Chapter03 자료의 정리

1. 자료의 종류

- **질적자료: 숫자에 의해 표현되지 않는 자료** ex: 혈액형, 만족도, 학년별

- **양적자료: 숫자로 표현되고, 그 숫자에 의미가 부여되는 자료** ex: 스팸문자 횟수, 몸무게, 키

1. 이산자료: 셈을 할 수 있는 자료 (스팸문자 횟수)
2. 연속자료: 어떤 구간 안에서 측정되는 자료 (몸무게 , 키)

자료를 표현하는 방법

[실습] [예제 3-2]

A.표 만들기 (pandas)

In [2]:

```
1 import pandas as pd
```

방법1: values(context), columns, index 지정해서 만들기

In [28]:

```
1 columns = list(range(2005,2015,1))
2 values  = [[15,7,2,10,8,5,14,9,18,8]]
3 index   = ['횟수']
4
5 df = pd.DataFrame(values, columns=columns, index=index)
6
7 df.name = '강도 3.0인 지진 횟수'
8 df.columns.name = '연도'
9 df
```

Out[28]:

연도	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014
횟수	15	7	2	10	8	5	14	9	18	8

방법2: dictionary 이용

In [30]:

```
df.columns
```

Out[30]:

```
Int64Index([2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013,
2014], dtype='int64', name='연도')
```

In [222]:

```
data = {2005:15,2006:7,2007:2,2008:10,2009:8,
        2010:5,2011:14,2012:9,2013:18,2014:8}

df = pd.DataFrame(data, index=['횟수'])

df.columns.name = '연도'
df
```

Out[222]:

연도	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014
횟수	15	7	2	10	8	5	14	9	18	8

DataFrame의 컬럼 목록 추출

In [37]:

```
list(df.columns)
```

Out[37]:

```
[2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014]
```

DataFrame의 values 목록 추출

In [38]:

```
list(df.values[0])
```

Out[38]:

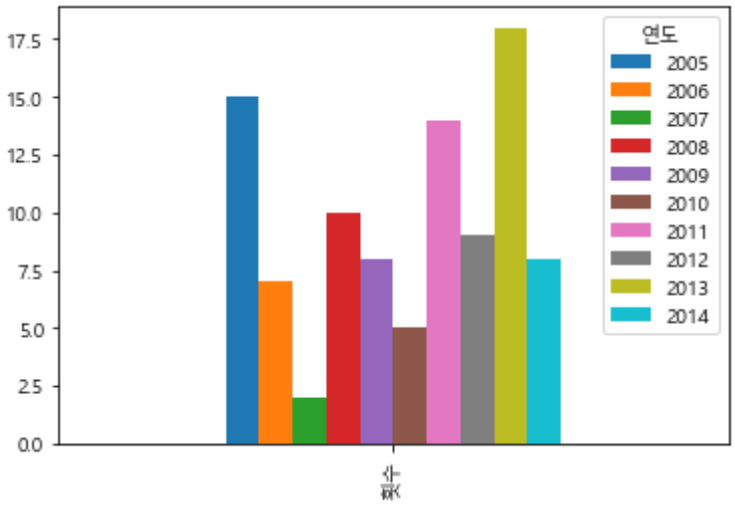
```
[15, 7, 2, 10, 8, 5, 14, 9, 18, 8]
```

In [224]:

```
1 df.plot(kind='bar')
```

Out[224]:

<AxesSubplot:>



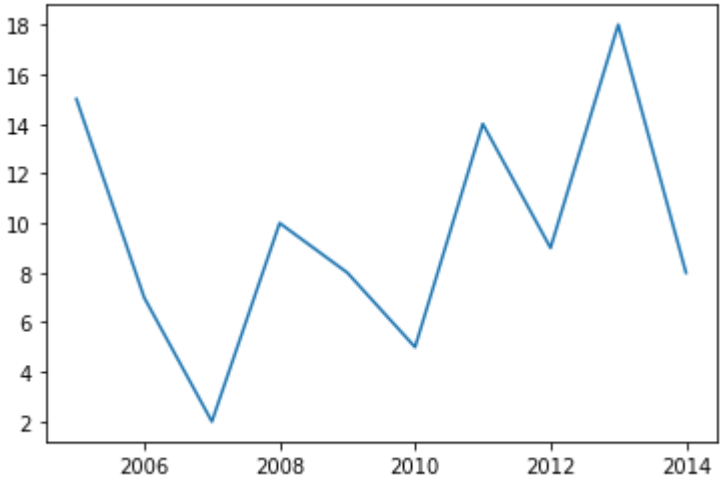
B.그래프 그리기

1.선 그래프

In [42]:

```
import matplotlib.pyplot as plt

x = list(df.columns)   #x = df.columns
y = list(df.values[0]) #y = df.values[0]
print(f'x축: {x}')
print(f'y축: {y}')

plt.plot(x, y)
plt.show()
```

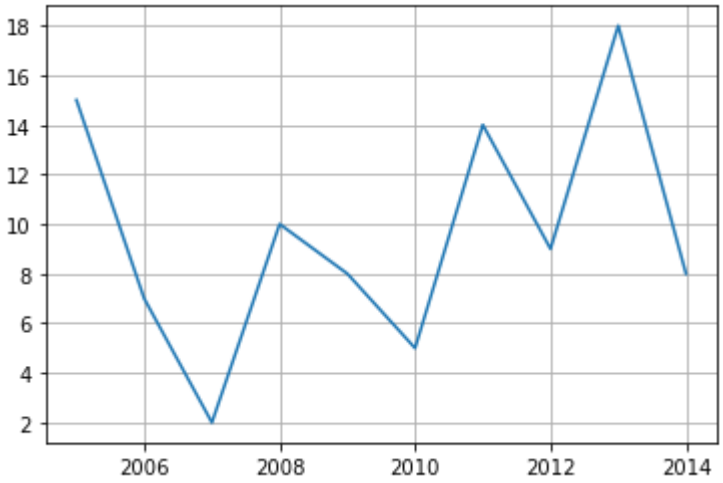
```
x축: [2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014]
y축: [15, 7, 2, 10, 8, 5, 14, 9, 18, 8]
```



그래프에 그리드 표시

In [9]:

```
plt.plot(x, y)
plt.grid()   # 그리드 표시, plt.grid(True)
```



2개의 선그래프와 축 범위 지정

In [15]:

```
# 여러 개의 그래프 표시
x1 = [i+1 for i in x]
y1 = [i+1 for i in y]

plt.plot(x, y, '-', x1, y1, '-.')
plt.xlim(2004, 2017)   #x축의 범위
plt.ylim(-10, 25)      #y축의 범위
plt.show()

```

한글 표현

설치된 폰트 확인

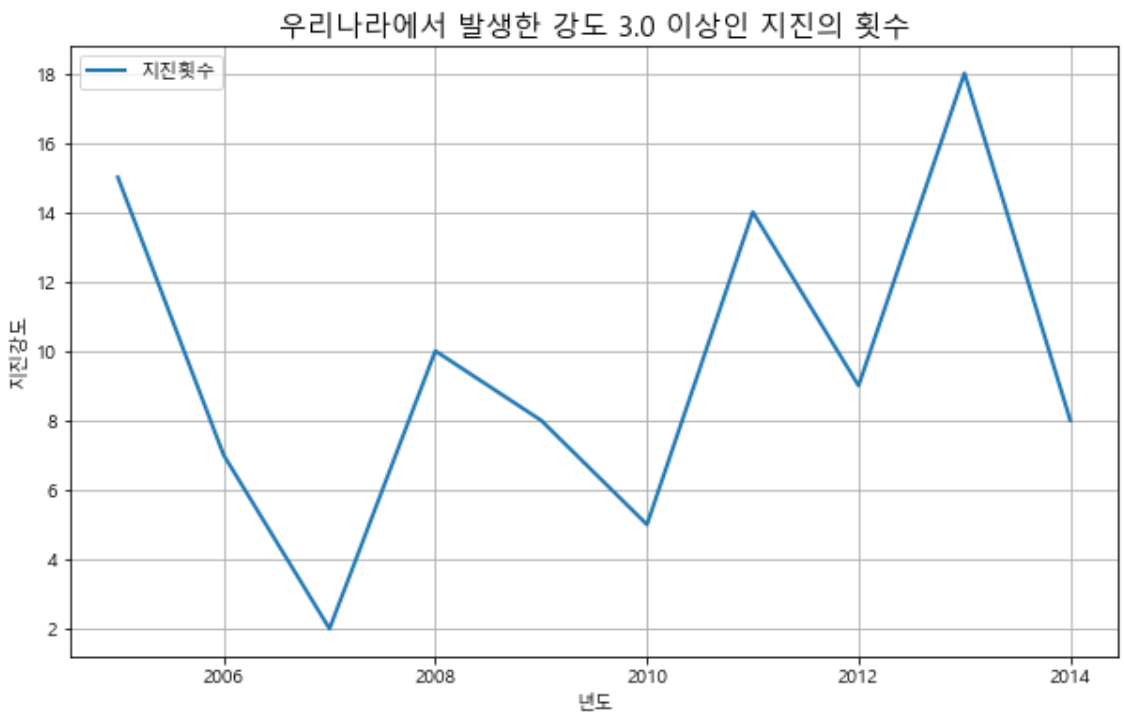
In [164]:

```
import matplotlib.font_manager

for f in matplotlib.font_manager.fontManager.ttflist:
    if f.name.startswith('Malgun'):
        print(f.name)
```

```
Malgun Gothic
Malgun Gothic
Malgun Gothic
```

In [87]:

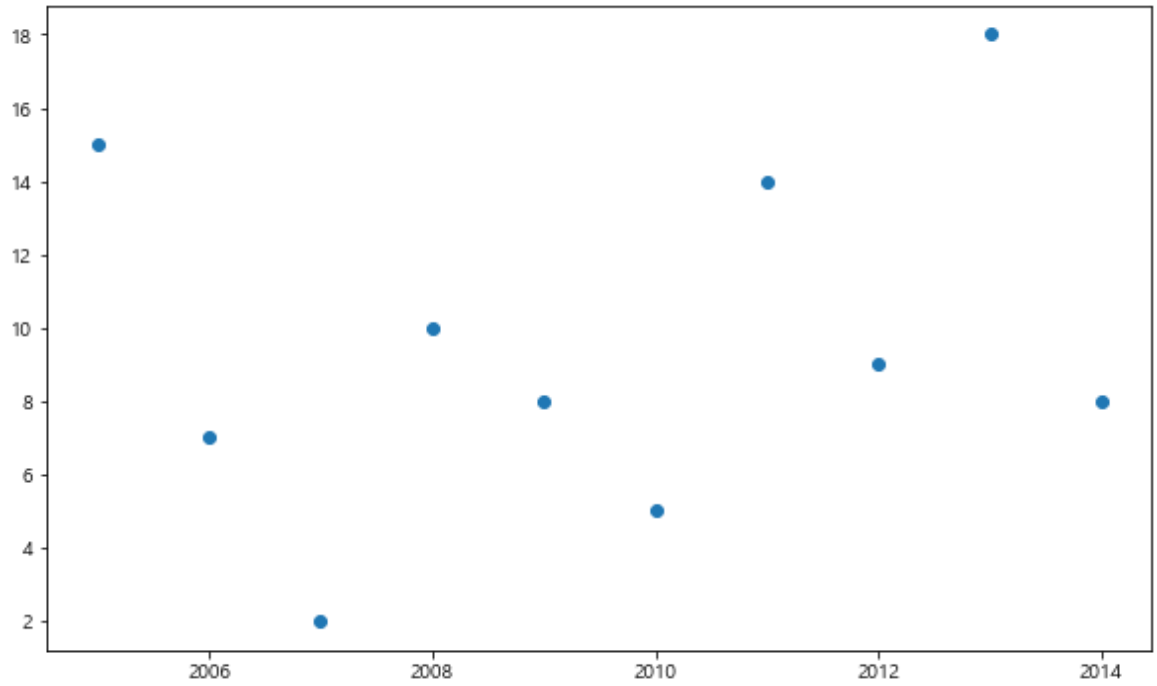
```
import matplotlib.pyplot as plt

# 한글출력 설정
plt.rcParams['font.family'] = 'Malgun Gothic'# '맑은 고딕'으로 설정
# 그래프 크기 지정
plt.rcParams['figure.figsize'] = (10, 6)
# 선 굵기 지정
plt.rcParams['lines.linewidth'] = 2
#matplotlib.rcParams['axes.unicode_minus'] = False

# 그래프 제목, 레이블, 범례,
plt.plot(x, y)
plt.xlabel('년도')       # x축 레이블
plt.ylabel('지진강도')  # y축 레이블
plt.legend(['지진횟수']) #범례,기본 위치 : loc='upper left'
plt.title('우리나라에서 발생한 강도 3.0 이상인 지진의 횟수', size=15)
plt.grid()              # 격자 표시
plt.show()
```



2.점그래프

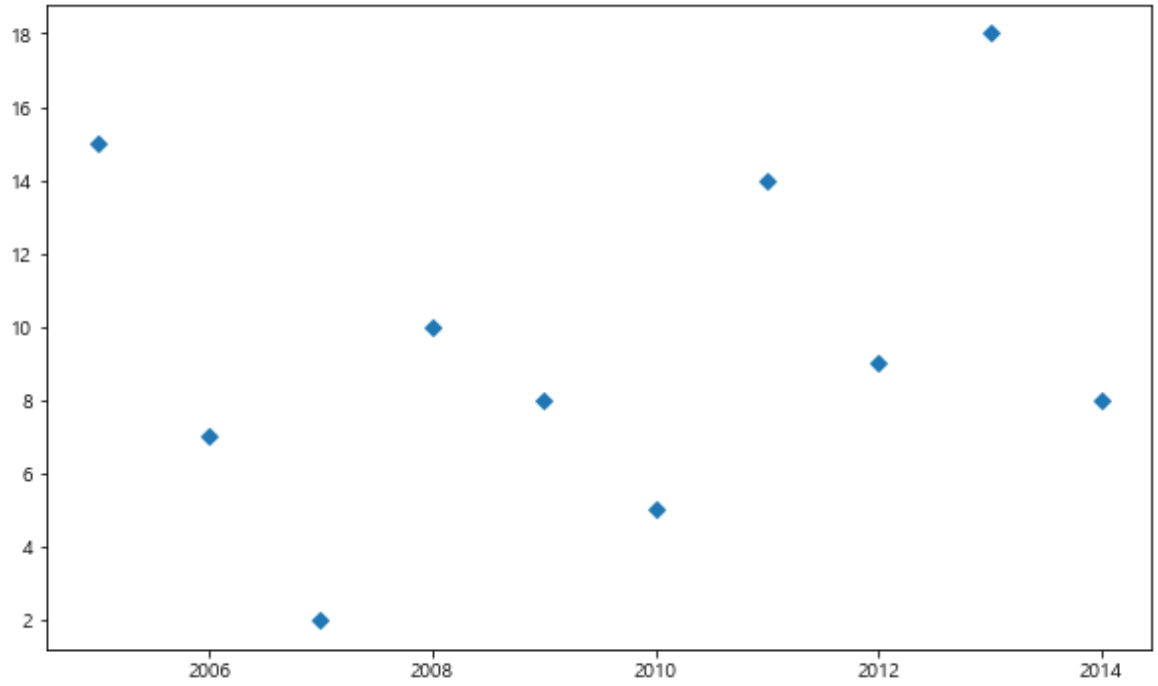
In [72]:

```
1 plt.scatter(x, y)
2 plt.show()
```



In [89]:

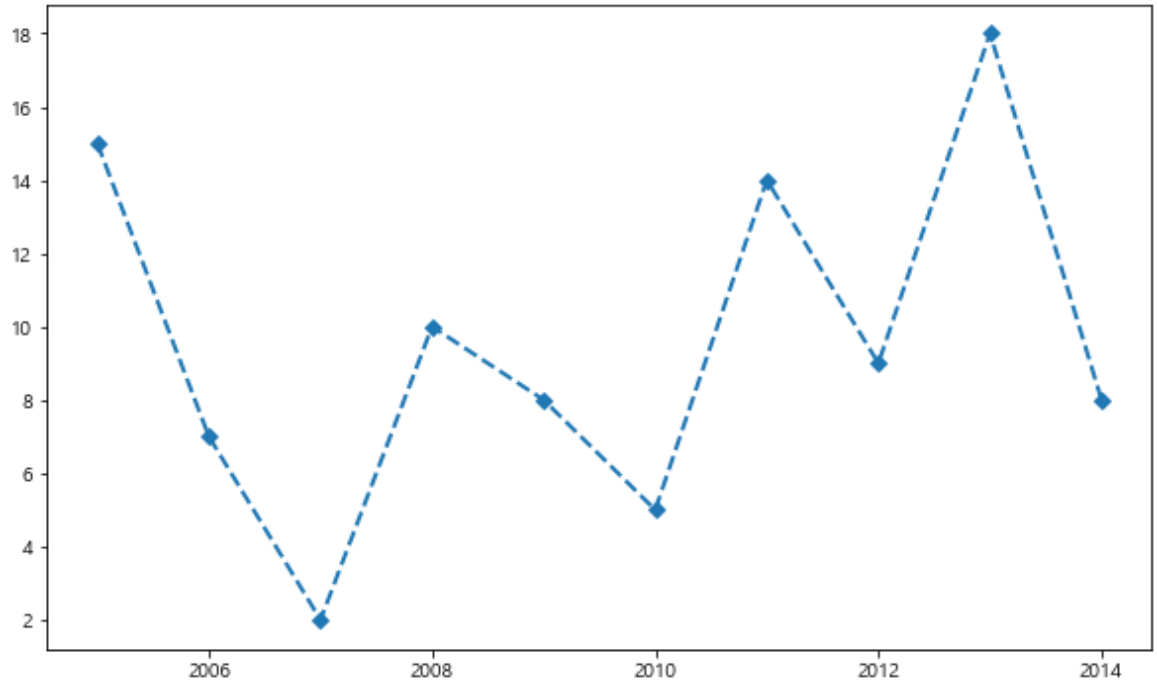
```
# 선그래프에서 마커 표시
# https://matplotlib.org/stable/api/markers_api.html?highlight=marker#module

plt.plot(x, y,'D')
plt.show()
```



3.선+marker 그래프

In [73]:

```
# https://matplotlib.org/stable/gallery/lines_bars_and_markers/linestyles.

plt.plot(x, y, 'D', linestyle='--')
plt.show()
```

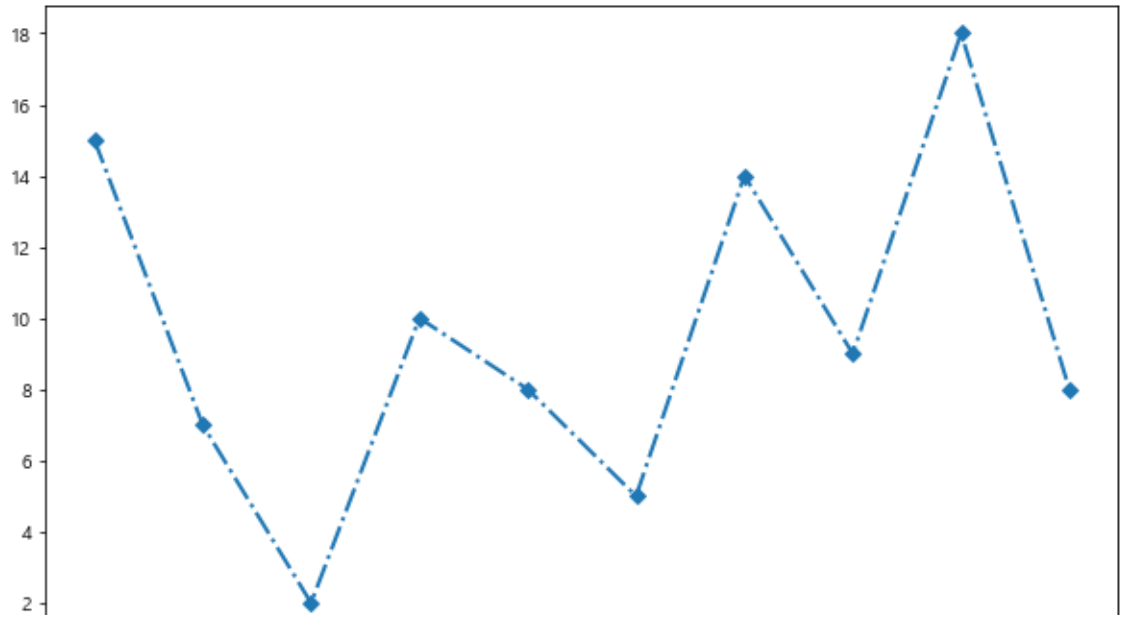


In [74]:

```
plt.plot(x, y,'D', linestyle='dashdot')
```

Out[74]:

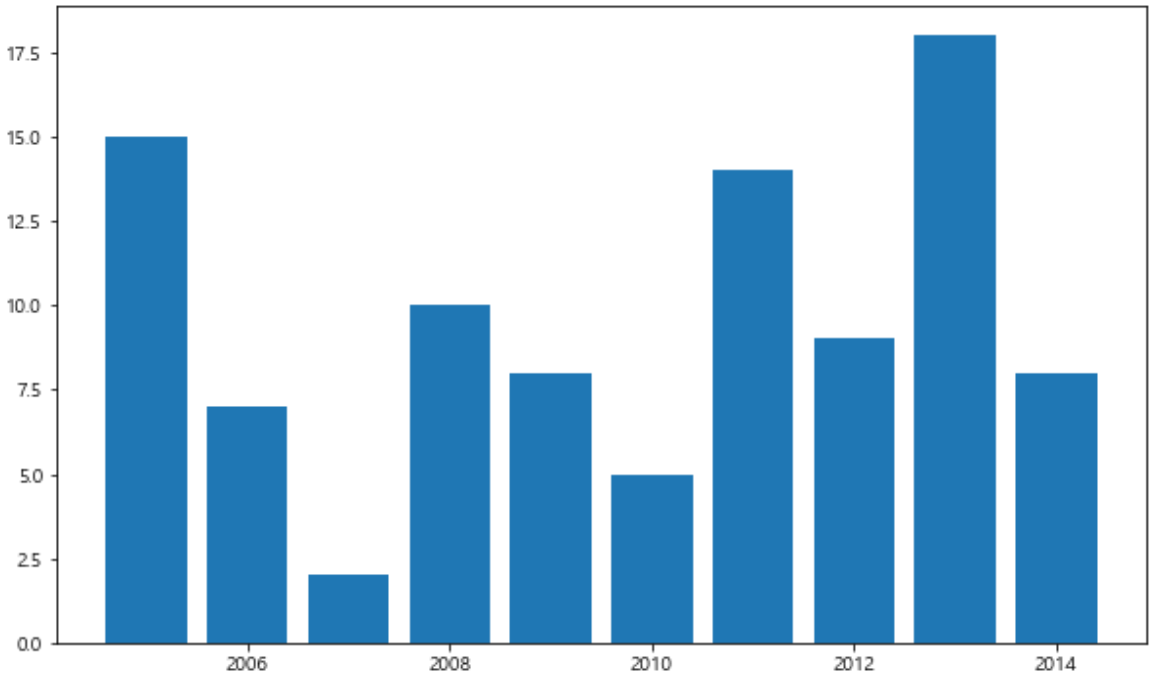
```
[<matplotlib.lines.Line2D at 0x23f328795b0>]
```



4.막대 그래프

In [75]:

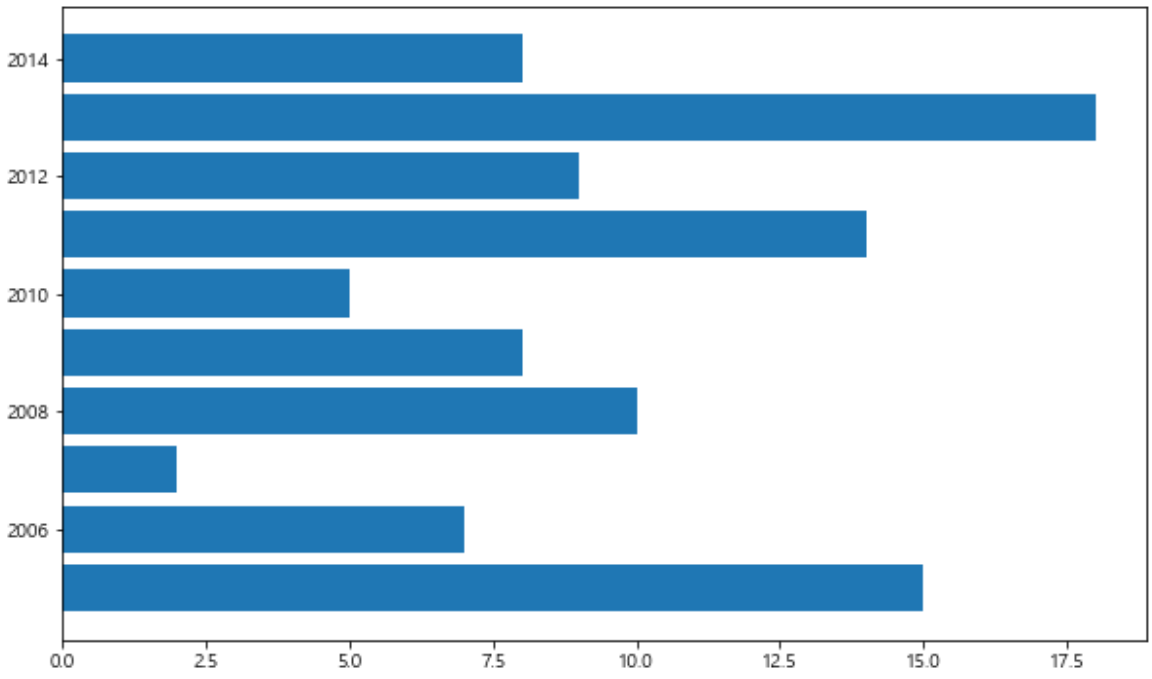
```
1 plt.bar(x,y)
2 plt.show()
```



가로 막대그래프

In [76]:

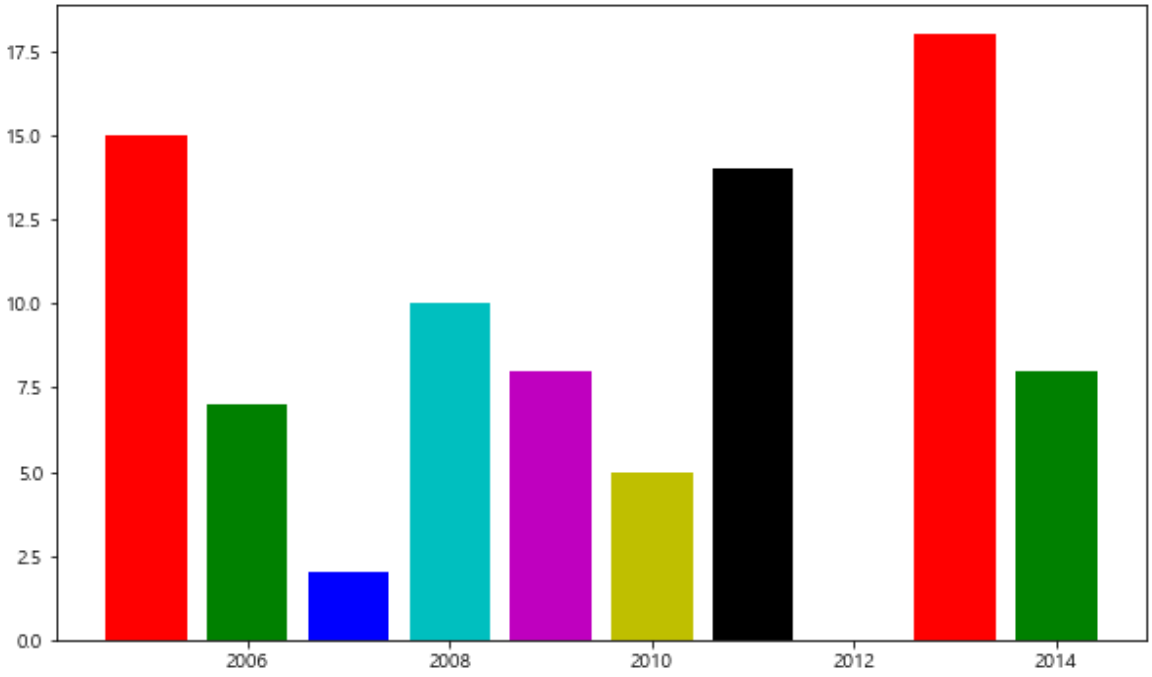
```
1 #가로 막대그래프
2 plt.barh(x,y)
3 plt.show()
```



그래프 색상 지정

In [77]:

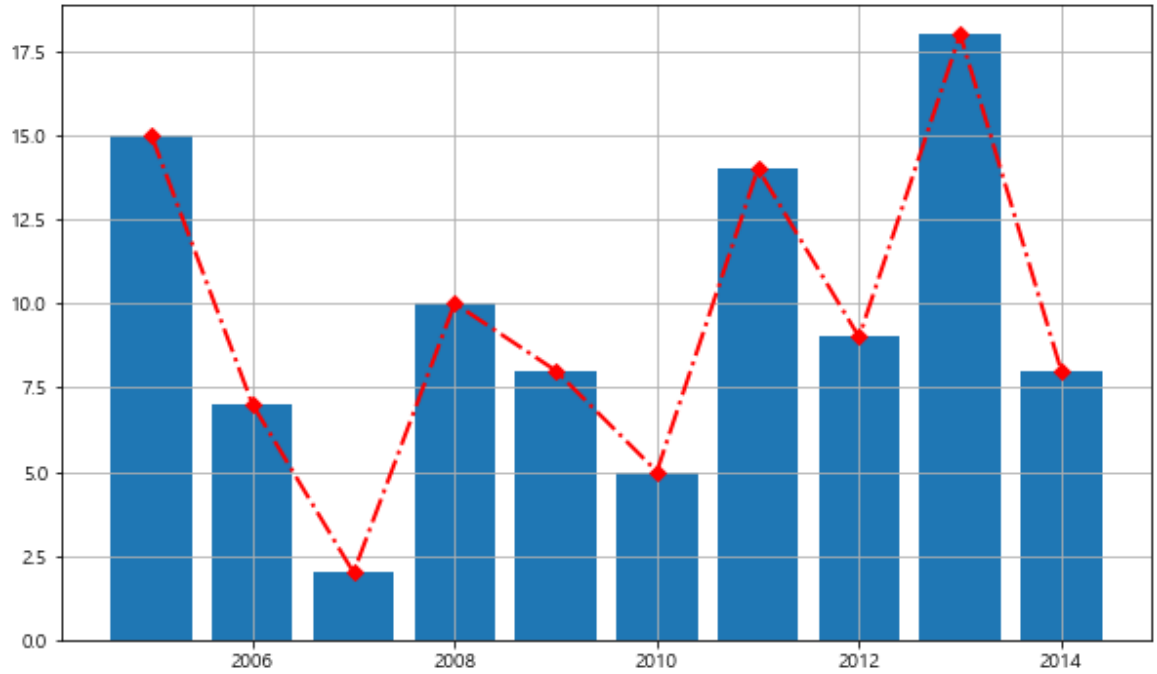
```
#그래프 색상 지정
colors = ['r','g','b','c','m','y','k','w']  # 기본색상  # Hexa코드 or CSS컬
plt.bar(x,y,color=colors)
plt.show()
```



막대그래프 + 선그래프

In [78]:

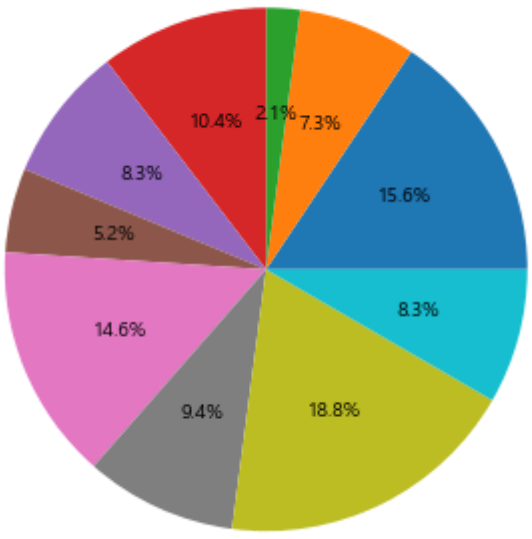
```
#막대 그래프 + 선그래프
plt.bar(x,y)
plt.plot(x, y,'D', linestyle='dashdot', color='r')
plt.grid()
```



5. 원(파이) 그래프

In [98]:

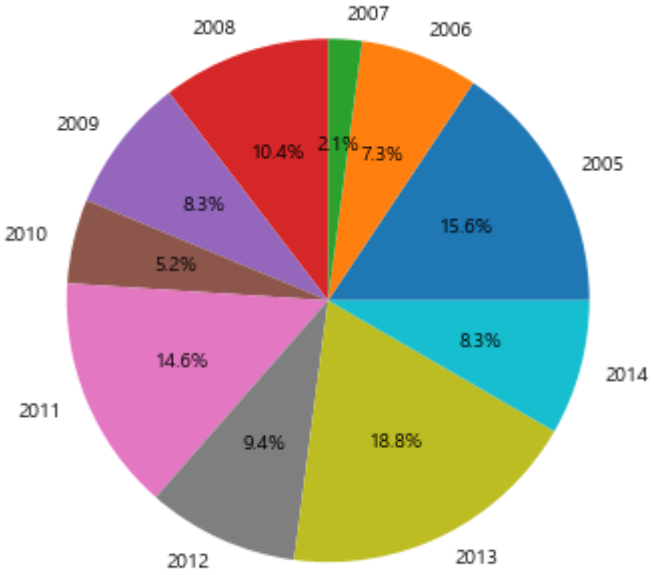
```
plt.pie(y, autopct='%.1f%%')
plt.show()
```



레이블 보여주기

In [130]:

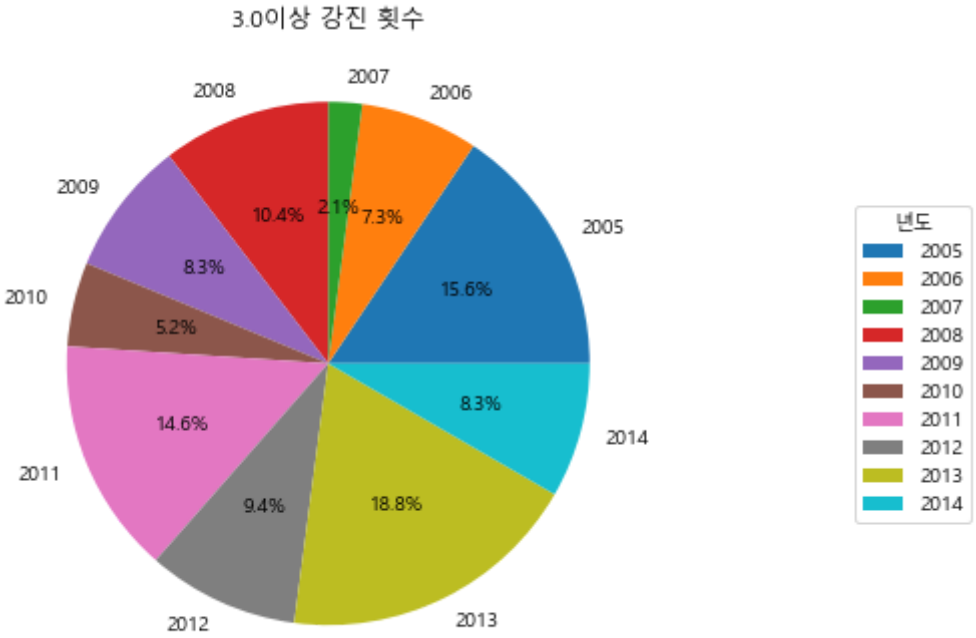
```
1 plt.pie(y, labels=x, autopct='%.1f%%')
2 plt.show()
```



제목과 범례 표시

In [135]:

```
plt.pie(y, labels=x,  autopct='%1.1f%%')
plt.legend(x, title='년도', loc="center right",  bbox_to_anchor=(1, 0, 0.5
plt.title("3.0이상 강진 횟수")
plt.show()
```

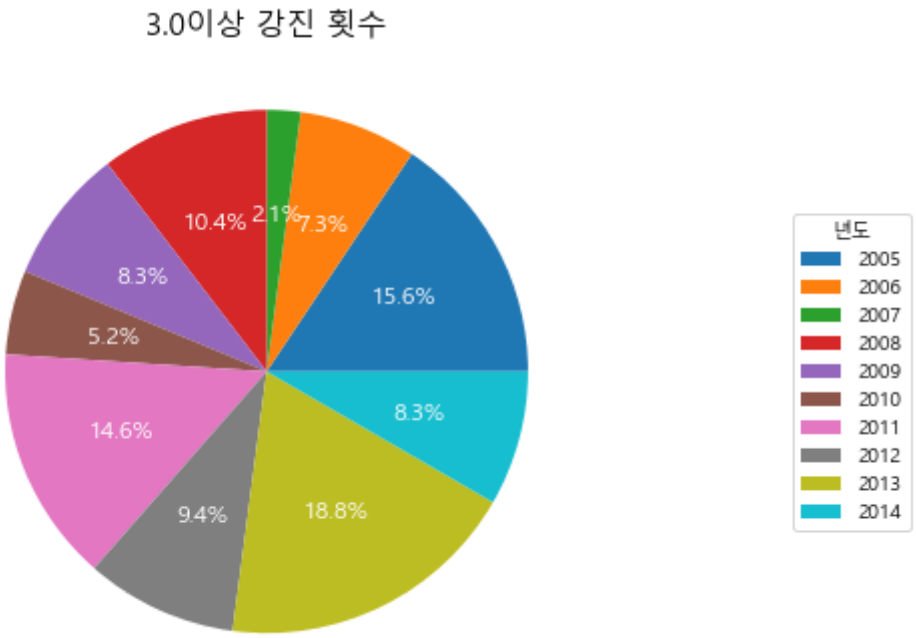


글씨 크기&색상 조정

- 단, textprops 옵션 중 color를 사용하면 labels=x 이 표시되지 않는다.

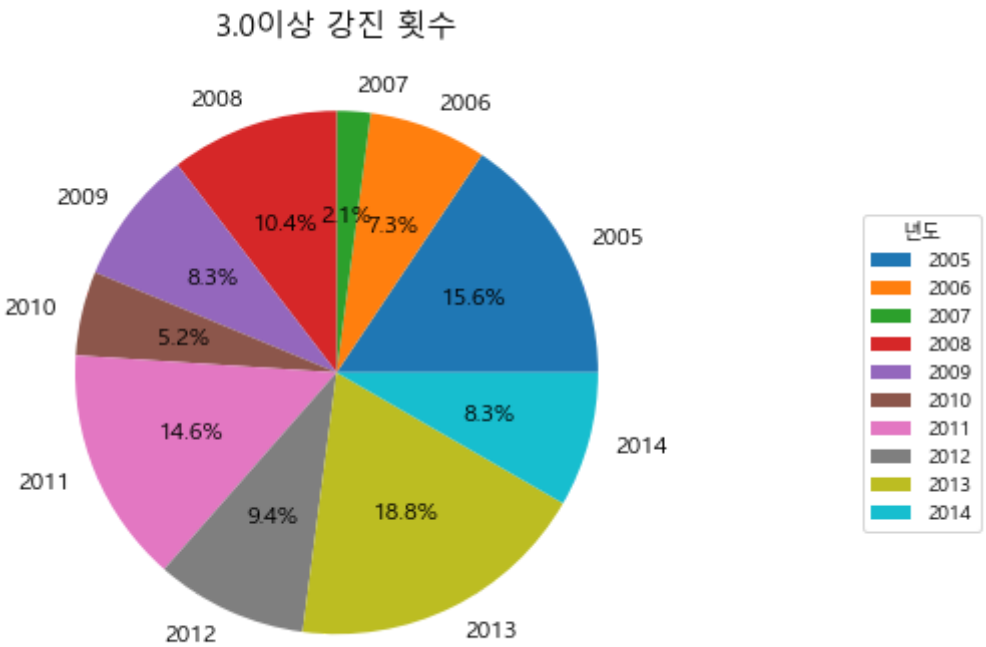
In [136]:

```
plt.pie(y, labels=x, autopct='%1.1f%%', textprops={'size':12, 'color':'w'}
plt.legend(x, title='년도', loc="center right",  bbox_to_anchor=(1, 0, 0.5
plt.title("3.0이상 강진 횟수", size=15)
plt.show()
```



In [144]:

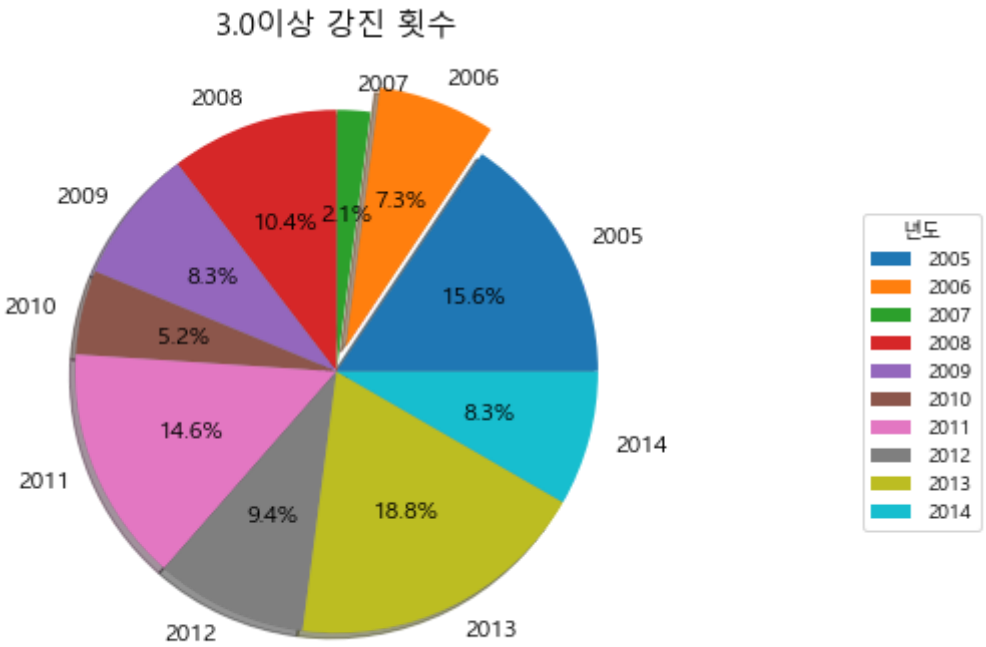
```
plt.pie(y, labels=x, autopct='%1.1f%%', textprops={'size':12})
plt.legend(x, title='년도', loc="center right",  bbox_to_anchor=(1, 0, 0.5
plt.title("3.0이상 강진 횟수", size=15)
plt.show()
```



특정 조각이 돌출되도록 표시

In [145]:

```
explode = (0, 0.1, 0, 0, 0, 0, 0, 0, 0, 0) #조각이 돌출되도록 표현

plt.pie(y, labels=x, autopct='%1.1f%%', textprops={'size':12},
        explode=explode, shadow=True, startangle=0)
plt.legend(x, title='년도', loc="center right",  bbox_to_anchor=(1, 0, 0.5,
plt.title("3.0이상 강진 횟수", size=15)
plt.show()
```



[예제 3-2]

In [155]:

```
data = {'한국계중국인':1076,'베트남':1183,'중국':684,'인도네이시아':579,'필리핀
        '캄보디아':366,'스리랑카':207,'일본':220,'네팔':119,'타이':135,'기타':49

df = pd.DataFrame(data, index=['인원'])
df.columns.name = '국가'

df
```

Out[155]:

국가	한국계중국인	베트남	중국	인도네이시아	필리핀	캄보디아	스리랑카	일본	네팔	타이	기타
인원	1076	1183	684	579	466	366	207	220	119	135	490

행별 합계(모집단 크기)

In [156]:

```
df.sum(axis=1)  # 행, 열(axis=0)
```

Out[156]:

```
인원     5525
dtype: int64
```

2.질적자료의 정리

In [169]:

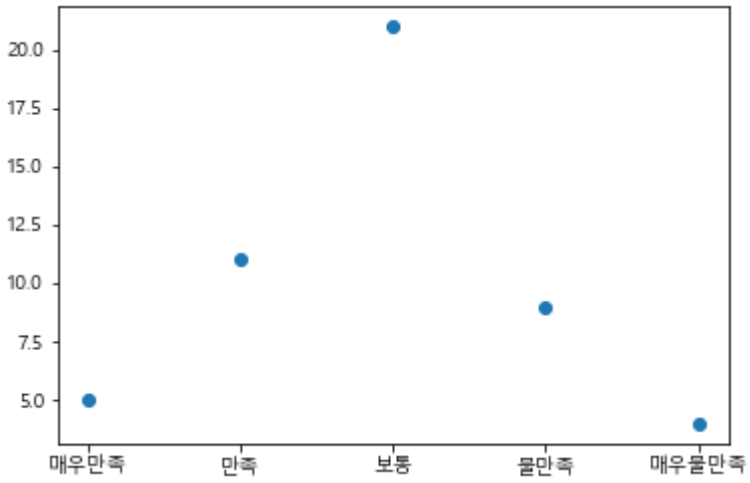
```
import matplotlib.pyplot as plt

# 한글출력 설정
plt.rcParams['font.family'] = 'Malgun Gothic'# '맑은 고딕'으로 설정
# 그래프 크기 지정
plt.rcParams['figure.figsize'] = (6, 4)
# 선 굵기 지정
plt.rcParams['lines.linewidth'] = 2
#matplotlib.rcParams['axes.unicode_minus'] = False
```

점도표 만들기

산점도

In [174]:

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

index = ['매우만족','만족','보통','불만족','매우불만족']
data  = [5,11,21,9,4]

plt.scatter(index, data)  # index:x, data:y
plt.show()
```

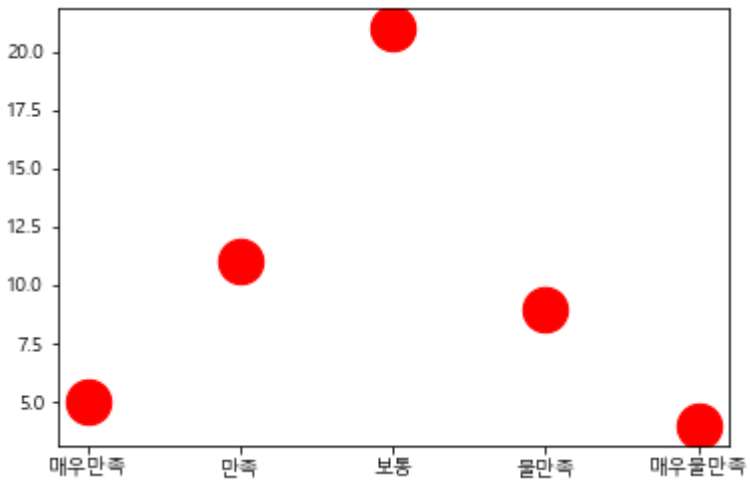


점 크기 & 색상 지정

- 색상
- 마커 모양: https://matplotlib.org/stable/api/markers_api.html (https://matplotlib.org/stable/api/markers_api.html)

In [176]:

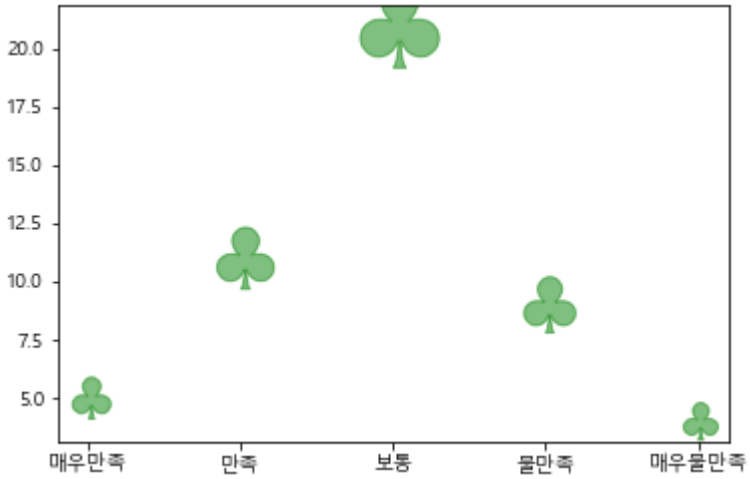
```
plt.scatter(index, data,  s=500, c='r')  # s:마커크기: 500, 컬러:red
plt.show()
```



점(마커) 모양 변경

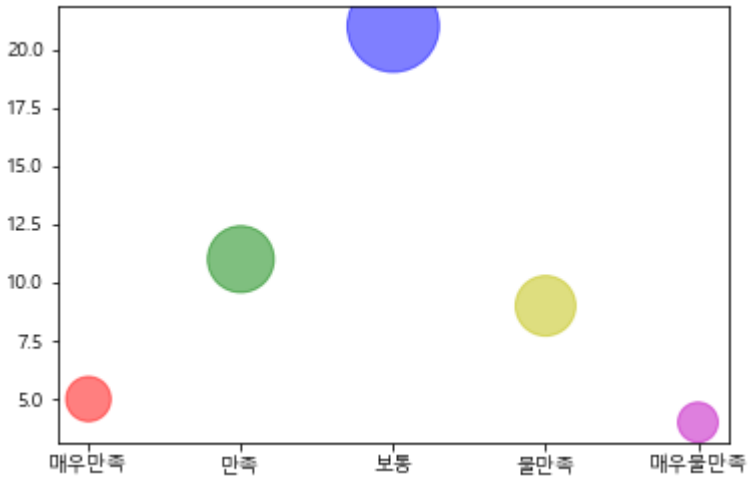
In [186]:

```
plt.scatter(index, data, s=data*100, c="g",
                alpha=0.5, marker=r'$\clubsuit$', label="Luck")
plt.show()
```



In [37]:

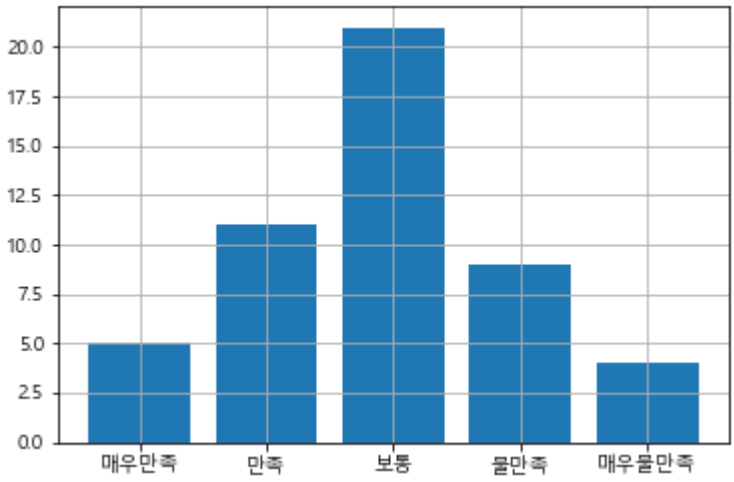
```
#값에 따라 점 크기, 컬러 다르게 지정
size = data * 100
colors=['r','g','b','y','m']
plt.scatter(index, data, s=size, c=colors, alpha=0.5)
plt.show()
```



막대 그래프

In [187]:

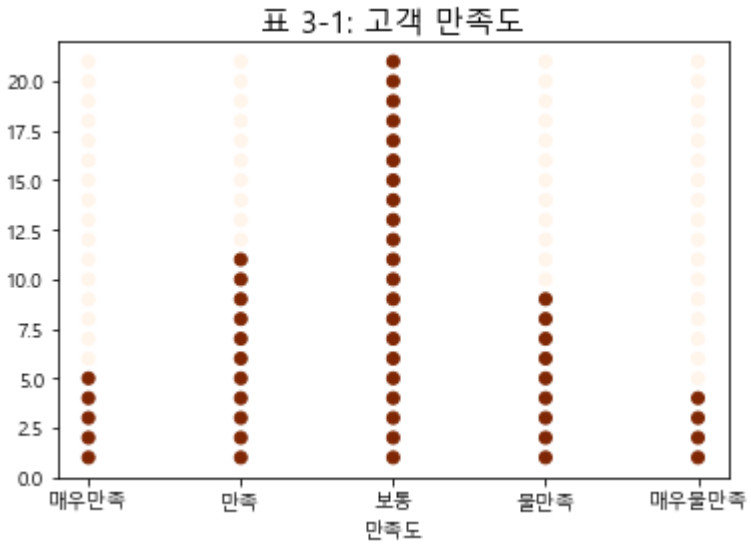
```
1 plt.bar(index, data)
2 plt.grid()
3 plt.show()
```



[실습] Q. 만족도 점도표 표현하기

In [204]:

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

# X축 Y축 데이터
index = ['매우만족','만족','보통','불만족','매우불만족']
data  = [5,11,21,9,4]

# 점도표를 위해 meshgrid()를 이용해,X,Y데이터를 가로 세로의 평면 배치로 만든다.
X = np.arange(len(index)) + 1  # X축: index
Y = np.arange(1, max(data)+1)  # Y축: data(도수)
x, y = np.meshgrid(X, Y)       # x,y 평면 범위(격자형태)

# 점도표 그리기:
# Y축이 실제값보다 작을 때까지 찍기
hist = np.array([5,11,21,9,4])
#plt.scatter(x, y, c= y<=hist, cmap="Greys") # c=The marker colors:array-lik
plt.scatter(x, y, c= y<=hist, cmap="Oranges")
plt.xlabel('만족도')
plt.xticks(ticks=X, labels=index)
plt.title('표 3-1: 고객 만족도', size=15)
plt.show()
```



[실습] Q. 도수표 만들기

Q.학년별 동아리 회원 수에 대한 도수표 작성하기

In [209]:

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

index = ['1학년','2학년','3학년','4학년']
data = [16,12,7,5]
df  = pd.DataFrame(data, index=index, columns=['도수'])
df.columns.name='인원'
df.index.name = '학년'

df['상대도수']  = [x/sum(data) for x in data]
df['백분율(%)'] = [x/sum(data)*100 for x in data]

df
```

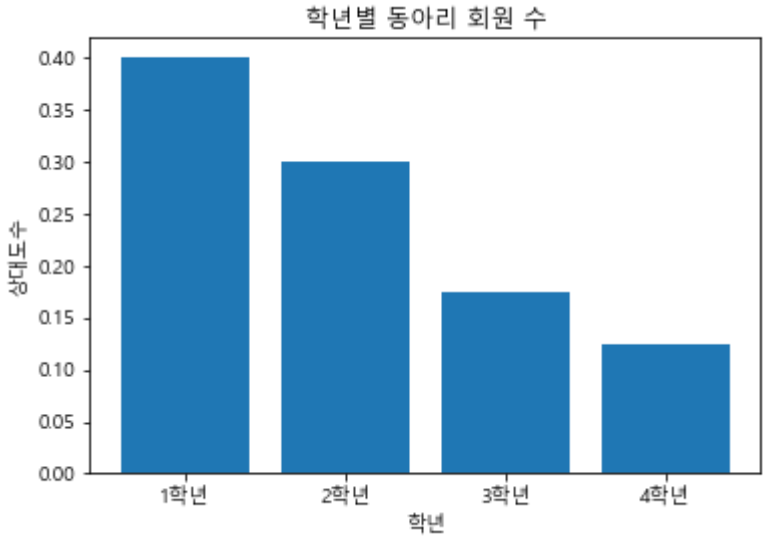
Out[209]:

인원	도수	상대도수	백분율(%)
학년			
1학년	16	0.400	40.0
2학년	12	0.300	30.0
3학년	7	0.175	17.5
4학년	5	0.125	12.5

Q.학년별 동아리 회원 수에 대한 막대 그래프

In [217]:

```
plt.bar(index, df['상대도수'], label=df['상대도수'])
plt.xlabel('학년')
plt.ylabel('상대도수')
plt.title('학년별 동아리 회원 수')
plt.show()
```

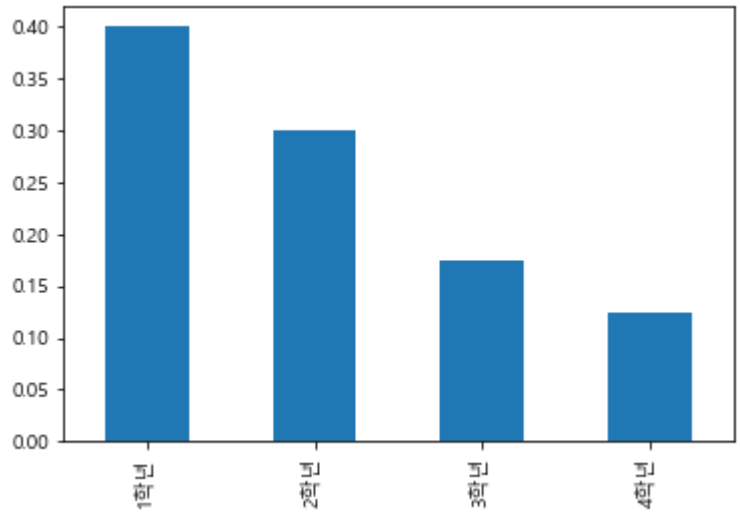


In [58]:

```
df['상대도수'].plot(kind='bar')   # kind='line' , pie
```

Out[58]:

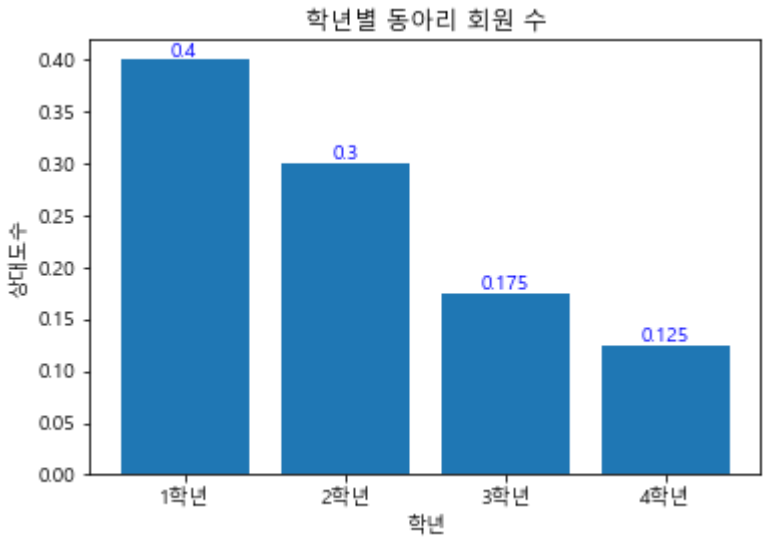
<AxesSubplot:>



막대그래프에 숫자 값 표시하기

In [219]:

```
plt.bar(index, df['상대도수'], label=df['상대도수'])
plt.xlabel('학년')
plt.ylabel('상대도수')
plt.title('학년별 동아리 회원 수')

# 막대그래프에 값 표시하기
for i, x in enumerate(index):
    plt.text(x, df['상대도수'][i], df['상대도수'][i],
             fontsize=10,
             color="blue",
             horizontalalignment='center',
             verticalalignment='bottom')

plt.show()
```



[실습] 꺾은선 그래프

Q.성별에 따른 고객 만족도

In [227]:

```
df  = pd.DataFrame([[3,7,10,4,2],[2,4,11,5,2]],
                   index=['남자','여자'],
                   columns=['매우만족','만족','보통','불만족','매우불만족'])
df
```

Out[227]:

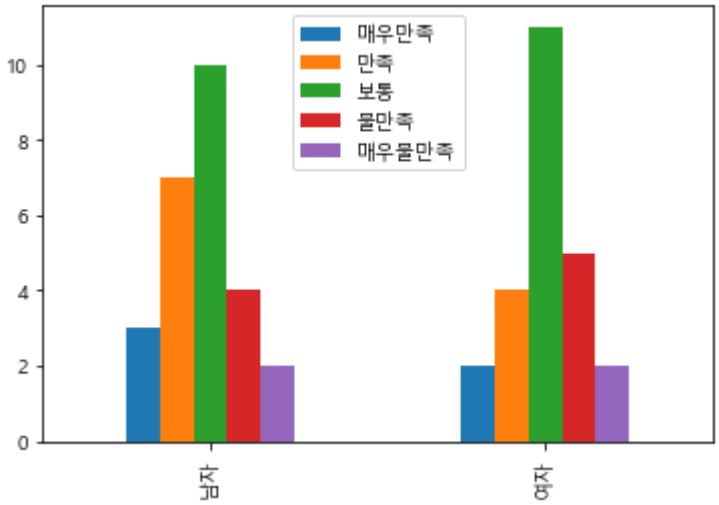
	매우만족	만족	보통	불만족	매우불만족
남자	3	7	10	4	2
여자	2	4	11	5	2

In [228]:

```
df.plot(kind='bar')
```

Out[228]:

<AxesSubplot:>



행 데이터 추출

- df.loc[인덱스명]
- df.iloc[인덱스]

In [231]:

```
df.loc['남자']
```

Out[231]:

```
매우만족      3
만족        7
보통       10
불만족       4
매우불만족     2
Name: 남자, dtype: int64
```

In [232]:

```
type(df.loc['남자'])  # 남자에 해당하는 행 데이터(시리즈 객체)
```

Out[232]:

```
pandas.core.series.Series
```

In [234]:

```
df.iloc[0]  # 남자
df.iloc[1]  # 여자
```

Out[234]:

```
매우만족      2
만족        4
보통       11
불만족       5
매우불만족     2
Name: 여자, dtype: int64
```

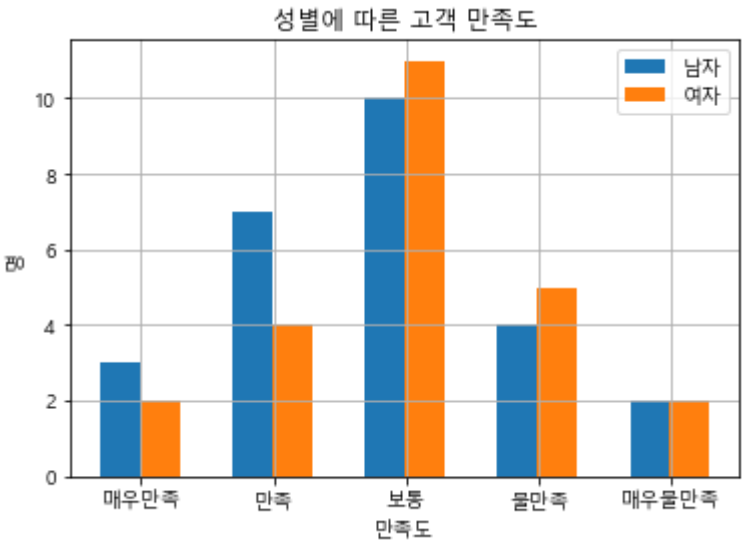
막대그래프

In [250]:

```
# 막대그래프
x = np.arange(len(df.columns))
width = 0.3

plt.bar(x - width/2, df.loc['남자'], width,label='남자')
plt.bar(x + width/2, df.loc['여자'], width,label='여자')
plt.title('성별에 따른 고객 만족도')
plt.xlabel('만족도')
plt.ylabel('명')
plt.xticks(x + 0.01, df.columns)  # 적절하게 그리드 선에 맞춘다.
plt.grid()
plt.legend()
```

Out[250]:

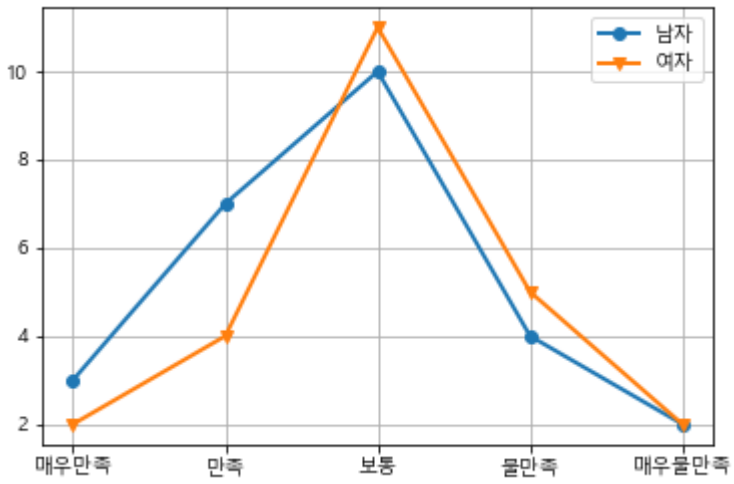
```
<matplotlib.legend.Legend at 0x23f3442ce80>
```



꺾은선 그래프

In [251]:

```
# 꺽은선 그래 프
plt.plot(df.columns, df.loc['남자'], 'o', linestyle='solid', label='남자', )
plt.plot(df.columns, df.loc['여자'], 'v', linestyle='solid', label='여자', )
plt.legend()
plt.grid()
plt.show()
```



원그래프

In [267]:

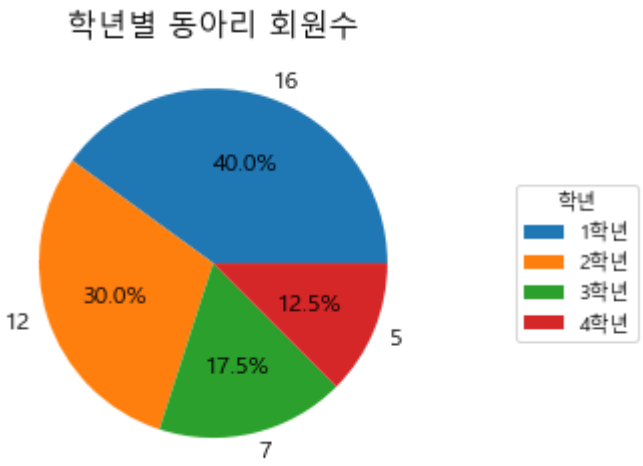
```
import matplotlib.pyplot as plt
import pandas as pd

index = ['1학년','2학년','3학년','4학년']
data  = [16,12,7,5]
df  = pd.DataFrame(val, index=index, columns=['도수'])

df['상대도수']  = [x/sum(data) for x in data]
df['백분율(%)'] = [x/sum(data)*100 for x in data]
df
```

Out[267]:

	도수	상대도수	백분율(%)
1학년	16	0.400	40.0
2학년	12	0.300	30.0
3학년	7	0.175	17.5
4학년	5	0.125	12.5

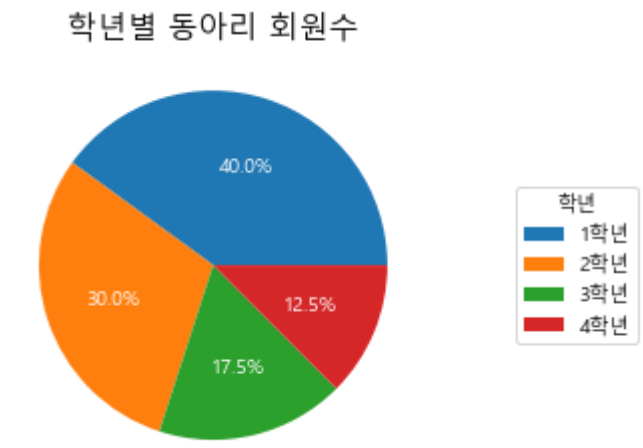
In [268]:

```
plt.pie(data, labels=data, autopct='%.1f%%', textprops={'size':12})
plt.legend(index, title='학년', loc="center right",
           bbox_to_anchor=(1, 0, 0.5, 1))
plt.title("학년별 동아리 회원수", size=15)
plt.show()
```



In [272]:

```
fig, ax = plt.subplots()
ax.pie(data, autopct='%.1f%%', textprops=dict(color="w"))
ax.legend(idx, title='학년', loc="center right",
          bbox_to_anchor=(1, 0, 0.5, 1))
ax.set_title("학년별 동아리 회원수", size=15)
plt.show()
```



3.양적자료의 정리

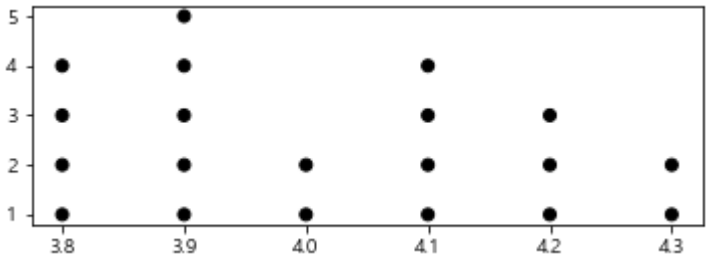
숫자료 표현할 수 있는 자료의 정리

[실습] 예제 3-8: 점도표 표현하기

Q.숫자데이터 점도표로 표현하기

In [280]:

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

datas = [4.1, 4.2, 3.8, 4.2, 3.9, 4.2, 3.9, 4.1, 3.9, 4.3,
         3.9, 3.8, 3.8, 4.0, 4.3, 3.8, 3.9, 4.1, 4.1, 4.0]

#1.고유한 측정값 찾기
index = list(set(datas))     # np.unique(datas)
index.sort()
data = [datas.count(i) for i in index]


#2.계급구간 만들기
X = np.arange(len(index)) + 1    # X축:데이터 속성
Y = np.arange(1, max(data) + 1)  # Y축:도수
x, y = np.meshgrid(X, Y)         # x-y 평면 범위(격자형태)


#3.점도표 그리기
plt.figure(figsize=(6, 2)) # 그래프 사이즈
plt.scatter(x, y, c= y<=data, cmap="Greys")
plt.xticks(ticks=X, labels=index)  #X축 레이블 지정함
plt.show()
```



[실습] 도수분포표 만들기

Q.핸드폰 사용시간을 계급수 K=5인 도수분포표를 만드시오

In [281]:

```
# 이미지 파일 사용하려면 설치
!pip install IPython
```

. . .

In [283]:

```
from IPython.display import Image
Image("image/핸드폰_사용시간.png")
```

Out[283]:

[표 3-4] 청소년의 핸드폰 사용시간 (단위 : 시간)

10	37	22	32	18	15	15	18	22	15
20	25	38	28	25	30	20	22	18	22
22	12	22	26	22	32	22	23	20	23
23	20	25	51	20	25	26	22	26	28
28	20	23	30	12	22	35	11	20	25

- 올림: math.ceil() # math 모듈내 함수
- 내림: math.floor() # math 모듈내 함수
- 반올림: round() # 사사오입

In [352]:

```
import math
import numpy as np
import pandas as pd

data = [10,37,22,32,18,15,15,18,22,15,
        20,25,38,28,25,30,20,22,18,22,
        22,12,22,26,22,32,22,23,20,23,
        23,20,25,51,20,25,26,22,26,28,
        28,20,23,30,12,22,35,11,20,25]

# 1.계급 수
k = 5
# 2.R : 최대측정값 - 최소측정값
R = max(data) - min(data)
# 3.계급 간격
w = math.ceil(R/k)
# 4.시작 계급값
s = min(data) - 0.5

# 전체 계급
bins = np.arange(s, max(data)+w, step=w)  #계급

print(f'계급수:{k}, R:{R}, 계급간격:{w}, 계급시작값:{s}')
print(f'계급:{bins}')

```

```
계급수:5, R:41, 계급간격:9, 계급시작값:9.5
계급:[ 9.5 18.5 27.5 36.5 45.5 54.5]
```

In [353]:

```
#계급구간
# index = []
# for i in range(len(bins)):
#     if i<(len(bins)-1):
#         index.append(f'{bins[i]} ~ {bins[i+1]}')
index = [f'{bins[i]} ~ {bins[i+1]}' for i in range(len(bins)) if i<(len(bins
index
```

Out[353]:

```
['9.5 ~ 18.5', '18.5 ~ 27.5', '27.5 ~ 36.5', '36.5 ~ 45.5', '45.5
~ 54.5']
```

In [354]:

```
#도수 데이터
hist, bins = np.histogram(data, bins)
hist
```

Out[354]:

```
array([10, 29,  8,  2,  1], dtype=int64)
```

In [355]:

```
# 도수분포표 만들기
df = pd.DataFrame(hist, index=index, columns=['도수'])
df.index.name = '계급간격'

df['상대도수'] = [x/sum(hist) for x in hist]
df['상대도수']
```

Out[355]:

```
계급간격
9.5 ~ 18.5     0.20
18.5 ~ 27.5    0.58
27.5 ~ 36.5    0.16
36.5 ~ 45.5    0.04
45.5 ~ 54.5    0.02
Name: 상대도수, dtype: float64
```

In [356]:

```
# tmp = []
# for i in range(len(hist)):
#     if i>0: tmp.append(sum(hist[:i+1]))
#     else: tmp.append(hist[i])
# df['누적도수'] = tmp
df['누적도수'] = [sum(hist[:i+1]) if i>0 else hist[i] for i in range(k)]
df['누적도수']
```

Out[356]:

```
계급간격
9.5 ~ 18.5     10
18.5 ~ 27.5    39
27.5 ~ 36.5    47
36.5 ~ 45.5    49
45.5 ~ 54.5    50
Name: 누적도수, dtype: int64
```

In [357]:

```
tmp = df['상대도수'].values
df['누적상대도수'] = [sum(tmp[:i+1]) if i>0 else tmp[i] for i in range(k)]
df['누적상대도수']
```

Out[357]:

```
계급간격
9.5 ~ 18.5     0.20
18.5 ~ 27.5    0.78
27.5 ~ 36.5    0.94
36.5 ~ 45.5    0.98
45.5 ~ 54.5    1.00
Name: 누적상대도수, dtype: float64
```

In [358]:

```
df['계급값'] = [ int((bins[x]+bins[x+1])/2) for x in range(k)]
df['계급값']
```

Out[358]:

```
계급간격
9.5 ~ 18.5     14
18.5 ~ 27.5    23
27.5 ~ 36.5    32
36.5 ~ 45.5    41
45.5 ~ 54.5    50
Name: 계급값, dtype: int64
```

In [359]:

```
df
```

Out[359]:

계급간격	도수	상대도수	누적도수	누적상대도수	계급값
9.5 ~ 18.5	10	0.20	10	0.20	14
18.5 ~ 27.5	29	0.58	39	0.78	23
27.5 ~ 36.5	8	0.16	47	0.94	32
36.5 ~ 45.5	2	0.04	49	0.98	41
45.5 ~ 54.5	1	0.02	50	1.00	50

In [360]:

```
df.loc['합계'] = [ sum(hist), sum(tmp),'','','' ]
df
```

Out[360]:

계급간격	도수	상대도수	누적도수	누적상대도수	계급값
9.5 ~ 18.5	10	0.20	10	0.2	14
18.5 ~ 27.5	29	0.58	39	0.78	23
27.5 ~ 36.5	8	0.16	47	0.94	32
36.5 ~ 45.5	2	0.04	49	0.98	41
45.5 ~ 54.5	1	0.02	50	1.0	50
합계	50	1.00			

히스토그램

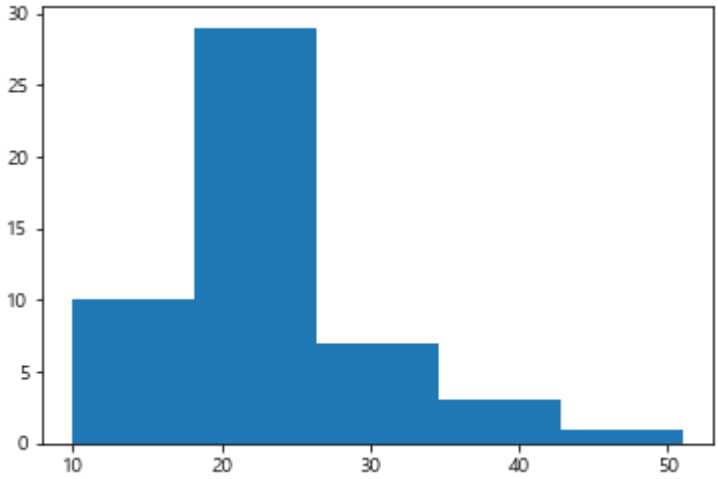
도수분포표로 작성한 자료를 시각적으로 쉽게 이해할 수 있도록 그린 그림

Q.청소년 1주일 동안의 핸드폰 사용시간을 계급수 K=5인 히스토그램

In [336]:

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

data = [10,37,22,32,18,15,15,18,22,15,
        20,25,38,28,25,30,20,22,18,22,
        22,12,22,26,22,32,22,23,20,23,
        23,20,25,51,20,25,26,22,26,28,
        28,20,23,30,12,22,35,11,20,25]

print(f'모집단: {len(data)}')
plt.hist(data)
plt.show()
```

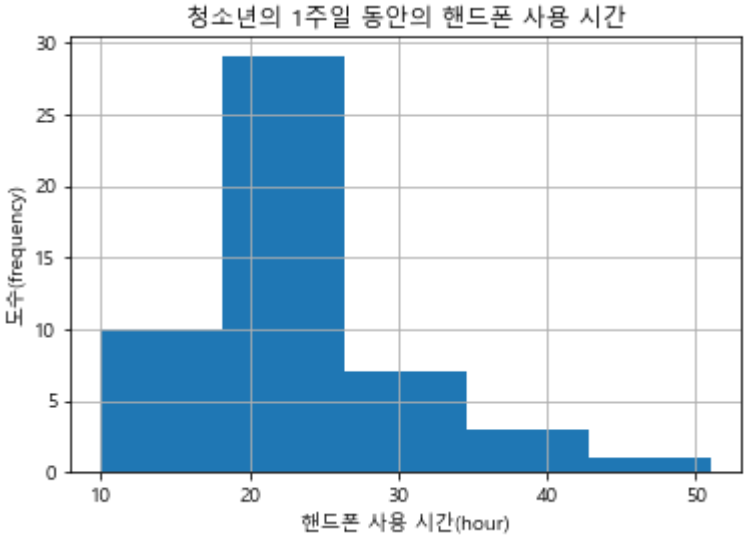
```
모집단: 50
```



계급 수 k=5 를 지정하여 히스토그램 그리기

In [337]:

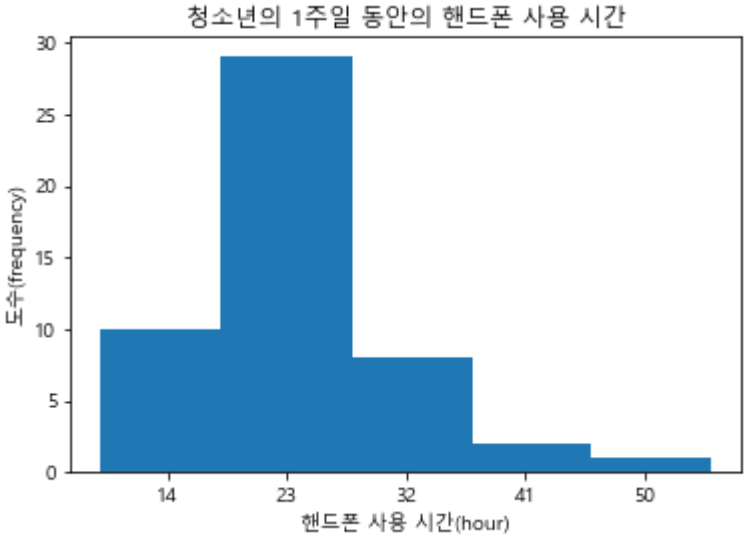
```
k = 5   #계급의 수

plt.hist(data, bins=k)
plt.grid()
plt.xlabel('핸드폰 사용 시간(hour)')
plt.ylabel('도수(frequency)')
plt.title('청소년의 1주일 동안의 핸드폰 사용 시간')
plt.show()
```



막대 그래프로 히스토그램 그리기

In [338]:

```
x = df['계급값'].values[:5]
y = df['도수'].values[:5]
plt.bar(x,y, width=10)
plt.xticks(ticks=x, labels=x)
plt.xlabel('핸드폰 사용 시간(hour)')
plt.ylabel('도수(frequency)')
plt.title('청소년의 1주일 동안의 핸드폰 사용 시간')
plt.show()
```



[실습] Q.3-9 예제 계급수 K=5인 히스토그램

In [377]:

```
import math
import numpy as np
import pandas as pd

data = [26,31,28,38,41,26,18,16,25,29,
        39,38,38,40,43,38,39,41,41,40,
        26,19,39,28,43,34,21,41,29,30,
        12,22,45,34,29,26,29,58,42,16,
        41,42,38,42,28,42,39,41,39,43]

# 1.계급 수
k = 5
# 2.R : 최대측정값 - 최소측정값
R = max(data) - min(data)
# 3.계급 간격
w = math.ceil(R/k)
# 4.시작 계급값
s = min(data) - 0.5

# 전체 계급
bins = np.arange(s, max(data)+w, step=w)  #계급

print(f'계급수:{k}, R:{R}, 계급간격:{w}, 계급시작값:{s}')
print(f'계급:{bins}')

#계급구간
index = [f'{bins[i]} ~ {bins[i+1]}' for i in range(len(bins)) if i<(len(bins

#도수 데이터
hist, bins = np.histogram(data, bins)


# 도수분포표 만들기
df = pd.DataFrame(hist, index=index, columns=['도수'])
df.index.name = '계급간격'

df['상대도수'] = [x/sum(hist) for x in hist]

df['누적도수'] = [sum(hist[:i+1]) if i>0 else hist[i] for i in range(k)]

tmp = df['상대도수'].values
df['누적상대도수'] = [sum(tmp[:i+1]) if i>0 else tmp[i] for i in range(k)]

df['계급값'] = [ int((bins[x]+bins[x+1])/2) for x in range(k)]

df.loc['합계'] = [ sum(hist), sum(tmp),'','','' ]

df

```

계급수:5, R:46, 계급간격:10, 계급시작값:11.5

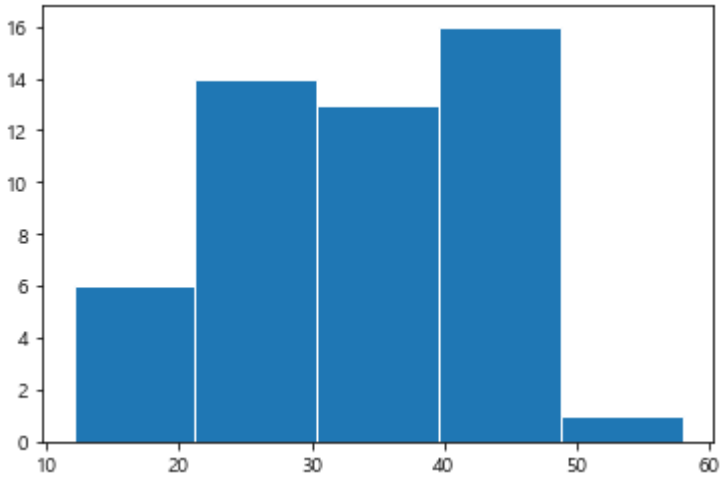
```
계급:[11.5 21.5 31.5 41.5 51.5 61.5]
```

Out[377]:

계급간격	도수	상대도수	누적도수	누적상대도수	계급값
11.5 ~ 21.5	6	0.12	6	0.12	16
21.5 ~ 31.5	15	0.30	21	0.42	26
31.5 ~ 41.5	20	0.40	41	0.82	36
41.5 ~ 51.5	8	0.16	49	0.98	46
51.5 ~ 61.5	1	0.02	50	1.0	56
합계	50	1.00			

In [378]:

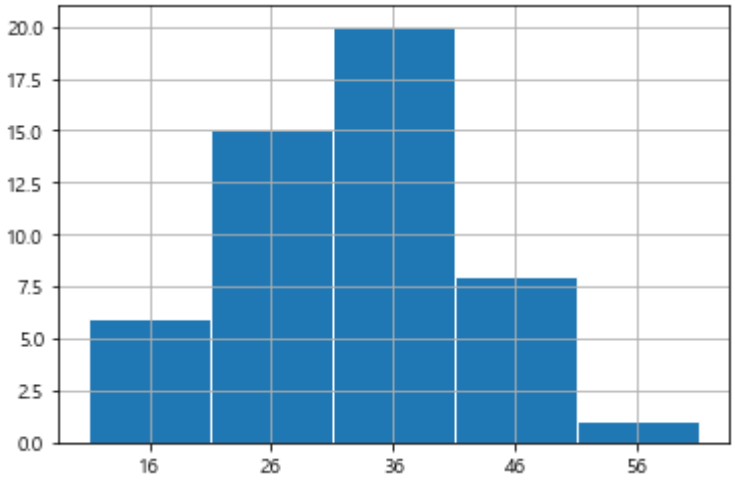
```
1 plt.hist(data, bins=5,  edgecolor='w')
2 plt.show()
```



In [383]:

```
x = df['계급값'].values[:5]
y = df['도수'].values[:5]
print(x)
plt.bar(x,y, width=10, edgecolor='w')
plt.xticks(ticks=x, labels=x)
plt.grid()
plt.show()
```

```
[16 26 36 46 56]
```



[실습] 예제 3-11 도수다각형 그리기

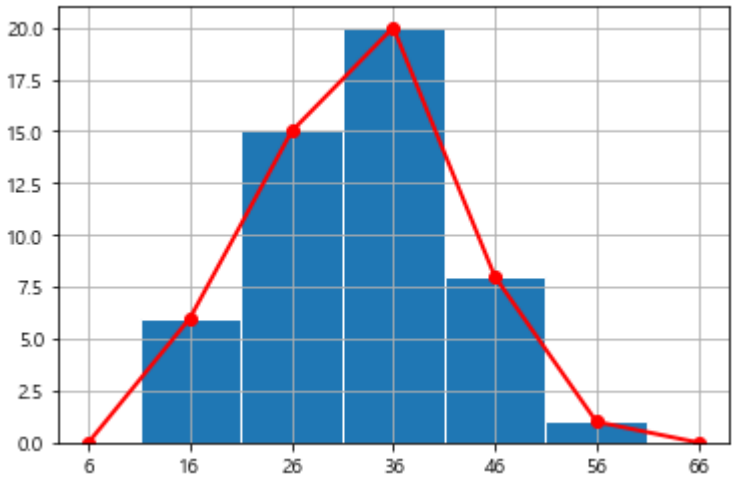
히스토그램에서 연속적인 막대의 상단 중심부를 선분으로 연결하여 다각형으로 표현한 그림

히스토그램 위에 도수다각형 그리기

In [388]:

```
x = df['계급값'].values[:5]
y = df['도수'].values[:5]
print(y)

# 선그래프의 점의 시작과 끝 추가하기
z = np.zeros(1)
xa= np.array([x[0]-(x[1]-x[0])])
xb= np.array([x[-1]+(x[1]-x[0])])
x1= np.hstack([np.hstack([xa,x]),xb]) # 시작점 끝점 추가
y1= np.hstack([np.hstack([z,y]),z])

plt.bar(x, y, width=10, edgecolor='w')
plt.plot(x1, y1, 'o', linestyle='solid', c='r')
plt.xticks(ticks=x1, labels=x1)
plt.grid()
plt.show()
```

```
[ 6 15 20  8  1]
```



[실습] 줄기-잎 그림

방법1: stemgraphic 라이브러리 이용

In [389]:

```
!pip install stemgraphic
```

. . .

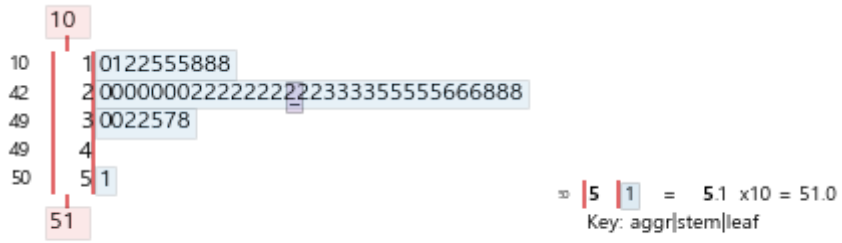
예제 3-4 데이터 이용

In [396]:

```
import stemgraphic

data = [10,37,22,32,18,15,15,18,22,15,
        20,25,38,28,25,30,20,22,18,22,
        22,12,22,26,22,32,22,23,20,23,
        23,20,25,51,20,25,26,22,26,28,
        28,20,23,30,12,22,35,11,20,25]

#stemgraphic.stem_graphic(data, scale=10)
stemgraphic.stem_graphic(data, scale=10, asc=False)
```

Out[396]:

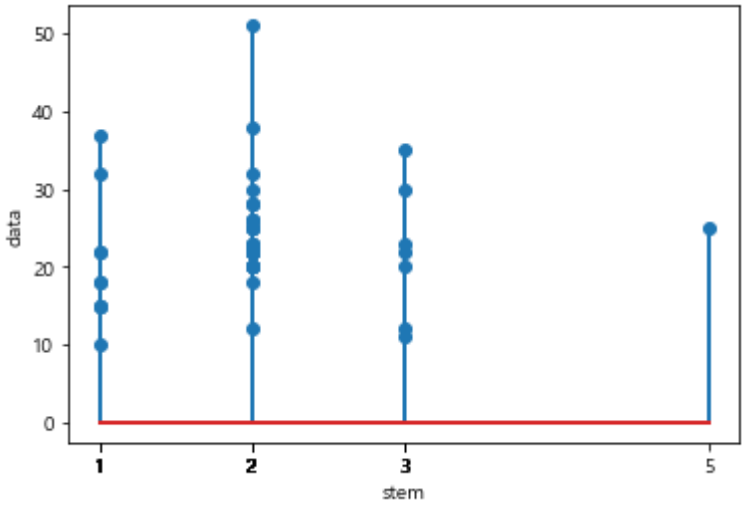
(<Figure size 540x144 with 1 Axes>, <Axes:>)



방법2: pyplot.stem 이용

In [391]:

```
stems = [i//10 for i in data]
stems.sort()
plt.stem(stems, data)
plt.xlabel('stem')
plt.ylabel('data')
plt.xticks(ticks=stems, label=stems)
plt.show()
```



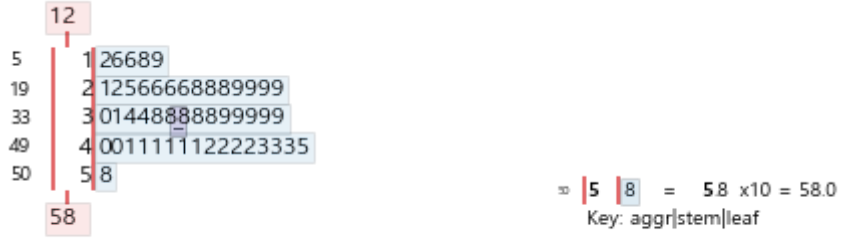
[실습] 예제 3-9 줄기-잎 그래프 그리기

In [395]:

```
data = [26,31,28,38,41,26,18,16,25,29,
        39,38,38,40,43,38,39,41,41,40,
        26,19,39,28,43,34,21,41,29,30,
        12,22,45,34,29,26,29,58,42,16,
        41,42,38,42,28,42,39,41,39,43]

import stemgraphic

#stemgraphic.stem_graphic(data, scale=10) asc=True
stemgraphic.stem_graphic(data, scale=10, asc=False)
```

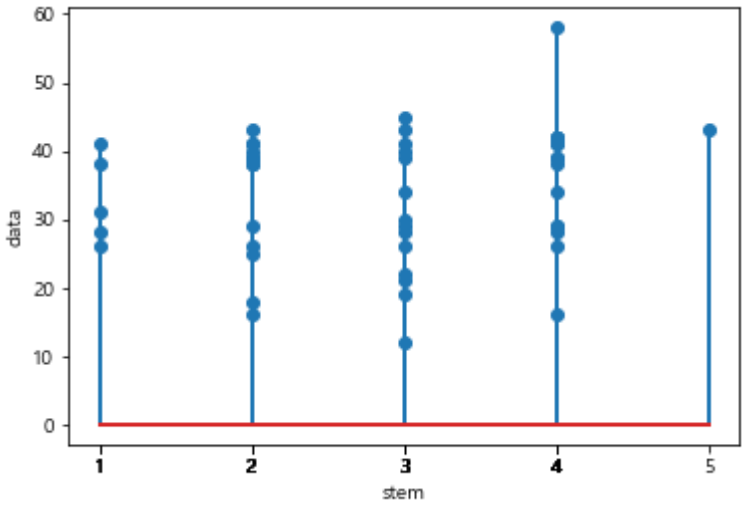
Out[395]:

```
(<Figure size 540x144 with 1 Axes>, <Axes:>)
```



In [393]:

```
stems = [i//10 for i in data]
stems.sort()
plt.stem(stems, data)
plt.xlabel('stem')
plt.ylabel('data')
plt.xticks(ticks=stems, label=stems)
plt.show()
```



끝