungcheong	59	54	52	56	52	54	58	55	54	54	50	46	49	47	49	47	44	42	44
13	Jeolla	55	53	45	49	47	49	52	48	50	48	46	43	45	45	44	44	42	39	38
14	Gyeongsang	51	55	52	58	55	54	52	51	48	47	47	44	49	49	46	43	42	43	39
15	Jeju	46	45	37	43	45	49	44	43	42	48	42	34	40	47	44	41	37	35	35

2. 맏플로트립 라이브러리

■ 선 그래프

- 세션 저장소에 업로드
 - 코랩 노트에서 왼쪽 폴더 모양의 '파일' 아이콘을 클릭한다.
 - 파일 화면이 표시되면 'fine_dust.xlsx' 엑셀 파일을 드래그하여 업로드한다.



2. 맷플롯립 라이브러리

■ 선 그래프

- 데이터 읽어와서 선 그래프 그리기

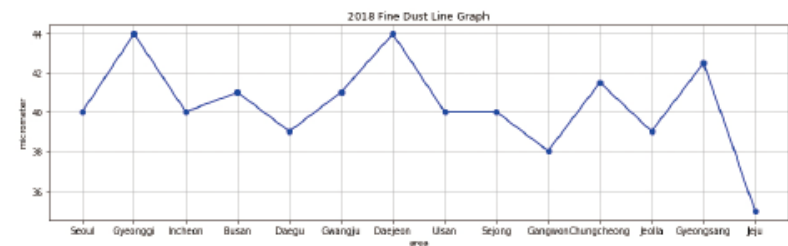
'미세먼지' 엑셀 파일 읽어오기

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 data=pd.read_excel('fine_dust.xlsx', index_col='area')
5 data2018=data[2018]
```

2018년 지역별 미세먼지 선 그래프

```
1 plt.figure(figsize=(15,4))
2 plt.plot(data2018, color='b', marker='o')
3 plt.xlabel('area')
4 plt.ylabel('micrometer')
5 plt.title('2018 Fine Dust Line Graph')
6 plt.grid()
7 plt.show()
```

〈실행결과〉



2. 맷플롯립 라이브러리

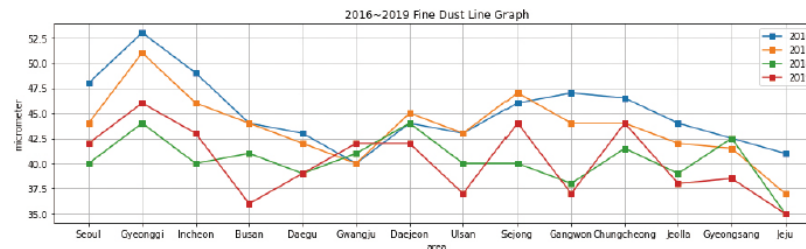
■ 선 그래프

- 데이터 읽어와서 선 그래프 그리기
 - 이러한 과정을 통해 다중 선 그래프를 그리게 된다.

2016년~2019년 지역별 미세먼지 선 그래프

```
1 plt.figure(figsize=(15,4))
2
3 for year in range(2016, 2020):
4     chartdata=data[year]
5     plt.plot(chartdata, marker='s', label=year)
6
7 plt.xlabel('area')
8 plt.ylabel('micrometer')
9 plt.title('2016~2019 Fine Dust Line Graph')
10 plt.legend()
11 plt.grid()
12 plt.show()
```

〈실행결과〉



2. 맷플롯립 라이브러리

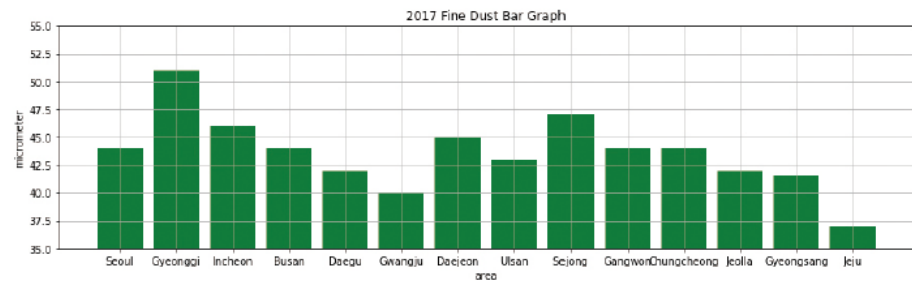
■ 막대 그래프

- 세로 막대 그래프 그리기

2017년 지역별 미세먼지 세로 막대 그래프

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 data=pd.read_excel('fine_dust.xlsx', index_col='area')
5 data2017=data[2017]
6
7 plt.figure(figsize=(15,4))
8 plt.bar(data2017.index, data2017, color='g')
9 plt.ylim(35,55)
10 plt.xlabel('area')
11 plt.ylabel('micrometer')
12 plt.title('2017 Fine Dust Bar Graph')
13 plt.grid()
14 plt.show()
```

〈실행결과〉



2. 맷플롯립 라이브러리

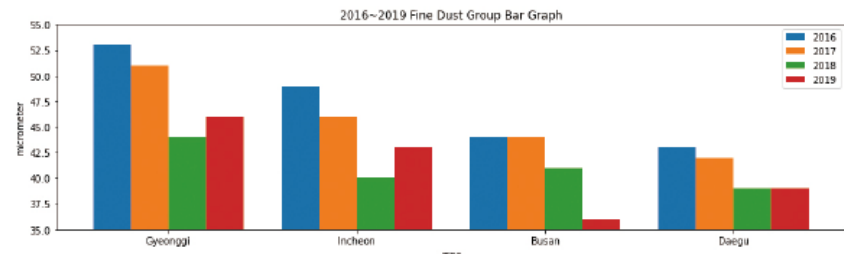
■ 막대 그래프

- 그룹 세로 막대 그래프 그리기

2016년 ~ 2019년 경기, 인천, 부산, 대구 지역별 미세먼지 그룹 세로 막대 그래프

```
1 import numpy as np
2 index=np.arange(4)
3
4 plt.figure(figsize=(15,4))
5
6 data4=data.loc['Gyeonggi':'Daegu', 2016:2019]
7 for year in range(2016, 2020):
8     chartdata=data4[year]
9     plt.bar(index, chartdata, width=0.2, label=year)
10    index=index + 0.2
11
12 plt.ylim(35,55)
13 plt.xlabel('area')
14 plt.ylabel('micrometer')
15 plt.xticks(index-0.5, ['Gyeonggi', 'Incheon', 'Busan', 'Daegu'])
16 plt.title('2016~2019 Fine Dust Group Bar Graph')
17 plt.legend()
18 plt.show()
```

〈실행결과〉



2. 맷플롯립 라이브러리

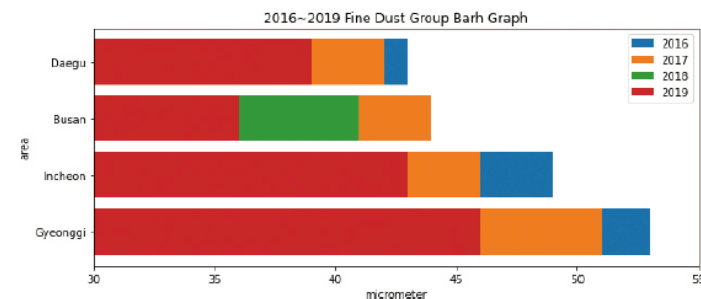
■ 막대 그래프

- 그룹 누적 가로 막대 그래프 그리기

2016년~2019년 지역별 미세먼지 그룹 누적 가로 막대 그래프

```
1 plt.figure(figsize=(10,4))
2 index=np.arange(4)
3
4 data4=data.loc['Gyeonggi':'Daegu', 2016:2019]
5 for year in range(2016, 2020):
6     chartdata=data4[year]
7     plt.barh(index, chartdata, label=year)
8
9 plt.xlim(30,55)
10 plt.ylabel('area')
11 plt.xlabel('micrometer')
12 plt.yticks(index, ['Gyeonggi', 'Incheon', 'Busan', 'Daegu'])
13 plt.title('2016~2019 Fine Dust Group Barh Graph')
14 plt.legend()
15 plt.show()
```

〈실행결과〉



2. 맷플롯립 라이브러리

■ 산점도 그래프

■ 데이터 셋 정보

- 공공데이터포털(<https://www.data.go.kr/>)에서 2020년 건강검진 일부 데이터 'health screenings 2020 1000ea.xlsx' 파일을 다운로드

변수명	데이터형	설 명			
year	int64	기준년도(2020년)	diastolic	int64	이완기 혈압
city_code	int64	시도 코드	blood_sugar	int64	식전혈당(공복혈당)
gender	int64	성별 코드(남자 : 1, 여자 : 2)	cholesterol	int64	총 콜레스테롤
age_code	int64	연령대 코드(5세 단위)	triglycerides	int64	트리글리세라이드
height	int64	신장(5cm 단위)	HDL	int64	HDL 콜레스테롤
weight	int64	몸무게(5kg 단위)	LDL	float64	LDL 콜레스테롤
waist	float64	허리둘레	hemoglobin	float64	혈색소
eye_left	float64	시력(좌)	urine_protein	float64	요단백
eye_right	float64	시력(우)	serum	float64	혈청크레아티닌
hear_left	float64	청력(좌)	AST	int64	(혈청지오티)AST
hear_right	float64	청력(우)	ALT	int64	(혈청지피티)ALT
systolic	int64	수축기 혈압	GTP	int64	감마 지티피
			smoking	int64	흡연상태 (비흡연 : 1, 흡연 끊음 : 2, 흡연 : 3)
			drinking	float64	음주 여부 (비음주 : 0, 음주 : 1)
			oral_check	int64	구강검진 수검여부 (미수검 : 0, 수검 : 1)
			dental_caries	float64	치아우식증 유무 (없음 : 0, 있음 : 1)
			tartar	float64	치석 (없음 : 0, 있음 : 1)
			open_date	int64	데이터 공개일자

2. 맷플롯립 라이브러리

■ 산점도 그래프

■ 데이터 읽어와서 산점도 그래프 그리기

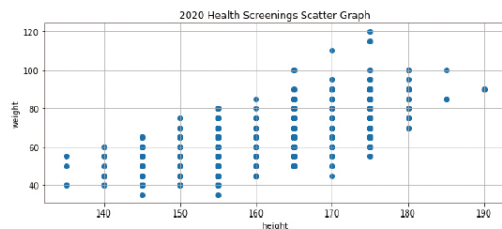
'2020년 건강검진' 엑셀 파일 읽어오기

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3 data=pd.read_excel('health_screenings_2020_1000ea.xlsx')
```

키 및 몸무게 산점도 그래프

```
1 heightdata=data['height']
2 weightdata=data['weight']
3
4 plt.figure(figsize=(10,4))
5 plt.scatter(heightdata, weightdata)
6
7 plt.xlabel('height')
8 plt.ylabel('weight')
9 plt.title('2020 Health Screenings Scatter Graph')
10 plt.grid()
11 plt.show()
```

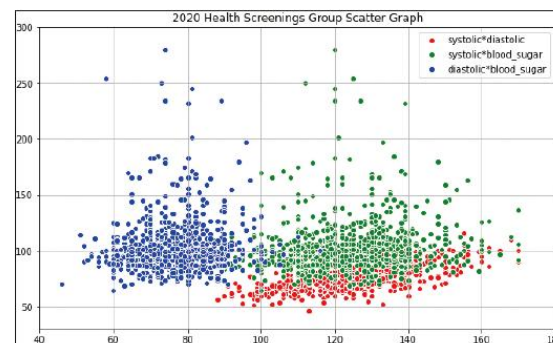
〈실행결과〉



수축기 혈압, 이완기 혈압, 공복 혈당 그룹 산점도 그래프

```
1 systolic_data=data['systolic']
2 diastolic_data=data['diastolic']
3 blood_sugar_data=data['blood_sugar']
4
5 plt.figure(figsize=(10,6))
6
7 plt.scatter(systolic_data, diastolic_data, color='r', edgecolor='w',
8 label='systolic*diastolic')
9 plt.scatter(systolic_data, blood_sugar_data, color='g', edgecolor='w',
10 label='systolic*blood_sugar')
11 plt.scatter(diastolic_data, blood_sugar_data, color='b', edgecolor='w',
12 label='diastolic*blood_sugar')
13
14 plt.xlim(40,180)
15 plt.ylim(30,300)
16 plt.title('2020 Health Screenings Group Scatter Graph')
17 plt.legend()
18 plt.grid()
19 plt.show()
```

〈실행결과〉



2. 맷플롯립 라이브러리

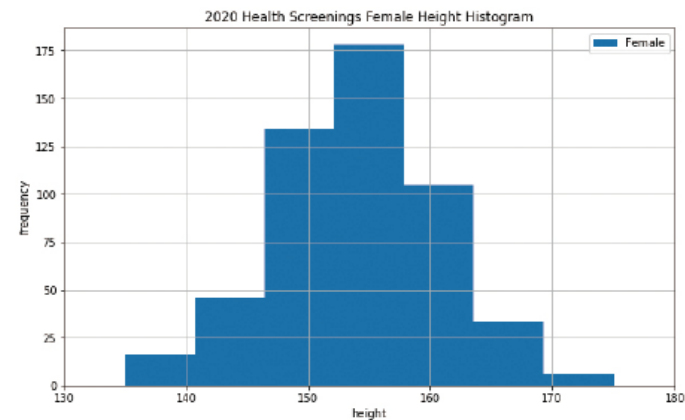
■ 히스토그램 그래프

■ 히스토그램 그리기

남성 키 히스토그램

```
1 female_data=data.loc[data.gender==2,['gender','height']]
2
3 plt.figure(figsize=(10,6))
4 plt.hist(female_data['height'], bins=7, label='Female')
5
6 plt.xlim(130,180)
7 plt.xlabel('height')
8 plt.ylabel('frequency')
9 plt.title('2020 Health Screenings Female Height Histogram')
10 plt.legend()
11 plt.grid()
12 plt.show()
```

〈실행결과〉



2. 맷플롯립 라이브러리

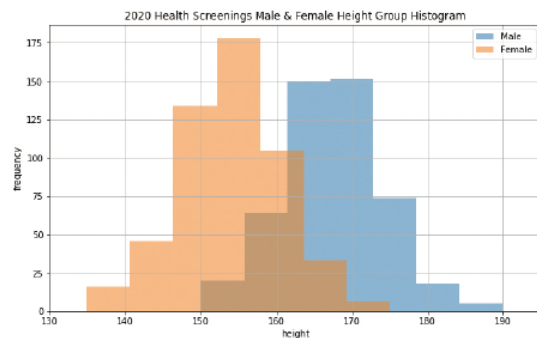
■ 히스토그램 그래프

■ 그룹 히스토그램 그리기

남성 및 여성 키 그룹 히스토그램

```
1 male_data=data.loc[data.gender==1,['gender','height']]
2 female_data=data.loc[data.gender==2,['gender','height']]
3
4 plt.figure(figsize=(10,6))
5 plt.hist(male_data['height'], bins=7, alpha=0.5, label='Male')
6 plt.hist(female_data['height'], bins=7, alpha=0.5, label='Female')
7
8 plt.xlim(130,195)
9 plt.xlabel('height')
10 plt.ylabel('frequency')
11 plt.title('2020 Health Screenings Male & Female Height Group Histogram')
12 plt.legend()
13 plt.grid()
14 plt.show()
```

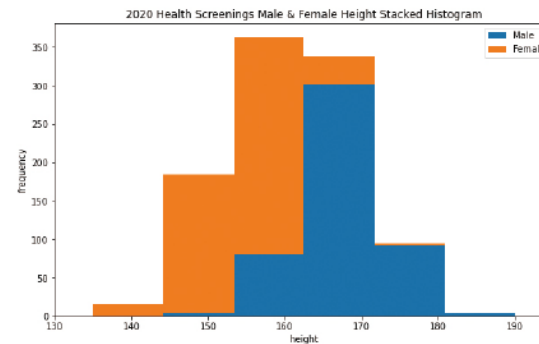
〈실행결과〉



남성 및 여성 키 그룹 누적 히스토그램

```
1 import numpy as np
2 height=np.array([male_data['height'], female_data['height']], dtype=object)
3
4 plt.figure(figsize=(10,6))
5 plt.hist(height, bins=6, label=['Male','Female'], stacked=True)
6
7 plt.xlim(130,195)
8 plt.xlabel('height')
9 plt.ylabel('frequency')
10 plt.title('2020 Health Screenings Male & Female Height Stacked Histogram')
11 plt.legend()
12 plt.show()
```

〈실행결과〉



2. 맷플롯립 라이브러리

■ 상자수염 그래프

- 데이터 분포를 시각화하여 탐색하는 방법

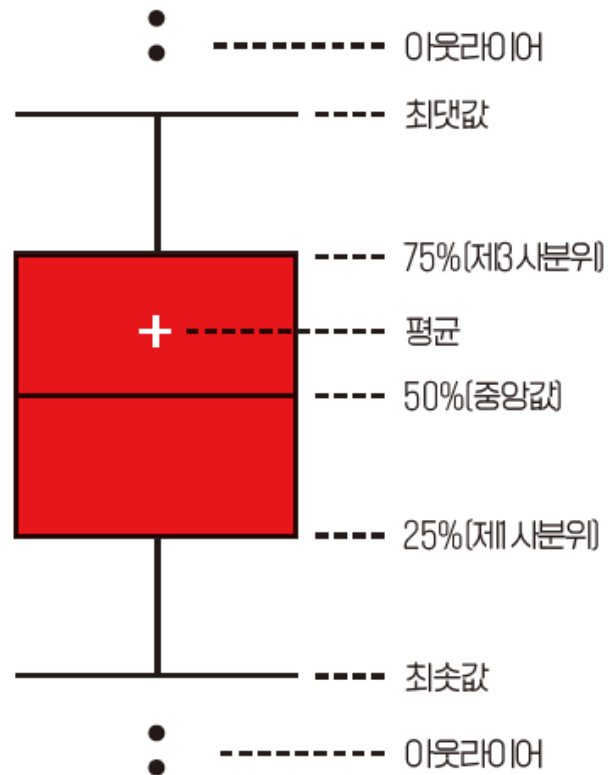


그림 7-5. 상자수염 예시

2. 맷플롯립 라이브러리

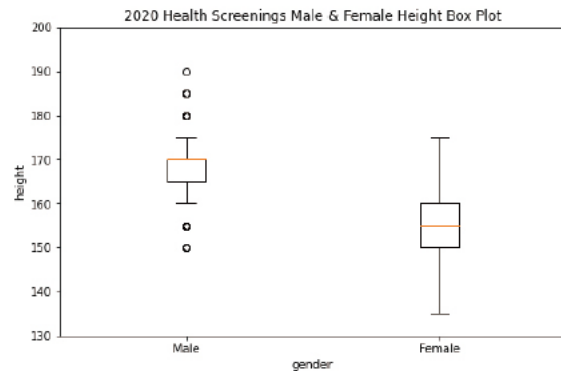
■ 상자수염 그래프

■ 상자수염 그래프 그리기

남성 및 여성 키 상자수염 그래프

```
1 import numpy as np
2 height=np.array([male_data['height'], female_data['height']], dtype=object)
3
4 plt.figure(figsize=(8,5))
5 plt.boxplot(height, labels=['Male', 'Female'])
6
7 plt.ylim(130,200)
8 plt.xlabel('gender')
9 plt.ylabel('height')
10 plt.title('2020 Health Screenings Male & Female Height Box Plot')
11 plt.show()
```

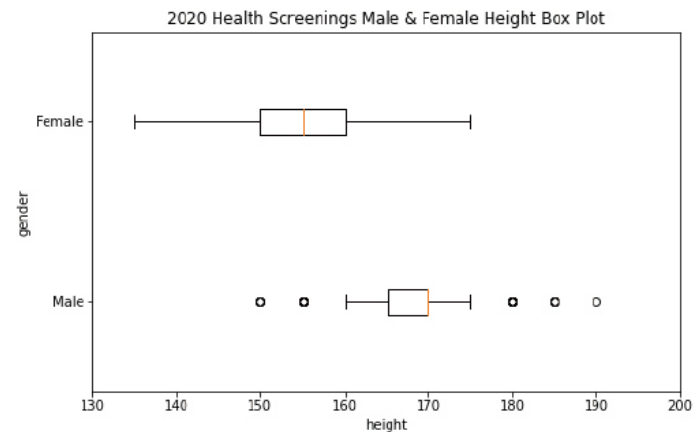
〈실행결과〉



남성 및 여성 키 가로 상자수염 그래프

```
1 plt.figure(figsize=(8,5))
2 plt.boxplot(height, labels=['Male', 'Female'], vert=False)
3
4 plt.xlim(130,200)
5 plt.ylabel('gender')
6 plt.xlabel('height')
7 plt.title('2020 Health Screenings Male & Female Height Box Plot')
8 plt.show()
```

〈실행결과〉



03

시본 라이브러리

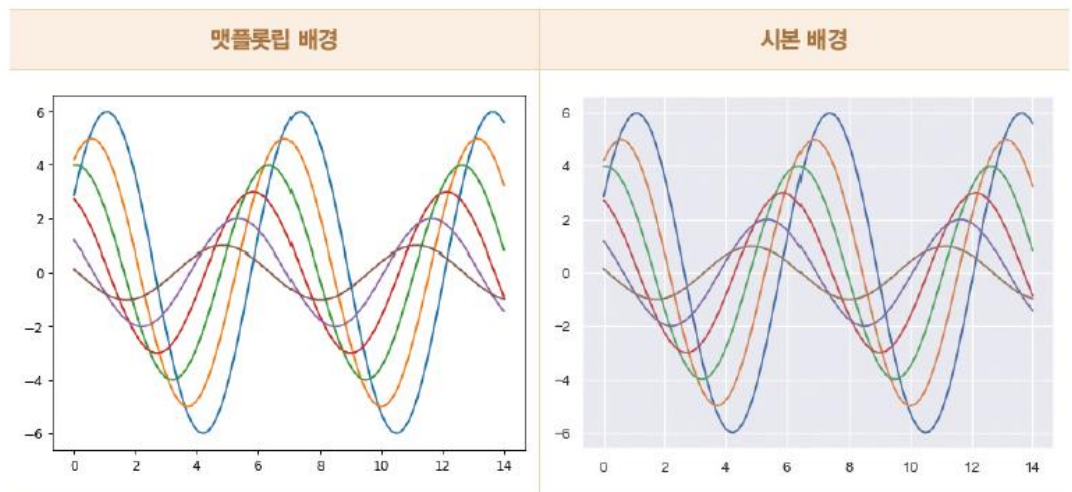
3. 시본 라이브러리

■ 시본

- 맷플롯립을 기반으로 다양한 테마와 통계용 차트 등의 동적인 기능을 추가한 시각화 라이브러리

표 7-3. 시본 라이브러리의 특징 및 단계

시본 주요 특징	시본 시각화 단계
뛰어난 시각화 효과	① 데이터 준비
간결한 구문 제공	② 배경 설정
판다스 데이터프레임에 최적화	③ 시각화
쉬운 데이터프레임 집계 및 차트 요약	④ 개별 그래프 상세 설정



3. 시본 라이브러리

■ 데이터 시각화 준비하기

- 라이브러리 및 데이터 읽어오기
 - 2020년 건강검진 'health_screenings_2020_1000ea.xlsx' 엑셀 파일을 읽어와 동적인 시각화를 표현

라이브러리 импорт 및 '2020년 건강검진' 엑셀 파일 읽어오기

```
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import seaborn as sns
5
6 data=pd.read_excel('health_screenings_2020_1000ea.xlsx')
```

3. 시본 라이브러리

■ 데이터 시각화 준비하기

- 데이터 전처리
 - 문자열로 변경하여 데이터 전처리하는 과정

'2020년 건강검진' 정보에서 성별, 음주 여부, 흡연상태 데이터 전처리

```
1 data6=data.loc[:,['gender', 'height', 'weight', 'waist', 'drinking', 'smoking']]
2 data6.loc[data6['gender']==1,['gender']]='Male'
3 data6.loc[data6['gender']==2,['gender']]='Female'
4 data6.loc[data6['drinking']==0,['drinking']]='Non-drinking'
5 data6.loc[data6['drinking']==1,['drinking']]='Drinking'
6 data6.loc[(data6['smoking']==1) | (data6['smoking']==2),['smoking']]='Non-smoking'
7 data6.loc[data6['smoking']==3,['smoking']]='Smoking'
```

3. 시본 라이브러리

■ 시본 막대 그래프

- 데이터 준비하기

음주 여부 및 흡연 상태 데이터 준비하기

```
1      drinking=data6.groupby(['gender', 'drinking'])['drinking'].count()
2      smoking=data6.groupby(['gender', 'smoking'])['smoking'].count()
3      drinking=drinking.to_frame(name='count')
4      smoking=smoking.to_frame(name='count')
5      drinking=drinking.reset_index()
6      smoking=smoking.reset_index()
```

3. 시본 라이브러리

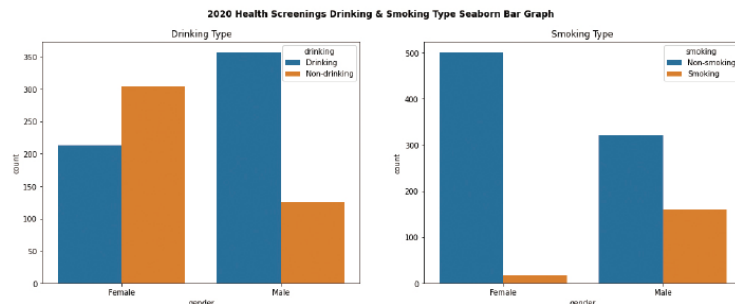
■ 시본 막대 그래프

■ 막대 그래프 그리기

성별 음주 여부 및 흡연상태 시본 막대 그래프

```
1 fig=plt.figure(figsize=(17,6))
2
3 area1=fig.add_subplot(1,2,1)
4 area2=fig.add_subplot(1,2,2)
5
6 ax1=sns.barplot(data=drinking,x='gender',y='count',hue='drinking',ax=area1)
7 ax2=sns.barplot(data=smoking,x='gender',y='count',hue='smoking',ax=area2)
8
9 fig.suptitle('2020 Health Screenings Drinking & Smoking Type Seaborn Bar Graph', fontweight='bold')
10 area1.set_title('Drinking Type')
11 area2.set_title('Smoking Type')
12
13 plt.show()
```

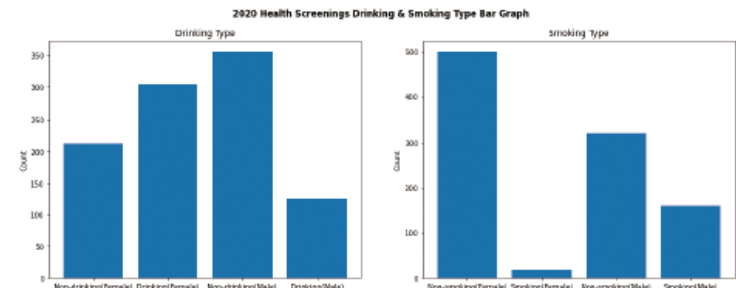
〈실행결과〉



성별 음주 여부 및 흡연 상태 막대 그래프

```
1 fig=plt.figure(figsize=(17,6))
2 fig.suptitle('2020 Health Screenings Drinking & Smoking Type Bar Graph',
3 fontweight='bold')
4 index=np.arange(4)
5
6 fig.add_subplot(1,2,1)
7
8 plt.bar(index, drinking['count'])
9 plt.title('Drinking Type')
10 plt.ylabel('Count')
11 plt.xticks(index, ['Non-drinking(Female)', 'Drinking(Female)', 'Non-drinking(Male)',
12 'Drinking(Male)'])
13
14 fig.add_subplot(1,2,2)
15
16 plt.bar(index, smoking['count'])
17 plt.title('Smoking Type')
18 plt.ylabel('Count')
19 plt.xticks(index, ['Non-smoking(Female)', 'Smoking(Female)', 'Non-smoking(Male)',
20 'Smoking(Male)'])
21
22 plt.show()
```

〈실행결과〉



3. 시본 라이브러리

■ 시본 막대 그래프



- `add_subplot()` 함수
 - `add_subplot()` 함수의 인자를 통해 서브플롯 개수를 조정한다.
 - `add_subplot(1, 2, 1)`은 1x2(행x열)의 서브플롯을 생성한다는 의미한다.
 - 세 번째 인자 1은 생성된 두 개의 서브플롯 중 첫 번째 서브플롯을 의미한다.
 - 마찬가지로 (1, 2, 2)는 1x2 서브플롯에서 두 번째 서브플롯을 의미한다.

3. 시본 라이브러리

■ 시본 산점도 그래프

- 데이터 준비하기

남성 및 여성의 성별, 몸무게, 허리둘레, 음주 여부, 흡연상태 데이터 가져오기

1	<code>male_data = data6.loc[data6.gender == 'Male', ['gender', 'weight', 'waist', 'drinking', 'smoking']]</code>
2	<code>female_data = data6.loc[data6.gender == 'Female', ['gender', 'weight', 'waist', 'drinking', 'smoking']]</code>

3. 시본 라이브러리

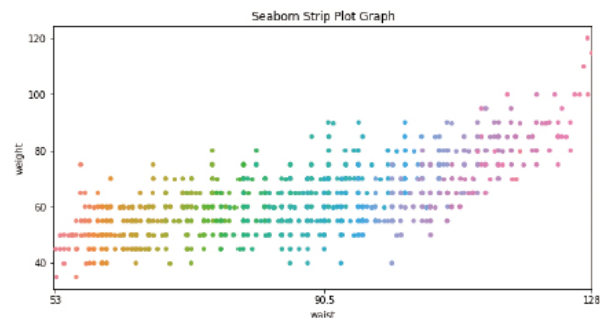
■ 시본 산점도 그래프

■ 시본 스트립 플롯 그래프

남성과 여성의 허리둘레 및 몸무게 시본 스트립 플롯 그래프

```
1 plt.figure(figsize=(10,5))
2 plt.title('Seaborn Strip Plot Graph')
3
4 sns.stripplot(data=male_data, x='waist', y='weight')
5 sns.stripplot(data=female_data, x='waist', y='weight')
6
7 plt.xticks(np.arange(0,127,63), labels=[53, 90.5, 128])
8 plt.show()
```

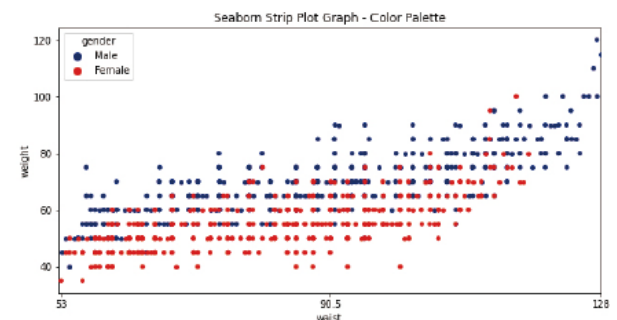
〈실행결과〉



남성과 여성의 허리둘레 및 몸무게 시본 스트립 플롯 그래프 - 색상 팔레트 지정

```
1 plt.figure(figsize=(10,5))
2 plt.title('Seaborn Strip Plot Graph - Color Palette')
3
4 sns.stripplot(data=male_data, x='waist', y='weight', hue='gender',palette='dark')
5 sns.stripplot(data=female_data, x='waist', y='weight', hue='gender',palette='Set1')
6
7 plt.xticks(np.arange(0,127,63), labels=[53, 90.5, 128])
8 plt.show()
```

〈실행결과〉



3. 시본 라이브러리

■ 시본 산점도 그래프

■ 시본 스웸 플롯 그래프 그리기

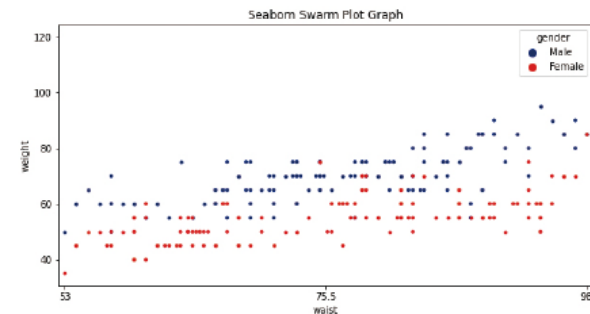
남성 및 여성 데이터 100개 가져오기

```
1 male_data_100=male_data.head(100)
2 female_data_100=female_data.head(100)
```

남성과 여성의 허리둘레 및 몸무게 시본 스웸 플롯 그래프

```
1 plt.figure(figsize=(10,5))
2 plt.title('Seaborn Swarm Plot Graph')
3
4 sns.swarmplot(data=male_data_100, x='waist', y='weight', hue='gender',
5               palette='dark',size=4)
6 sns.swarmplot(data=female_data_100, x='waist', y='weight', hue='gender',
7               palette='Set1',size=4)
8
9 plt.xticks(np.arange(0,46,22.5), labels=[53, 75.5, 98])
10 plt.show()
```

〈실행결과〉



3. 시본 라이브러리

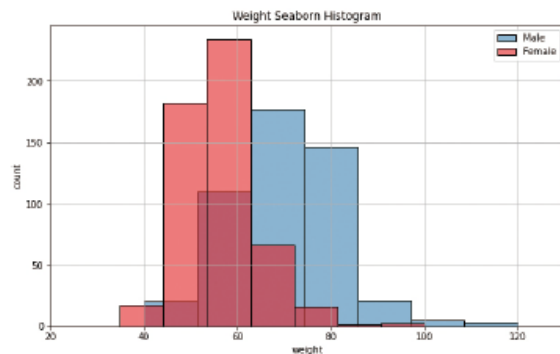
■ 시본 히스토그램 그래프

■ 시본 히스토그램 그래프 그리기

남성과 여성의 몸무게 히스토그램 그래프

```
1 plt.figure(figsize=(10,6))
2 sns.histplot(male_data['weight'], bins=7, alpha=0.5, label='Male')
3 sns.histplot(female_data['weight'], bins=7, alpha=0.5, label='Female',
4 color='r')
5
6 plt.xlim(20,130)
7 plt.xlabel('weight')
8 plt.ylabel('count')
9 plt.title('Weight Seaborn Histogram')
10 plt.legend()
11 plt.grid()
12 plt.show()
```

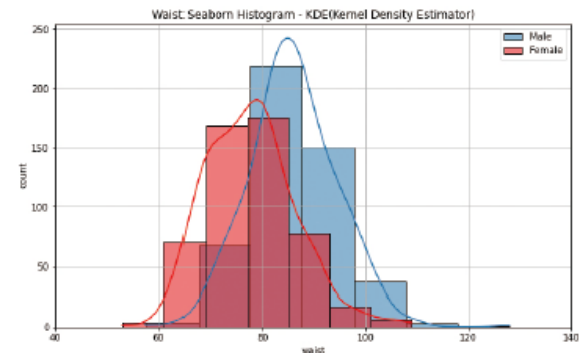
〈실행결과〉



남성과 여성의 허리둘레 히스토그램 - 커널 밀도 추정(KDE) 속성 지정

```
1 plt.figure(figsize=(10,6))
2 sns.histplot(male_data['waist'], bins=7, alpha=0.5, label='Male', kde=True)
3 sns.histplot(female_data['waist'], bins=7, alpha=0.5, label='Female',
4 color='r', kde=True)
5
6 plt.xlim(40,140)
7 plt.xlabel('waist')
8 plt.ylabel('count')
9 plt.title('Waist Seaborn Histogram - KDE(Kernel Density Estimator)')
10 plt.legend()
11 plt.grid()
12 plt.show()
```

〈실행결과〉



3. 시본 라이브러리

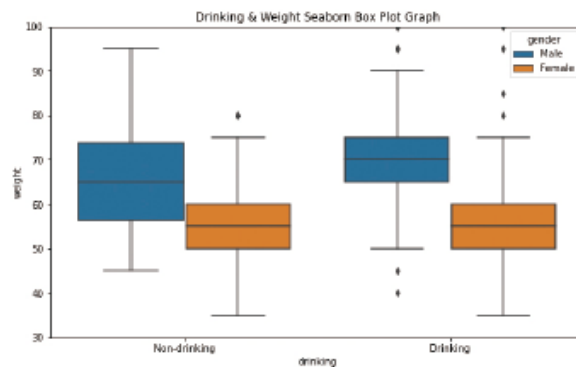
■ 시본 상자수염 그래프

■ 시본 상자수염 그래프 그리기

음주 여부와 몸무게 상자수염 그래프

```
1 plt.figure(figsize=(10,6))
2 plt.title('Drinking & Weight Seaborn Box Plot Graph')
3
4 sns.boxplot(data=data6, x='drinking', y='weight', hue='gender')
5 plt.ylim(30,100)
6 plt.show()
```

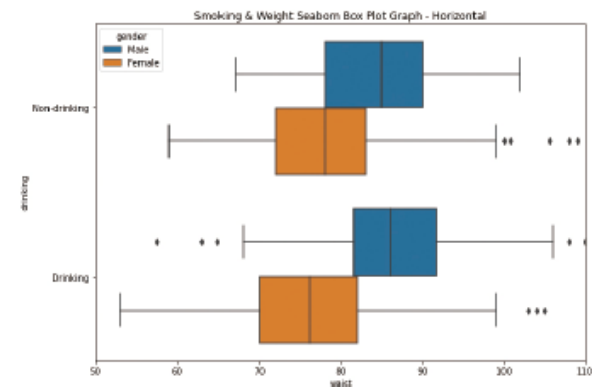
〈실행결과〉



음주 여부와 허리둘레 가로 상자수염 그래프

```
1 plt.figure(figsize=(10,7))
2 plt.title('Drinking & Waist Seaborn Box Plot Graph - Horizontal')
3
4 sns.boxplot(data=data6, x='waist', y='drinking', hue='gender', orient='h')
5 plt.xlim(50,110)
6 plt.show()
```

〈실행결과〉



3. 시본 라이브러리

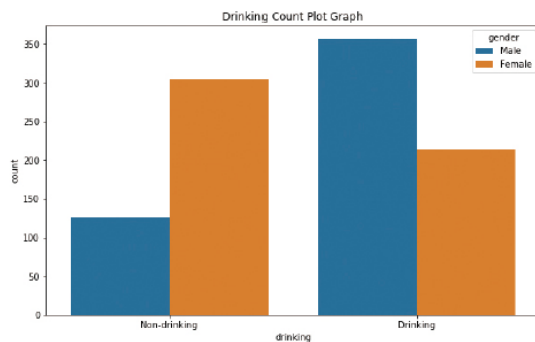
■ 카운트 플롯 그래프

■ 카운트 플롯 그래프 그리기

성별 음주 여부 카운트 플롯 그래프

```
1 plt.figure(figsize=(10,6))
2 plt.title('Drinking Count Plot Graph')
3 sns.countplot(data=data6, x='drinking', hue='gender')
4 plt.show()
```

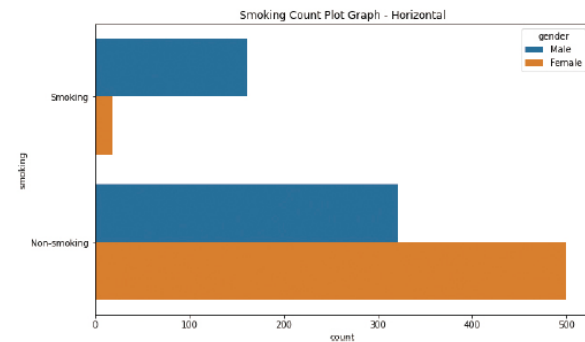
〈실행결과〉



성별 흡연상태 가로 카운트 플롯 그래프

```
1 plt.figure(figsize=(10,6))
2 plt.title('Smoking Count Plot Graph - Horizontal')
3 sns.countplot(data=data6, y='smoking', hue='gender', order=['Smoking',
4 'Non-smoking'])
5 plt.show()
```

〈실행결과〉



3. 시본 라이브러리

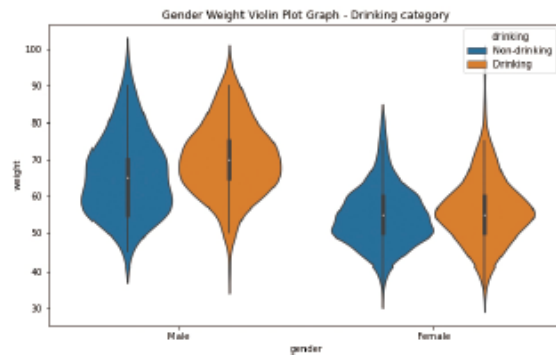
■ 바이올린 플롯 그래프

■ 바이올린 플롯 그래프 그리기

성별 몸무게를 음주 여부로 분리하여 그린 바이올린 플롯 그래프

```
1 plt.figure(figsize=(10,6))
2 plt.title('Gender Weight Violin Plot Graph - Drinking category')
3 sns.violinplot(data=data6[data6.weight<100], x='gender', y='weight',
4               hue='drinking')
5 plt.show()
```

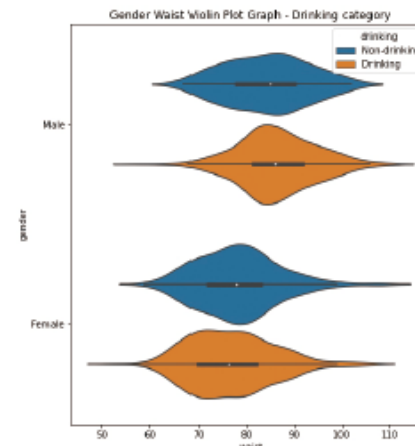
〈실행결과〉



성별 허리둘레를 음주 여부로 분리하여 그린 가로 바이올린 플롯 그래프

```
1 plt.figure(figsize=(7,8))
2 plt.title('Gender Waist Violin Plot Graph - Drinking category')
3 sns.violinplot(data=data6[data6.waist<120], y='gender', x='waist',
4               hue='drinking')
5 plt.show()
```

〈실행결과〉



3. 시본 라이브러리

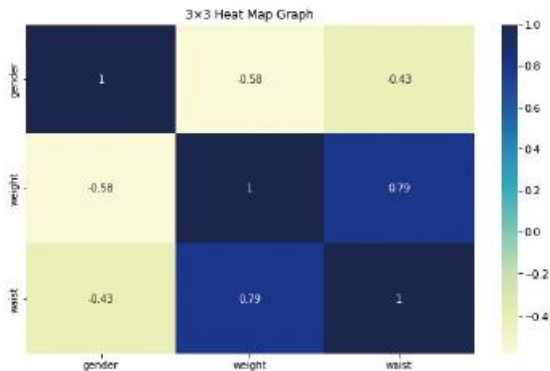
■ 시본 히트맵 그래프

■ 바이올린 플롯 그래프 그리기

3×3 히트맵 그래프

```
1 plt.figure(figsize=(10,6))
2 plt.title('3×3 Heat Map Graph')
3
4 correlation_data3=data3.corr()
5 sns.heatmap(correlation_data3, annot=True, cmap='YlGnBu')
6 plt.show()
```

<실행결과>

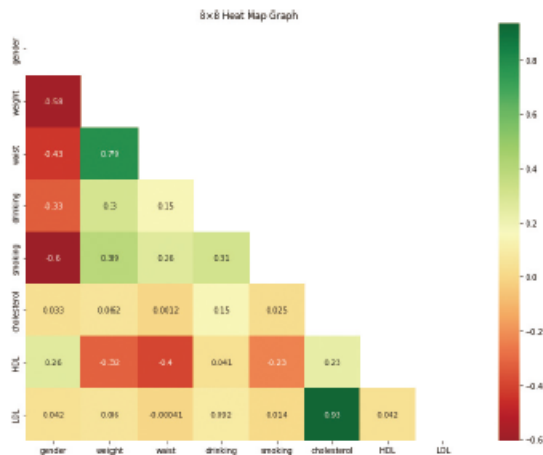


```
1 data3=data.loc[:,['gender','weight','waist']]
2 data8=data.loc[:,['gender','weight','waist','drinking','smoking','cholesterol',
   'HDL','LDL']]
```

8×8 히트맵 그래프

```
1 plt.figure(figsize=(13,10))
2 plt.title('8×8 Heat Map Graph')
3
4 correlation_data8= data8.corr()
5 upp_mat=np.triu(correlation_data8)
6
7 sns.heatmap(correlation_data8, annot=True, cmap='RdYlGn', mask=upp_mat)
8 plt.show()
```

<실행결과>



3. 시본 라이브러리

■ 시본 히트맵 그래프

- 바이올린 플롯 그래프 그리기



- triu() 함수와 tril() 함수
 - triu() 함수 : 히트맵 그래프에서 1로 표시된 대각선을 중심으로 오른쪽 상단의 삼각형 영역을 보이지 않도록 한다.
 - tril() 함수 : 히트맵 그래프에서 1로 표시된 대각선을 중심으로 왼쪽 하단의 삼각형 영역을 보이지 않도록 한다.

3. 시본 라이브러리

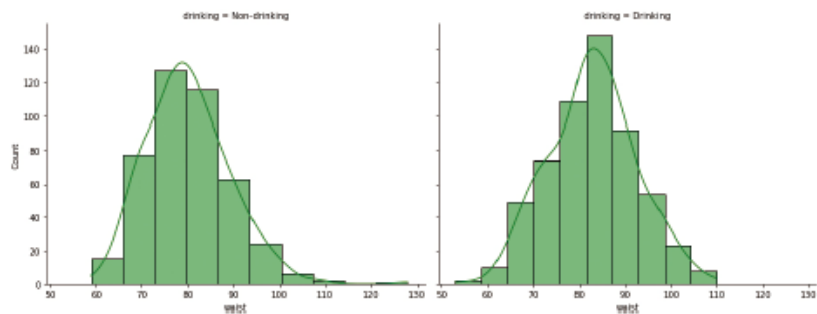
■ 다중 플롯 그리드 그래프

■ 패싯 그리드 그래프

열 기준으로 나눈 패싯 그리드 그래프

```
1 fg=sns.FacetGrid(data6, col='drinking', height=5, aspect=1.3)
2 fg.map(sns.histplot, 'waist', bins=10, color='g', kde=True)
3 plt.show()
```

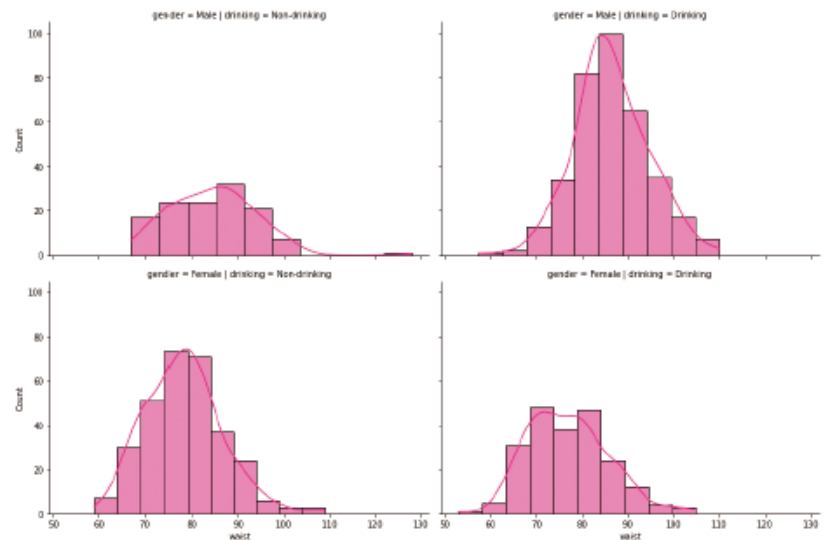
〈실행결과〉



행과 열 기준으로 나눈 패싯 그리드 그래프

```
1 fg=sns.FacetGrid(data6, row='gender', col='drinking', height=4.5, aspect=1.5)
2 fg.map(sns.histplot, 'waist', bins=10, color='deeppink', kde=True)
3 plt.show()
```

〈실행결과〉



3. 시본 라이브러리

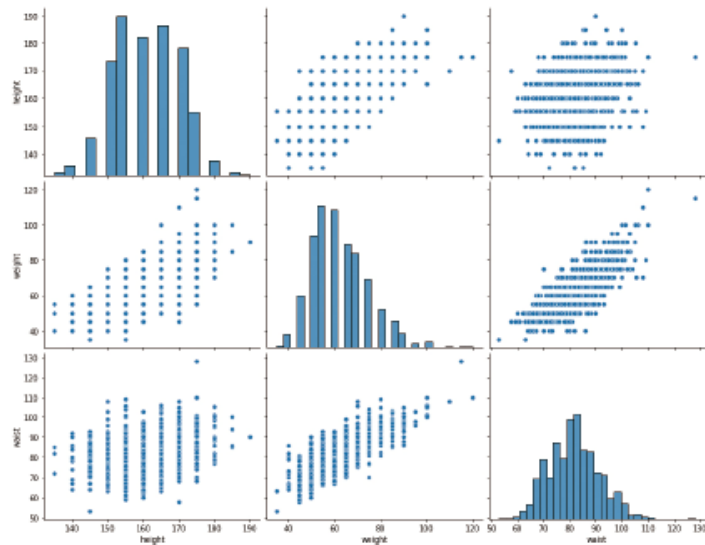
■ 다중 플롯 그리드 그래프

■ 패킷 그리드 그래프

페어 플롯 그래프

```
1 sns.pairplot(data6, height=3.3, aspect=1.3)
2 plt.show()
```

<실행결과>



PairGrid() 함수로 그린 페어 플롯 그래프

```
1 color=['#00994C', '#FF007F']
2 pp=sns.PairGrid(data6, hue='gender', palette=color, height=3.3, aspect=1.3)
3 pp.map_diag(sns.histplot, bins=10)
4 pp.map_offdiag(sns.scatterplot)
5 pp.add_legend()
6 plt.show()
```

<실행결과>

