
python™

#1. 공공 데이터 분석 : 서울의 기온 분석(텍스트)

#2. 데이터의 시각화

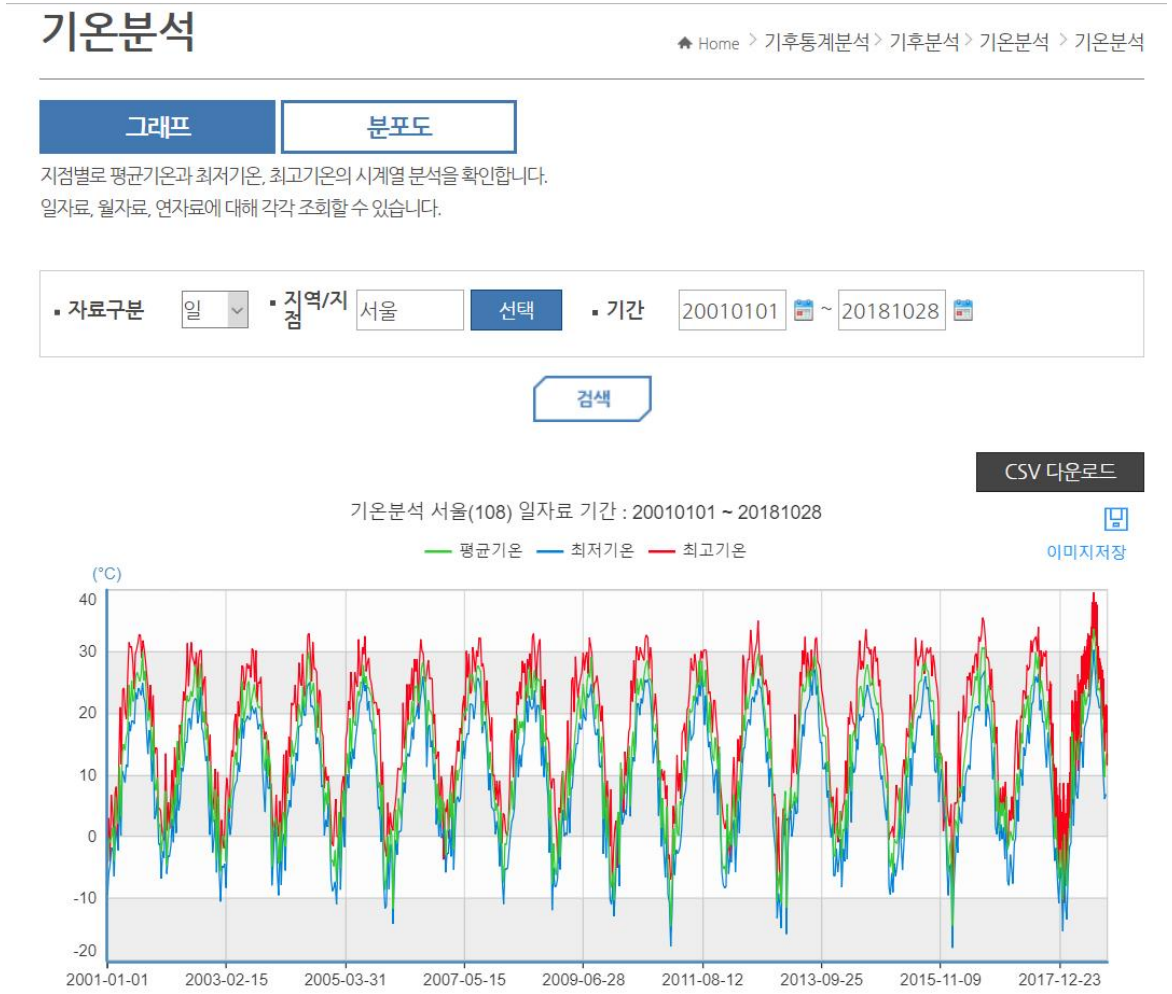
#3. 공공 데이터 분석 : 서울의 기온 분석(그래프)

#1. 공공데이터 분석

- 서울의 기온 분석(텍스트)

[공공 데이터 : 서울의 기온(텍스트)]

기상 자료 개방 포털 : https://data.kma.go.kr/stcs/grnd/grndTaList.do?pgmNo=70



[공공 데이터 : 서울의 기온(텍스트)]

2001년 01월 01일~현재 날짜(2018년 11월 22일)까지의 기온을 seoul_temp.csv로 저장

seoul_temp.csv파일을 이용하여
"서울이 가장 추웠던 날은 OOOO년 OO월 OO일 입니다 : OO.O도" 출력

[데이터 수집 : temp.csv(기상자료이용)]

	A	B	C	D	E	F
1	기온분석					
2	[검색조건]					
3	자료구분 : 일					
4	지역/지점 : 서울					
5	기간 : 20010101~20181122					
6						
7	날짜	지점	평균기온(°	최저기온(°	최고기온(℃)	
8	2001-01-01	108	-4.5	-9.5	-0.8	
9	2001-01-02	108	-4.3	-9.1	-0.2	
10	2001-01-03	108	-8.3	-10.4	-5.2	
11	2001-01-04	108	-9.8	-12.5	-6.7	
12	2001-01-05	108	-6.8	-11	-2.2	
13	2001-01-06	108	-2.9	-8.8	0.9	
14	2001-01-07	108	0.4	-0.9	1.1	
15	2001-01-08	108	2	-0.2	4.9	

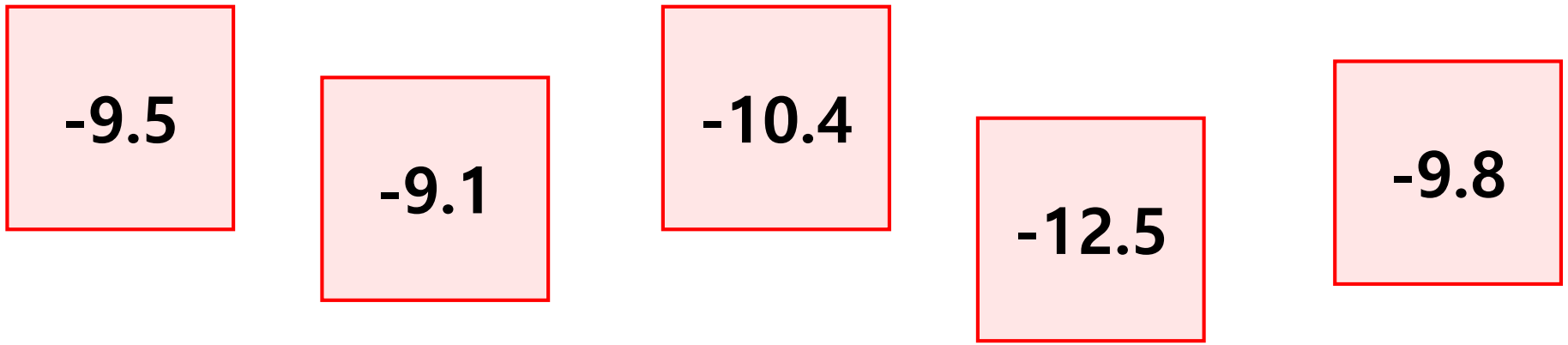
[데이터 정제 : 불필요한 자료 삭제 후 seoul_temp.csv저장]

	A	B	C	D
1	날짜	평균기온(℃	최저기온(℃	최고기온(℃)
2	2001-01-01	-4.5	-9.5	-0.8
3	2001-01-02	-4.3	-9.1	-0.2
4	2001-01-03	-8.3	-10.4	-5.2
5	2001-01-04	-9.8	-12.5	-6.7
6	2001-01-05	-6.8	-11	-2.2
7	2001-01-06	-2.9	-8.8	0.9
8	2001-01-07	0.4	-0.9	1.1
9	2001-01-08	2	-0.2	4.9
10	2001-01-09	-0.4	-6	3
11	2001-01-10	-5	-9.1	-1.5
12	2001-01-11	-8	-14.4	-2.2

[데이터 읽어 오기 : seoul_temp.csv]

```
f=open('seoul_temp.csv','r')
import csv
data=csv.reader(f)
for line in data :
    print(line)
```

[최소값을 찾는 알고리즘을 구현하여 최저기온 찾기]



[공공 데이터 : 서울의 기온(텍스트)]

2001년 01월 01일~현재 날짜(2018년 11월 22일)까지의 기온을 seoul_temp.csv로 저장

seoul_temp.csv파일을 이용하여
"서울이 가장 추웠던 날은 OOOO년 OO월 OO일 입니다 : OO.O도" 출력

서울이 가장 추웠던 날 : 2001-01-15 / -18.6도

서울이 가장 추웠던 날은 2001년 01월 15일 입니다 : -18.6도

#2. 데이터의 시각화

https://matplotlib.org/

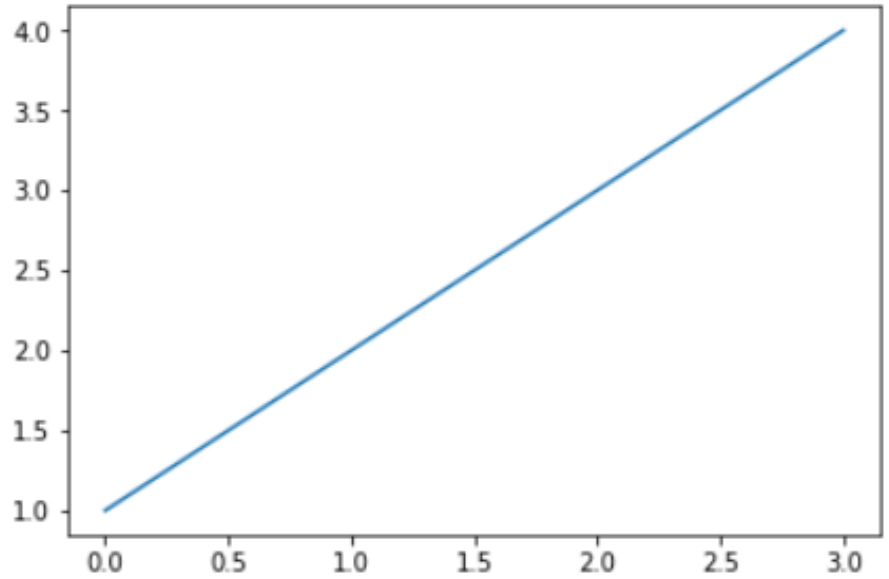
[matplotlib]

- python에서 그래프를 그리기 위한 라이브러리
- matplotlib.pyplot : matlab 스타일의 그래프를 그리기 위한 서브 라이브러리
- import matplotlib.pyplot as plt : plt(별칭)으로 matplotlib.pyplot을 임포트 함.

[선(plot) 그래프 그리기]

1. 그래프 라이브러리 임포트
2. 그래프에 그릴 데이터 결정
3. 그래프 보여주기

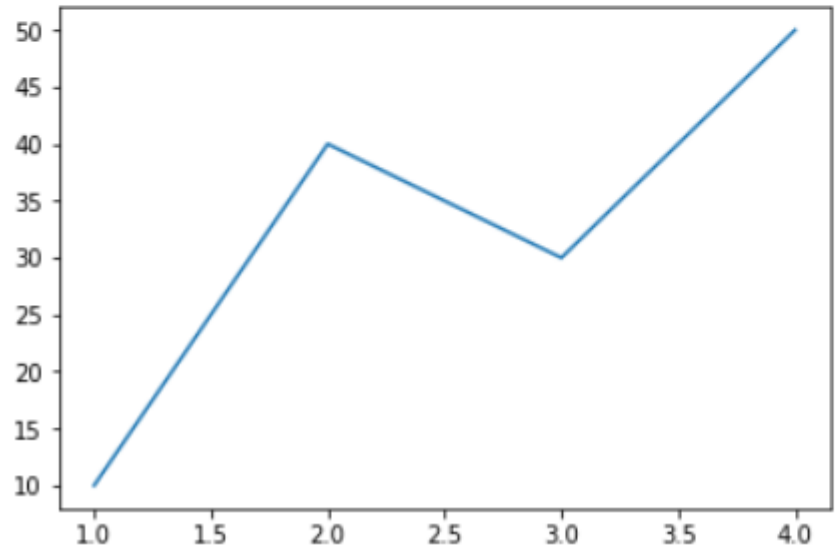
```
# 1. 그래프 라이브러리 임포트
import matplotlib.pyplot as plt

#2. 그래프에 그릴 데이터를 리스트에 넣기
data=[1,2,3,4] #y축 값을 리스트에 넣기
plt.plot(data) #data 리스트로 그래프 만들기 / plot함수는 선그래프

#3. 그래프 보여주기
plt.show()
```



[선(plot) 그래프 그리기 - x축, y축 값 설정]

1. 그래프 라이브러리 임포트
2. 그래프에 그릴 데이터 결정
    1) x축 데이터
    2) y축 데이터
3. 그래프 보여주기

```
# 1. 그래프 라이브러리 임포트
import matplotlib.pyplot as plt

#2. 그래프에 그릴 데이터를 리스트에 넣기
xdata=[1,2,3,4] #x축 값을 리스트에 넣기
ydata=[10,40,30,50] #y축 값을 리스트에 넣기

plt.plot(xdata,ydata) #data 리스트로 그래프 만들기 / (x축의 값, y축의 값)의 순서

#3. 그래프 보여주기
plt.show()
```



[그래프 꾸미기]

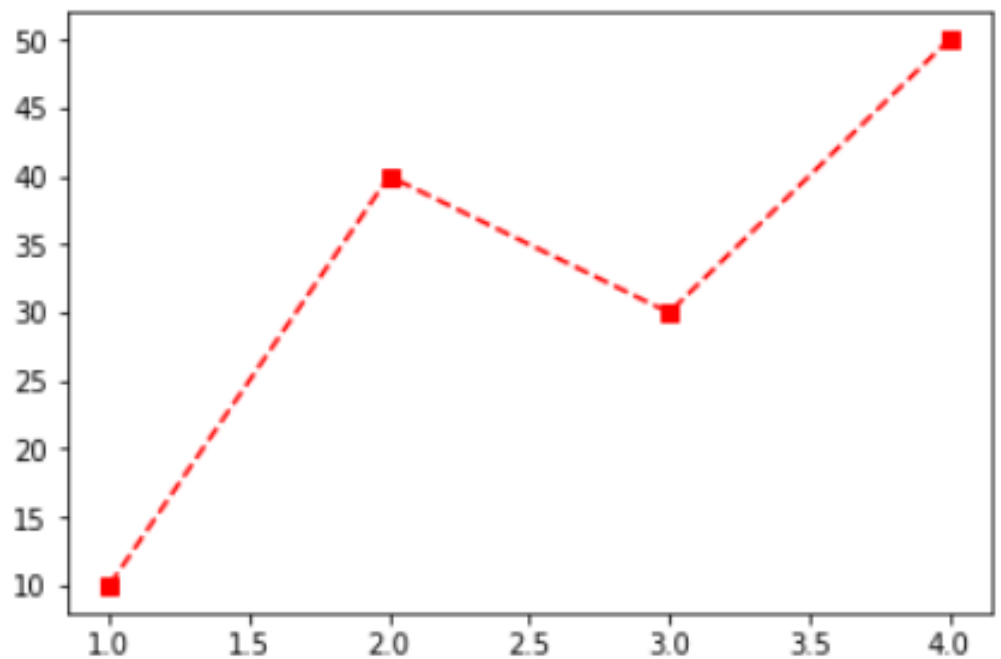
1. 색깔 : b / g / r / c / m / y / k / m
2. 마커(데이터의 위치를 나타내는 기호) : . / , / o / v / ^ / s / + / *
3. 선 스타일 : - / -- / -. / :

문자	색상
b	blue(파란색)
g	green(녹색)
r	red(빨간색)
c	cyan(청록색)
m	magenta(마젠타색)
y	yellow(노란색)
k	black(검은색)
w	white(흰색)

마커	의미
o	circle(원)
v	triangle_down(역 삼각형)
^	triangle_up(삼각형)
s	square(네모)
+	plus(플러스)
.	point(점)

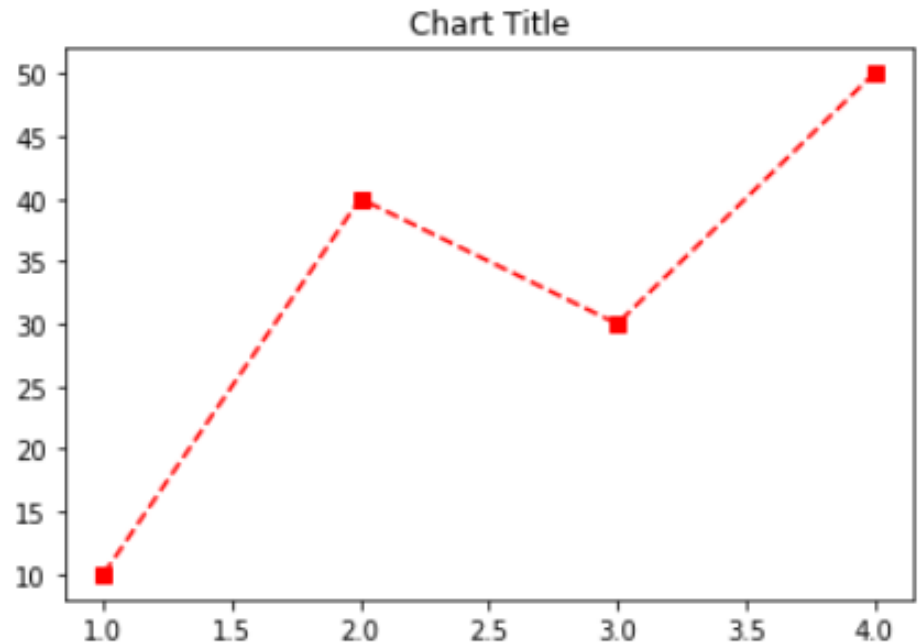
[그래프 꾸미기]

```
# 1. 그래프 라이브러리 임포트
import matplotlib.pyplot as plt

#2. 그래프 만들 며 동시에 데이터 넣기
plt.plot([1,2,3,4],[10,40,30,50],'rs--') # 그래프 꾸미기 /빨간색, 네모마커, --선스타일
#3. 그래프 보여주기
plt.show()
```



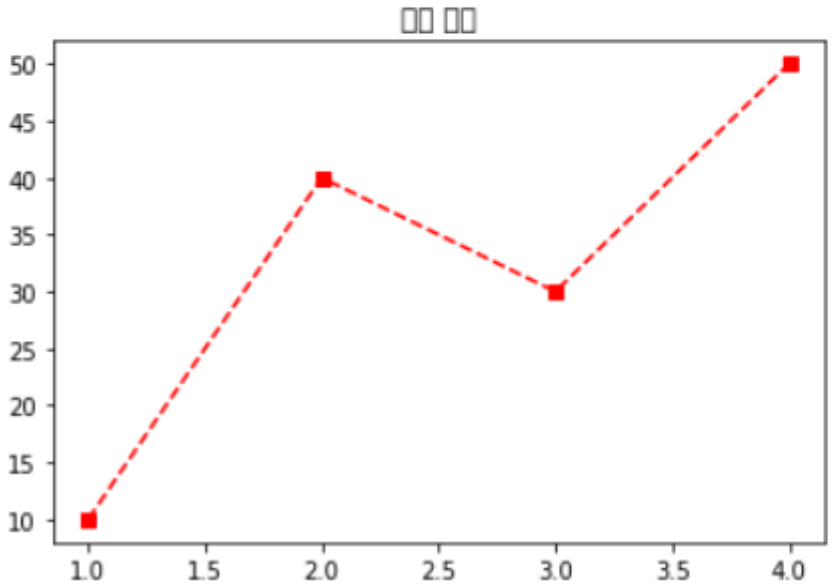
[그래프 제목 넣기]

```
# 1. 그래프 라이브러리 임포트
import matplotlib.pyplot as plt

#2. 그래프 만들 며 동시에 데이터 넣기
plt.plot([1,2,3,4],[10,40,30,50],'rs--') # 그래프 꾸미기 /빨간색, 네모마커, --선스타일
plt.title('Chart Title') #그래프 제목 넣기
#3. 그래프 보여주기
plt.show()
```

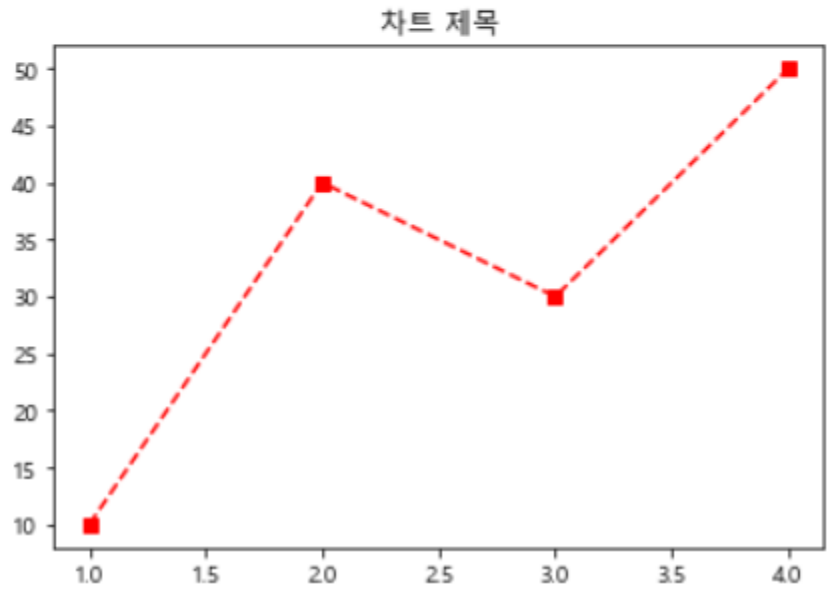


[그래프 제목 넣기 - 한글 사용]

```python
# 1. 그래프 라이브러리 임포트
import matplotlib.pyplot as plt

#2. 그래프 만들 며 동시에 데이터 넣기
plt.plot([1,2,3,4],[10,40,30,50],'rs--') # 그래프 꾸미기 /빨간색,

plt.title('차트 제목') #차트 제목을 한글로 넣기
#3. 그래프 보여주기
plt.show()
```



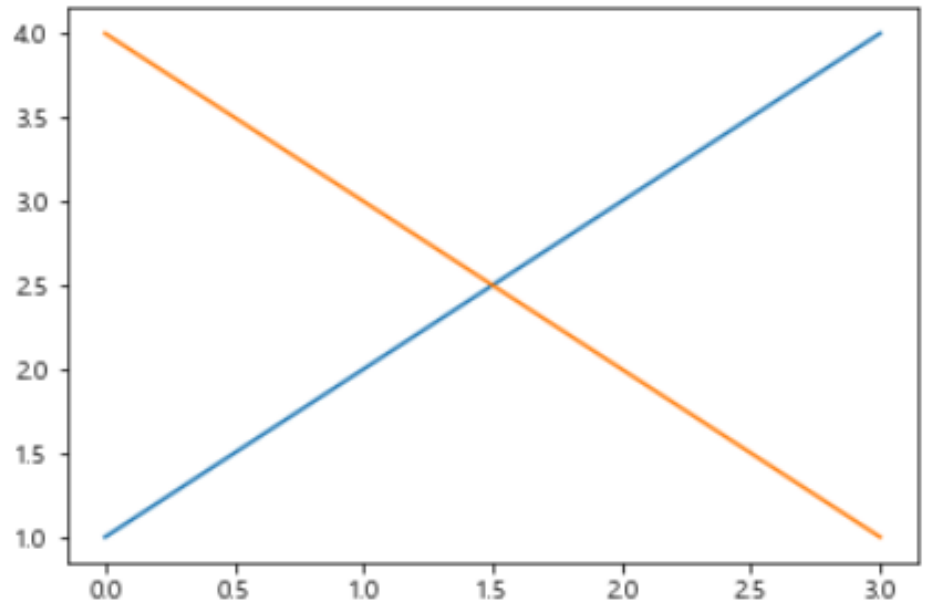

```python
# 1. 그래프 라이브러리 임포트
import matplotlib.pyplot as plt

plt.rc('font',family='Malgun Gothic')
#2. 그래프 만들 며 동시에 데이터 넣기
plt.plot([1,2,3,4],[10,40,30,50],'rs--') # 그래프 꾸미기 /빨간색, 네모마커, --

plt.title('차트 제목') #차트 제목을 한글로 넣기
#3. 그래프 보여주기
plt.show()
```



[여러 개의 선 그래프]

- 여러 개의 plot 함수 사용으로 하나의 그림에 겹쳐서 그릴 수 있다

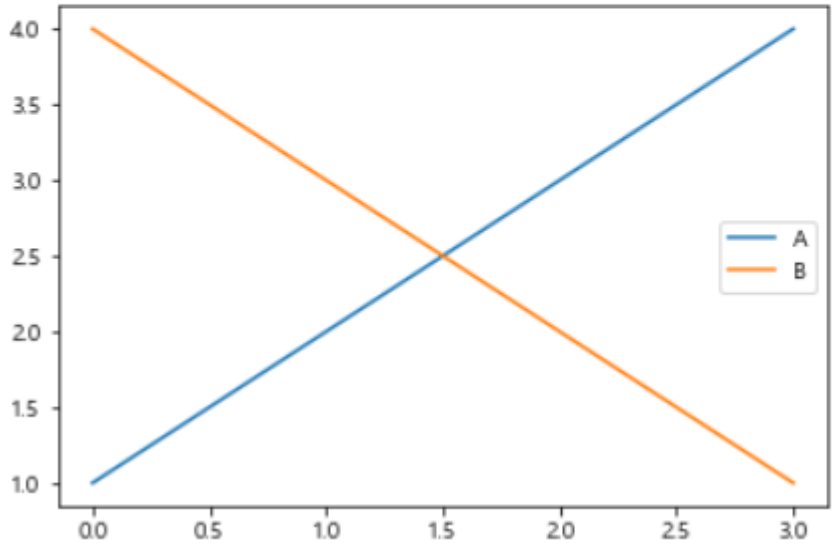
```
import matplotlib.pyplot as plt
plt.plot([1,2,3,4])
plt.plot([4,3,2,1])

plt.show()
```



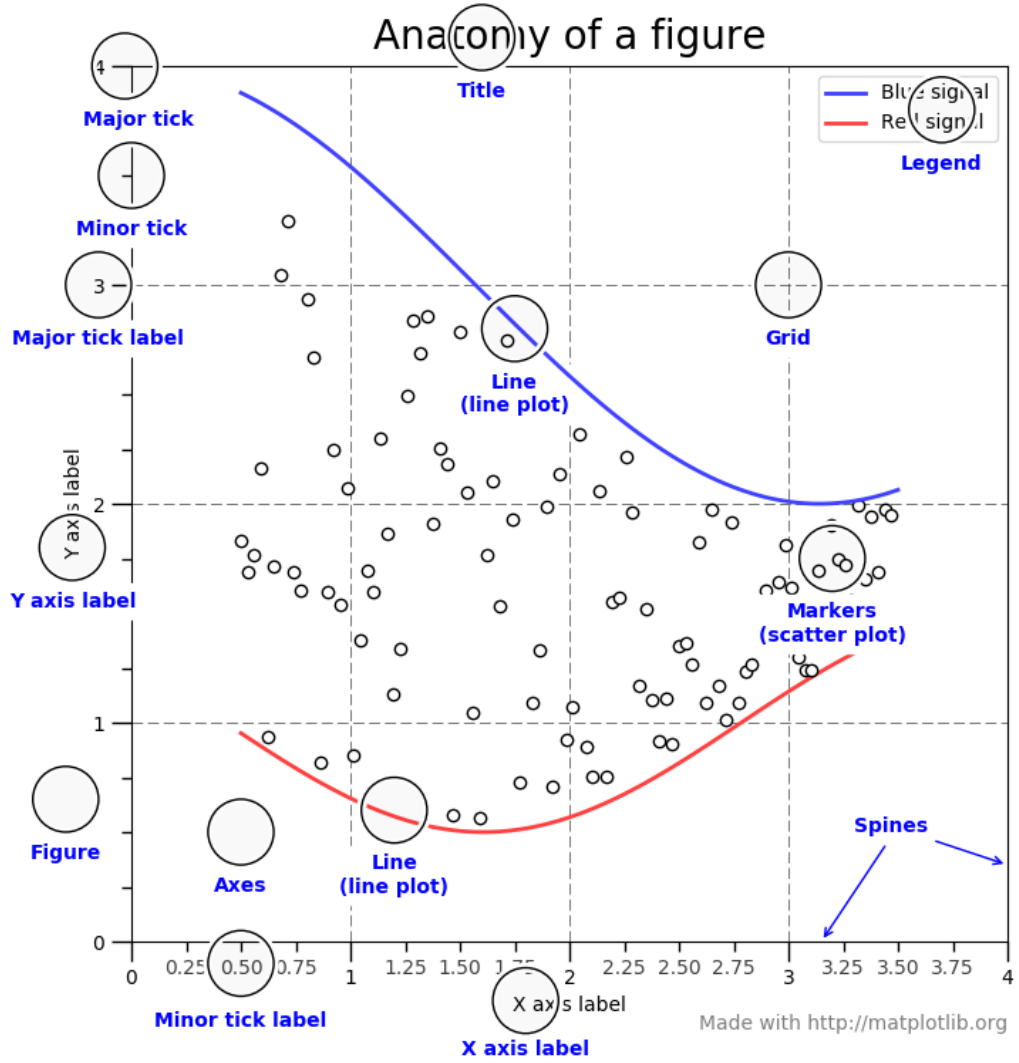
[여러 개의 선 그래프- 범례 표시하기]

- 레이블(label)을 설정한 후 범례(legend)를 그림

```
import matplotlib.pyplot as plt
plt.plot([1,2,3,4] , label='A') # 레이블 설정
plt.plot([4,3,2,1], label='B')
plt.legend() # 설정한 레이블을 이용하여 범례 그래프에 표시하기

plt.show()
```



[참고-그래프의 각 부분의 명칭]



[참고-그래프의 각 부분의 명칭]

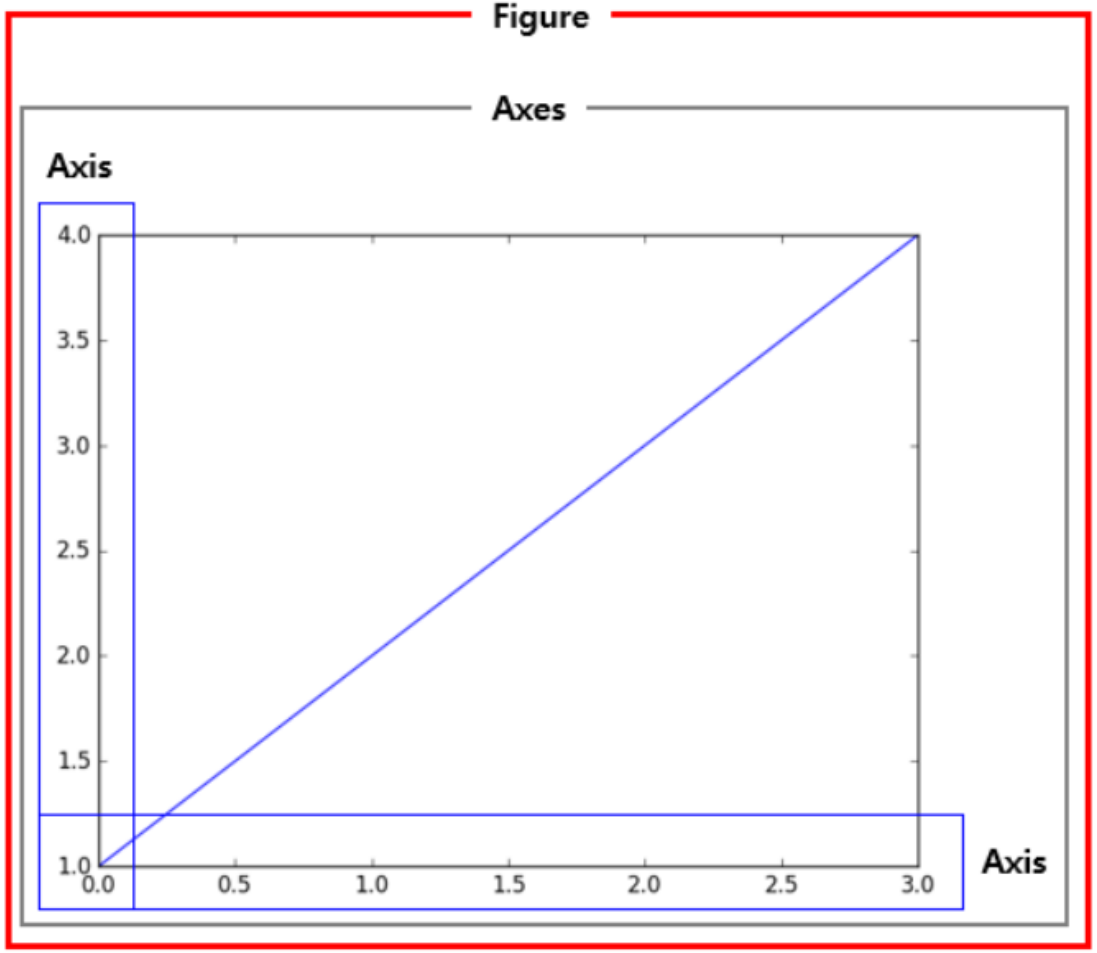
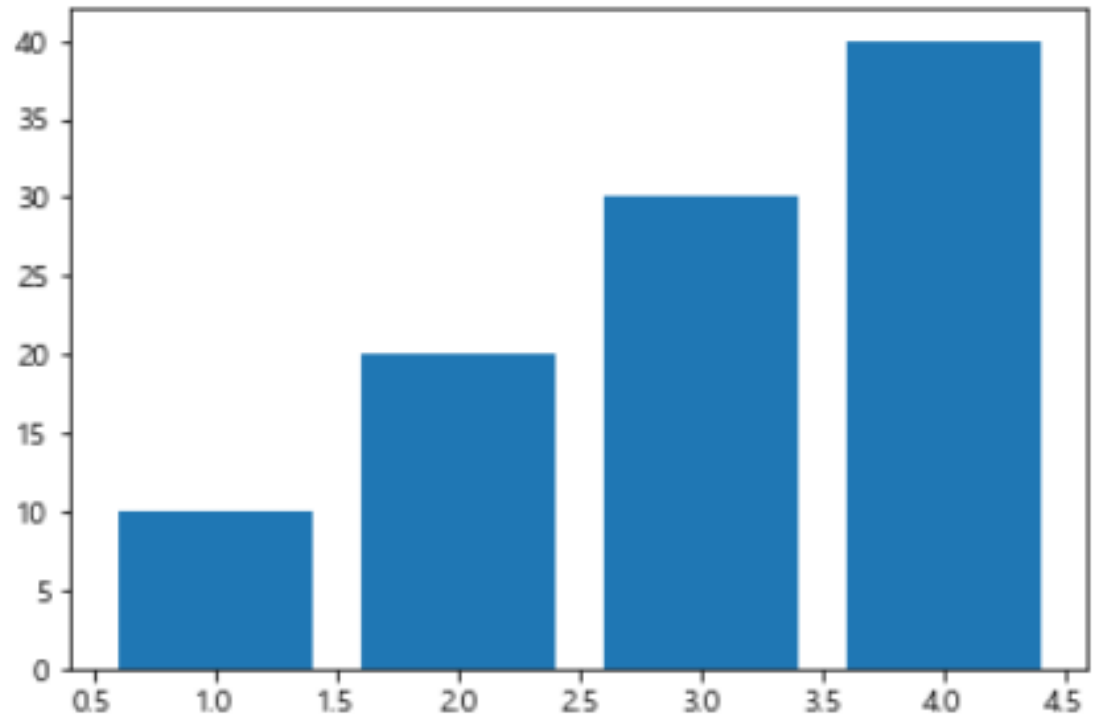


그림 15.10 Figure의 구성

https://wikidocs.net/4763

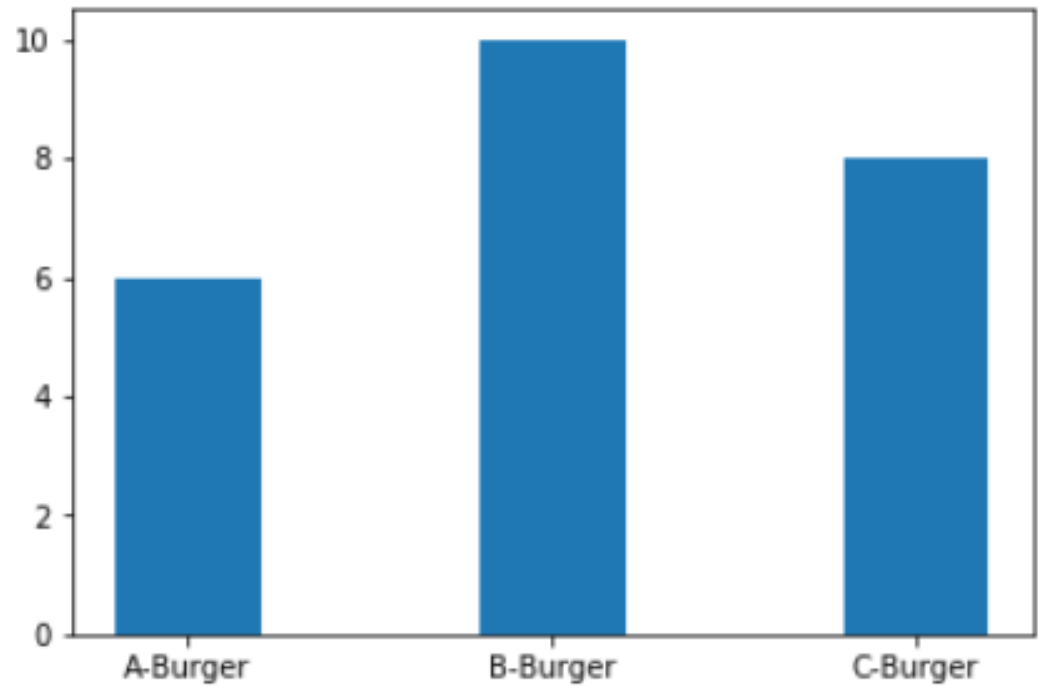
[막대(bar) 그래프 그리기 : bar(수직방향) / barh(수평방향)]

```
import matplotlib.pyplot as plt
plt.bar([1,2,3,4],[10,20,30,40])
plt.show()
```



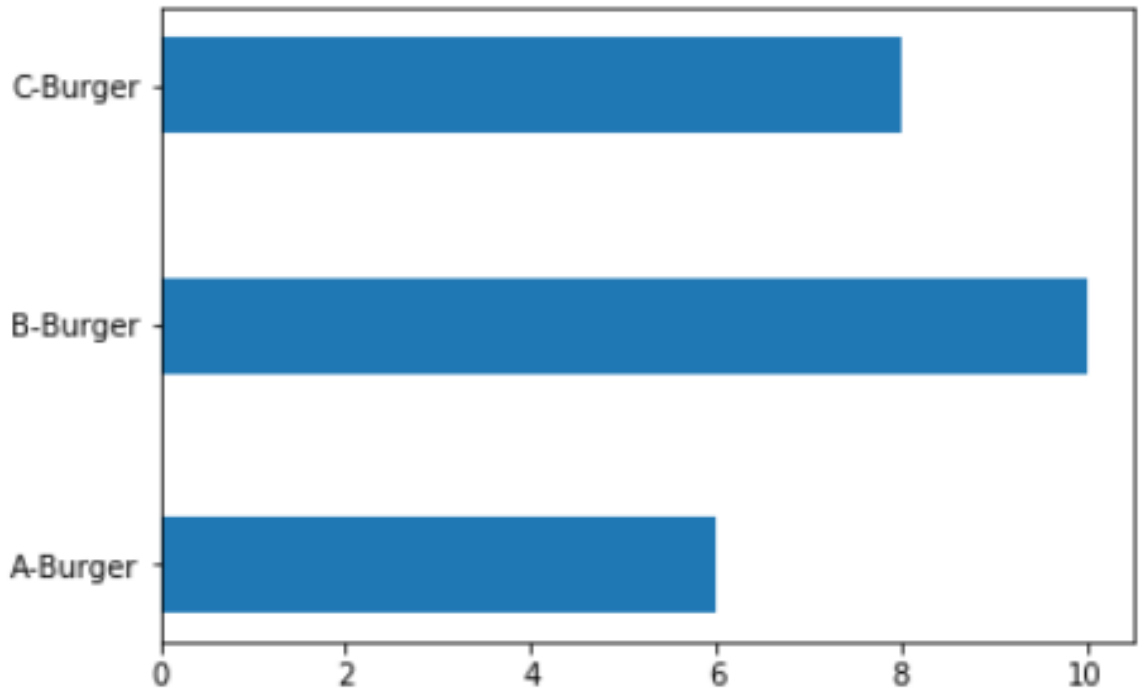
[막대(bar) 그래프 그리기 : bar(수직방향) / barh(수평방향)]

```
import matplotlib.pyplot as plt
brand=['A-Burger','B-Burger','C-Burger']
burger=[6,10,8]
plt.bar(range(3),burger, width=0.4)
plt.xticks(range(3), brand)
plt.show()
```



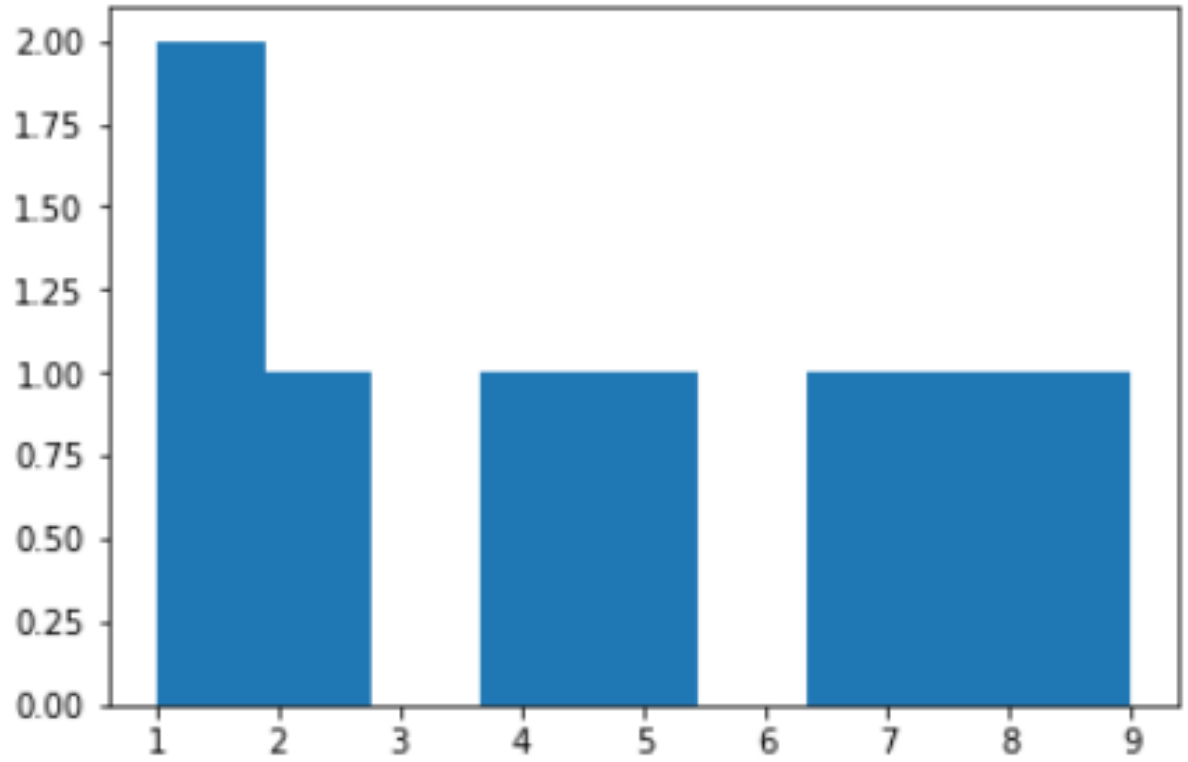
[막대(bar) 그래프 그리기 : bar(수직방향) / barh(수평방향)]

```
import matplotlib.pyplot as plt
brand=['A-Burger','B-Burger','C-Burger']
burger=[6,10,8]
plt.barh(range(3),burger, height=0.4)
plt.yticks(range(3), brand)
plt.show()
```



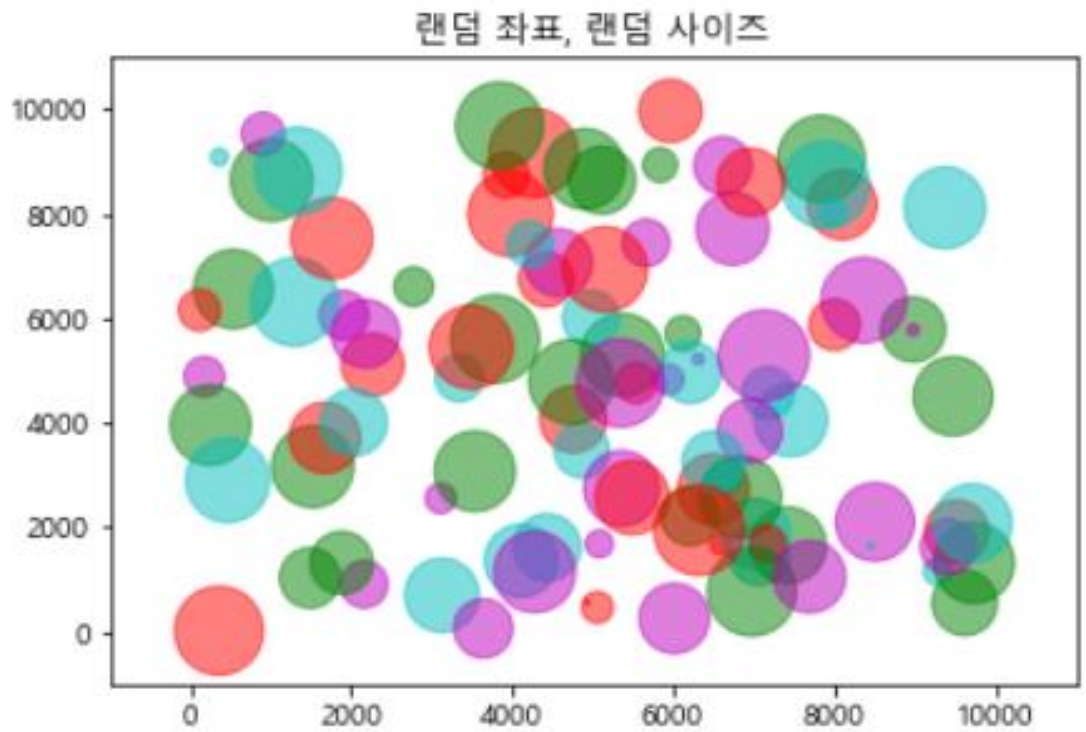
[히스토그램(histogram)]

```
import matplotlib.pyplot as plt

plt.hist([1,1,2,4,5,7,8,9], bins=9)
plt.show()
```



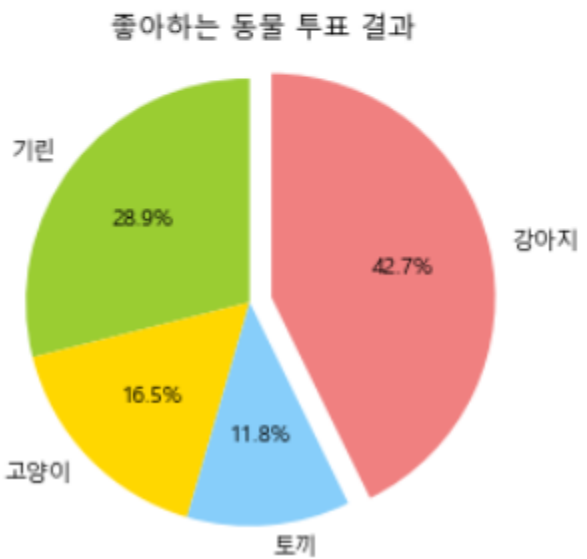
[산포도(scatter)]

```
import matplotlib.pyplot as plt
import random

x=[]
y=[]
z=[]

plt.rc('font',family='Malgun Gothic')
plt.title('랜덤 좌표, 랜덤 사이즈')

for i in range(100) :
    x.append(random.randint(1,10000))
    y.append(random.randint(1,10000))
    z.append(random.randint(1,1000))

plt.axis([-1000,11000, -1000, 11000])
plt.scatter(x,y,s=z,c=['r','g','m','c'],alpha=0.5)
plt.show()
```



[원 그래프(pie)]

```
import matplotlib.pyplot as plt

size=[231,132,94,341]
explode=(0,0,0,0.1)
labels='기린','고양이','토끼','강아지'
colors=['yellowgreen','gold','lightskyblue','lightcoral']

plt.title('좋아하는 동물 투표 결과')
plt.pie(size, explode=explode, labels=labels, colors=colors, autopct='%1.1f%%',startangle=90)

plt.axis('equal')
plt.show()
```



중앙일보

[팩트체크] 폭염 극심했던 여름 뒤엔 매서운 겨울이 온다고?

강찬수 입력 2018.11.01. 06:01 수정 2018.11.01. 08:58 댓글 57개

올 가을들어 가장 추운날씨를 보인 30일 오후 서울 명동거리에서 시민들이 두꺼운 외투를 입고 길을 거닐고 있다. [뉴스1]

지난달 30일 서울의 아침 기온은 영상 0.7도. 평년보다 6도 이상 낮았다. 차가운 땅 위에 놓인 그릇에는 얼음도 얼었다.

111년 기상 관측 이래 가장 더웠던 지난여름을 겪고 난 시민들은 올겨울 매서운 추위가 닥치지 않을까 걱정이다.

"유난히 더운 여름 뒤에는 매서운 겨울이 온다"는 속설 때문이다. 과연 이 속설은 사실일까.

#3. 공공데이터 분석

- 서울의 기온 분석(그래프)

[데이터 수집 : temp.csv(기상자료이용)]

[데이터 정제 : 불필요한 자료 삭제 후 seoul_temp.csv저장]

[데이터 읽어 오기 : seoul_temp.csv]

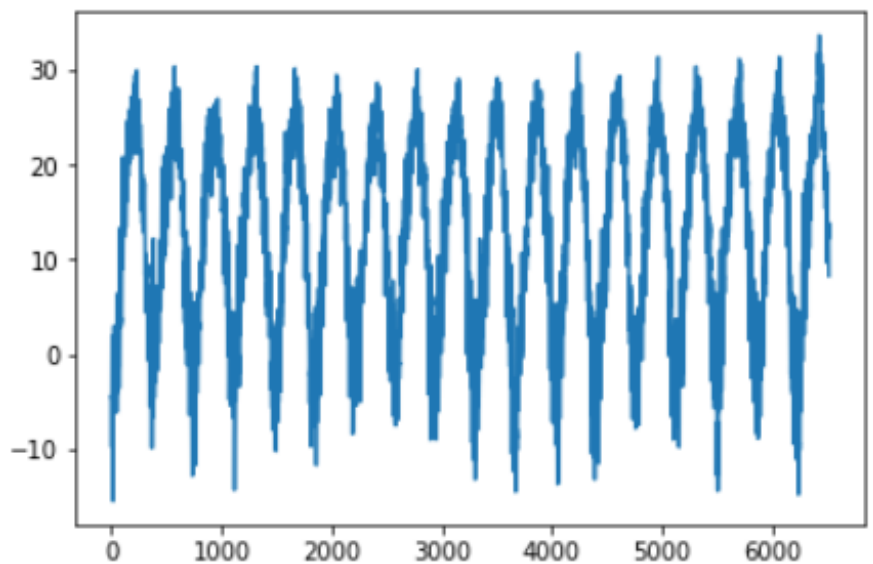
[데이터 시각화 하기 : 평균 기온 시각화 하기]

[데이터 시각화 하기 : 평균 기온 시각화 하기]

	A	B	C	D
1	날짜	평균기온(℃	최저기온(℃	최고기온(℃)
2	2001-01-01	-4.5	-9.5	-0.8
3	2001-01-02	-4.3	-9.1	-0.2
4	2001-01-03	-8.3	-10.4	-5.2
5	2001-01-04	-9.8	-12.5	-6.7
6	2001-01-05	-6.8	-11	-2.2
7	2001-01-06	-2.9	-8.8	0.9
8	2001-01-07	0.4	-0.9	1.1
9	2001-01-08	2	-0.2	4.9
10	2001-01-09	-0.4	-6	3
11	2001-01-10	-5	-9.1	-1.5

[데이터 시각화 하기 : 평균 기온 시각화 하기]

```
import matplotlib.pyplot as plt #그래프를 그리기 위해 임포트
import csv #csv파일을 읽기 위해 임포트

f=open("temp.csv")
data=csv.reader(f)
result=[]
for row in data :
    if row[-3]!='' :
        result.append(float(row[-3]))

plt.plot(result)
plt.show()
```



[(추가)데이터 시각화 하기]

여름(6월,7월,8월)의 평균 기온 출력
또는
겨울(12월,01월,02월)의 평균 기온 출력

[7월의 평균 기온을 시각화한 결과]

```
f=open('seoul_temp.csv','r')
import csv
import matplotlib.pyplot as plt

data=csv.reader(f)
header=next(data)
summer=[]
for line in data :
    month=line[0].split('-')[1]
    if(line[1]!='' and month =='07') :
        summer.append(float(line[1]))
plt.style.use('ggplot')
plt.hist(summer, bins=100)
plt.show()
```