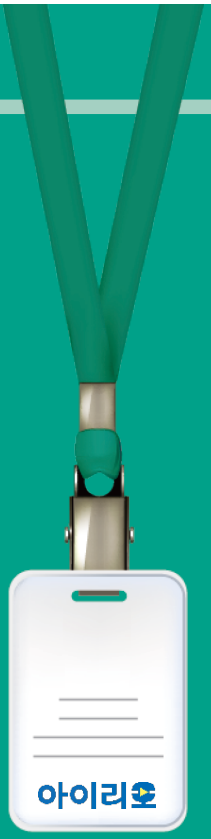


탐색적 데이터 분석 3강

- 데이터분석의 모든 것 -

강사명_이정인 저자

3. 데이터 분포 탐색

(실습 데이터 준비)

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

```
A_salary   = pd.Series([25, 28, 50, 60, 30, 35, 40, 70, 40, 70, 40, 100, 30, 30 ])
B_salary   = pd.Series([20, 40, 25, 25, 35, 25 ,20, 10, 55, 65, 100, 100, 150, 300])
```

```
df = pd.DataFrame ( {'A_salary': A_salary,
                     'B_salary' :B_salary    }
)
```

3.1 백분위수와 사분위수

- 백분위수 : 데이터를 정렬한 후, 특정 퍼센트 지점의 수
  - 최소값 : 0% 지점의 수
  - 최대값 : 100% 지점의 수
  - 중앙값 : 50% 지점의 수
- 백분위수 – quantile() 로 상위 10% 해당되는 지점의 두 회사의 연봉이 궁금하다면 90% 지점의 백분위수 구하기

```
# 90% 지점의 백분위 수
df.quantile(0.9)
```

```
A_salary     70.0
B_salary    135.0
Name: 0.9, dtype: float64
```

```
A_salary.quantile( 0.9 )
```

```
70.0
```

```
B_salary.quantile( 0.9 )
```

```
135.00000000000006
```

3.1 백분위수와 사분위수

◆ 사분위수 : 백분위수 중 0%, 25%, 50%, 75%, 100% 지점의 수

사분위 수

```
print( A_salary.quantile( 0 )  )
print( A_salary.quantile( 0.25 )  )
print( A_salary.quantile( 0.5 )  )
print( A_salary.quantile( 0.75 )  )
print( A_salary.quantile( 1 )  )
```

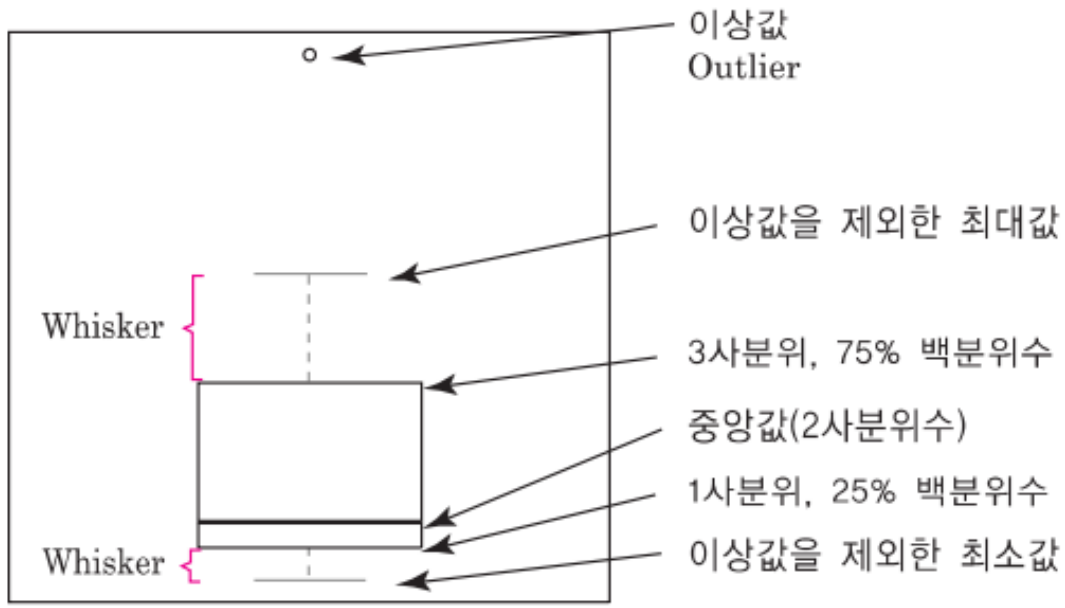
```
25.0
30.0
40.0
57.5
100.0
```

```
## 사분위수

print(np.percentile(A_salary, 0 )) # 최소값
print(np.percentile(A_salary, 25 )) # 1/4
print(np.percentile(A_salary, 50 )) # 2/4
print(np.percentile(A_salary, 75 )) # 3/4
print(np.percentile(A_salary, 100 )) # 최대값
```

```
25.0
30.0
40.0
57.5
100.0
```

3.2 상자그림

◆ 상자그림(boxplot) : 전체 관측값 범위와 사분위수, 그리고 이상값까지 시각적으로 확인해볼 수 있는 그래프



[그림 3-1] 상자그림

3.2 상자그림

◆ 상자그림 : A기업과 B기업의 연봉 데이터를 상자그림으로 비교

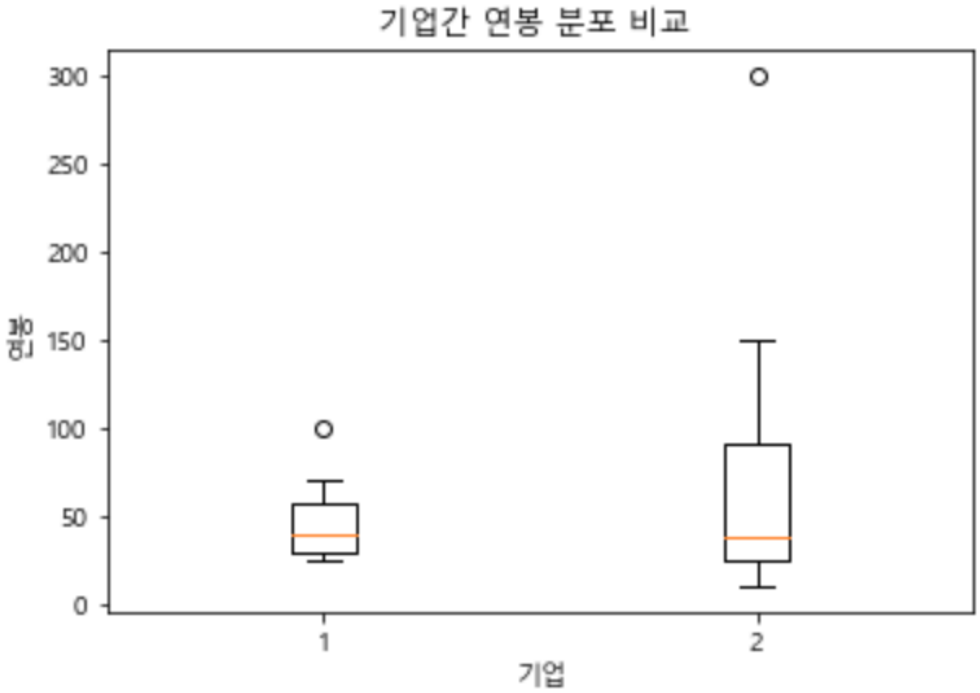
[그림 3-2] A기업과 B기업의 연봉 상자그림

(실습 시각화 라이브러리 준비)

```
import matplotlib
from matplotlib import font_manager,rc

#한글 폰트 등록
font_location = "c:/Windows/fonts/malgun.ttf" # Windows OS
# font_location = "/System/Library/fonts/AppleSDGothicNeo.ttc"  # Mac OS
font_name = font_manager.FontProperties(fname=font_location).get_name()
matplotlib.rc('font', family=font_name)


import matplotlib.pyplot as plt
%matplotlib inline

import warnings
warnings.filterwarnings("ignore")
```

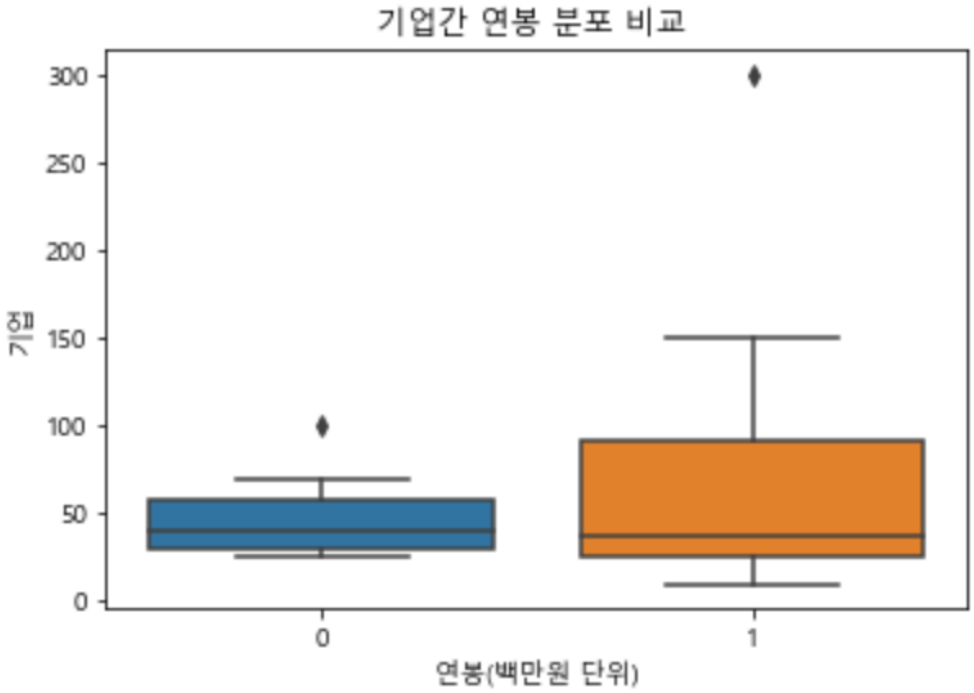
(실습) 상자그림 예1

```
plt.boxplot([A_salary, B_salary  ])
plt.title(" 기업간 연봉 분포 비교  ")
plt.xlabel("기업")
plt.ylabel("연봉")
plt.show()
```



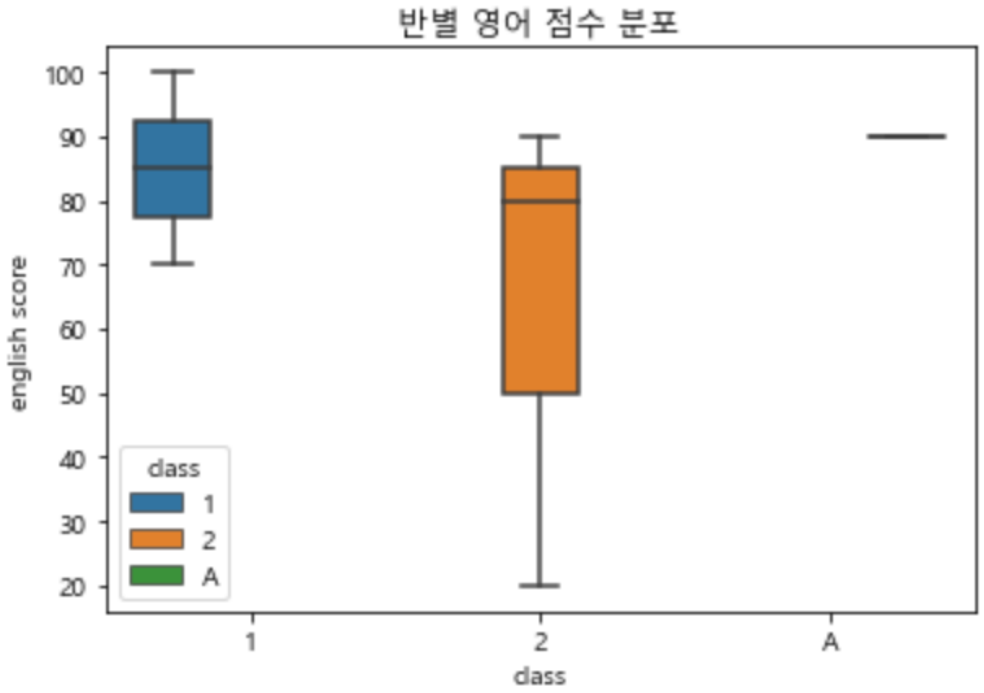
(실습) 상자그림 예2

```
import seaborn as sns
sns.boxplot( data=[A_salary , B_salary  ]     )
plt.title(" 기업간 연봉 분포 비교  ")
plt.xlabel("연봉(백만원 단위)")
plt.ylabel("기업")
plt.show()
```



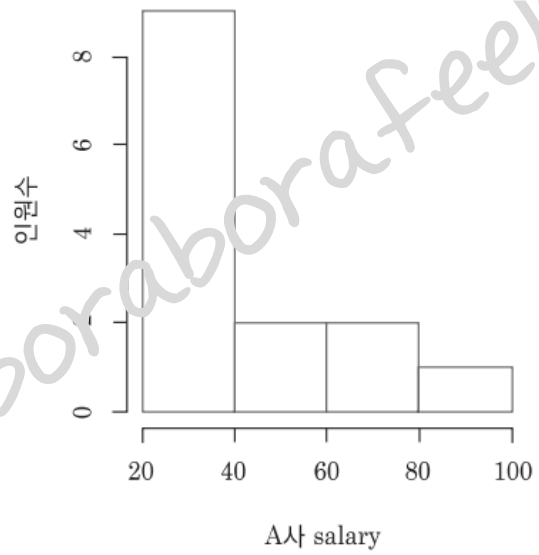
(실습) 상자그림 예3

```
students = pd.read_csv("data/students.csv")

import seaborn as sns
sns.boxplot(x="class", y='english'   ,data=students, hue="class" )
plt.title(" 반별 영어 점수 분포 ")
plt.xlabel("class")
plt.ylabel("english score")
plt.show()
```

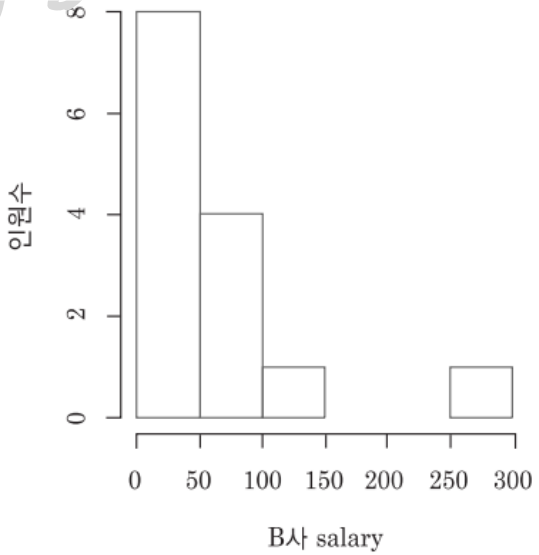


3.3 히스토그램

- 히스토그램 : 구간별 값의 분포 시각화, 데이터가 연속형 수치 데이터인 경우 데이터의 분포를 시각화하기에 좋은 그래프



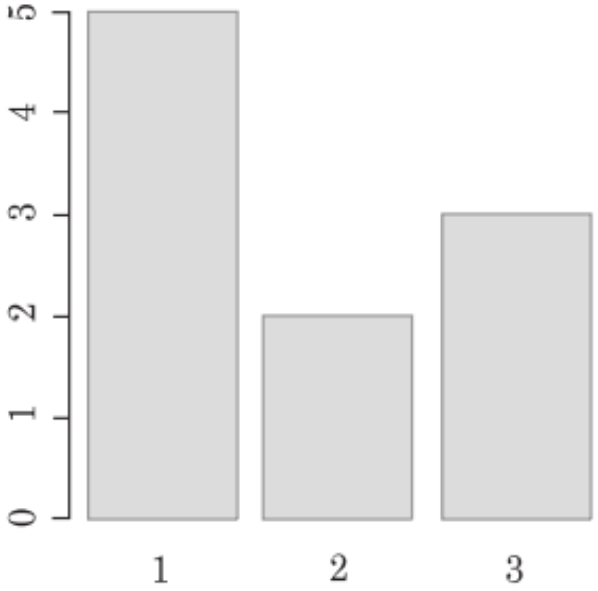
[그림 3-3] A기업 연봉 히스토그램



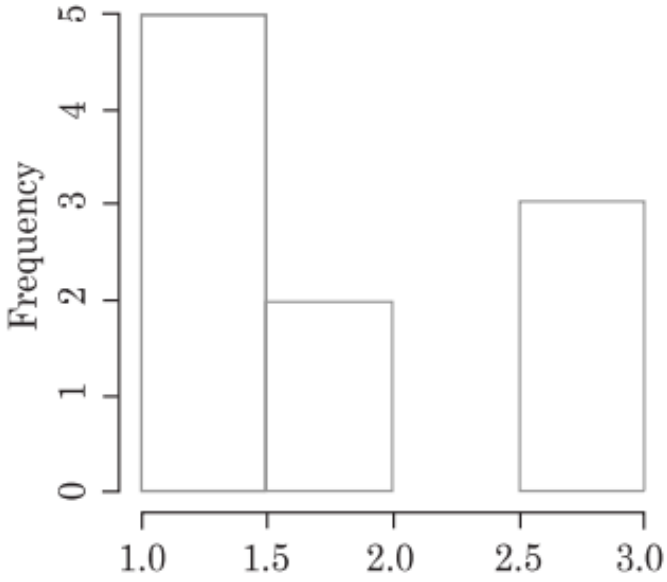
[그림 3-4] B기업 연봉 히스토그램

3.3 히스토그램과 막대그래프

- 막대 그래프 : 이산형 수치 데이터나 범주형 데이터의 경우 사용한다. 막대와 막대 사이를 떨어뜨려 표현한다.
- 히스토그램 : 연속형 수치 데이터의 경우 사용한다. 막대와 막대 사이를 붙여서 그린다.



[그림 3-5] 막대 그래프



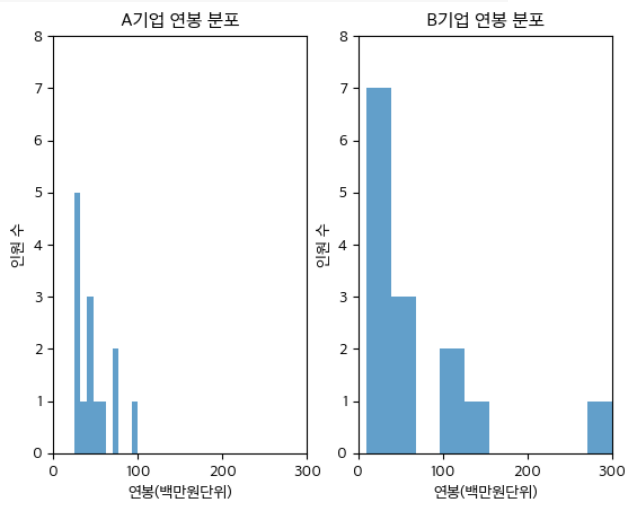
[그림 3-6] 히스토그램

(실습) 히스토그램

히스토그램

(연속형 변수 분포)

```
plt.figure()
plt.subplot(1,2,1)  # 1행 2열 그래프의 첫번째 그래프
plt.hist(A_salary, bins=10,  alpha=0.7, histtype='stepfilled')
plt.xlim(0,300)
plt.ylim(0,8)
plt.ylabel("인원 수")
plt.xlabel("연봉(백만원단위)")
plt.title (" A기업 연봉 분포 ")


plt.subplot(1,2,2) # 1행 2열 그래프의 두번째 그래프
plt.hist(B_salary, bins=10, alpha=0.7, histtype='stepfilled')
plt.xlim(0,300)
plt.ylim(0,8)
plt.ylabel("인원 수")
plt.xlabel("연봉(백만원단위)")
plt.title (" B기업 연봉 분포 ")
plt.show()
```



3.4 도수분포표

- 도수분포표 : 수집된 변수의 데이터를 범주 또는 동일한 크기의 구간으로 분류하고 각 구간마다 몇 개의 데이터가 존재하는지를 정리한 표로 많은 데이터를 알기 쉽게 정리하는 통계적인 방법 중의 하나
  - 데이터 특성을 요약하고 정리하는 기술 통계학에서 가장 기본적인 역할

3.4 도수분포표

◆ 범주형 데이터 도수분포표 생성 예

도수분포표

```
import pandas as pd

# 범주형 변수
blood = ['A', 'A', 'A', 'B', 'B', 'AB', 'O']
pd.Categorical(blood).value_counts()
```

```
A     3
AB    1
B     2
O     1
dtype: int64
```

3.4 도수분포표

◆ 수치형 (이산) 데이터 도수분포표 생성 예

```
# 수치형 (이산형) 변수
x = [1, 1, 1, 2, 3, 5, 5, 7, 8, 9]
import pandas as pd
pd.Series(x).value_counts()
```

```
1    3
5    2
2    1
3    1
7    1
8    1
9    1
dtype: int64
```

3.4 도수분포표

◆ 수치형 (연속) 데이터 도수분포표 생성 예

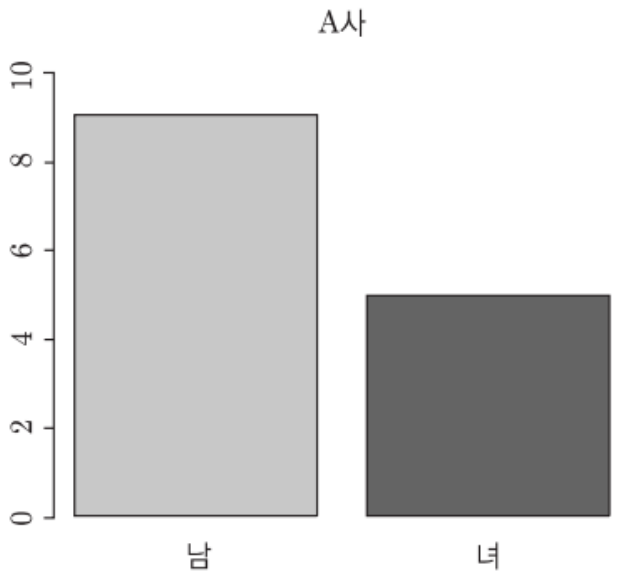
```
# 수치형 (연속형) 변수
weight = [47.2, 68.2, 55.3, 80.1, 47.5, 50.8, 71.1, 71.9, 81.9, 90.003]
hist, edges = np.histogram(weight, 4)
print(edges)
print(hist)
```

```
[47.2     57.90075 68.6015  79.30225 90.003  ]
[4 1 2 3]
```

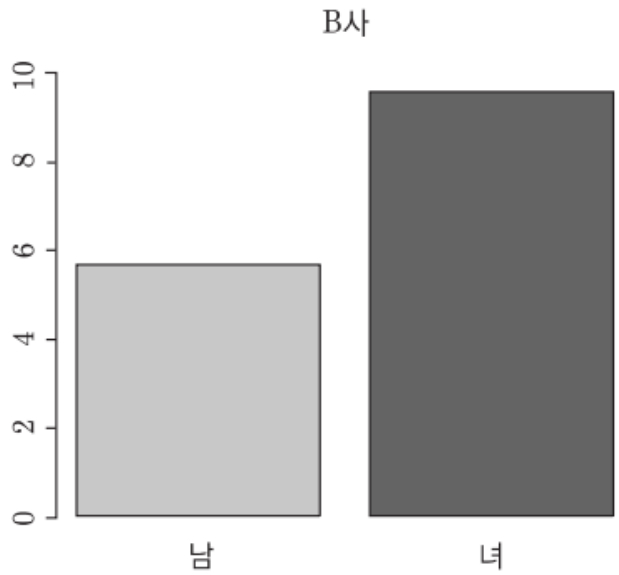
```
# 다음과 같은 빈도표
#. 47.2      ~  57.90075 : 4개
#. 57.90075  ~  68.6015  : 1개
#. 68.6015   ~  79.30225 : 2개
#. 79.30225  ~  90.003   : 3개
```

3.5 막대 그래프

◆ 막대그래프 : X축(범주형 데이터나 이산형 수치 데이터)의 도수분포표 또는 값(평균 등) 시각화



[그림 3-7] A사 남녀 분포 막대 그래프



[그림 3-8] B사 남녀 분포 막대 그래프

(실습) 데이터 준비

```
import pandas as pd

students = pd.read_csv("data/students.csv")
students
```

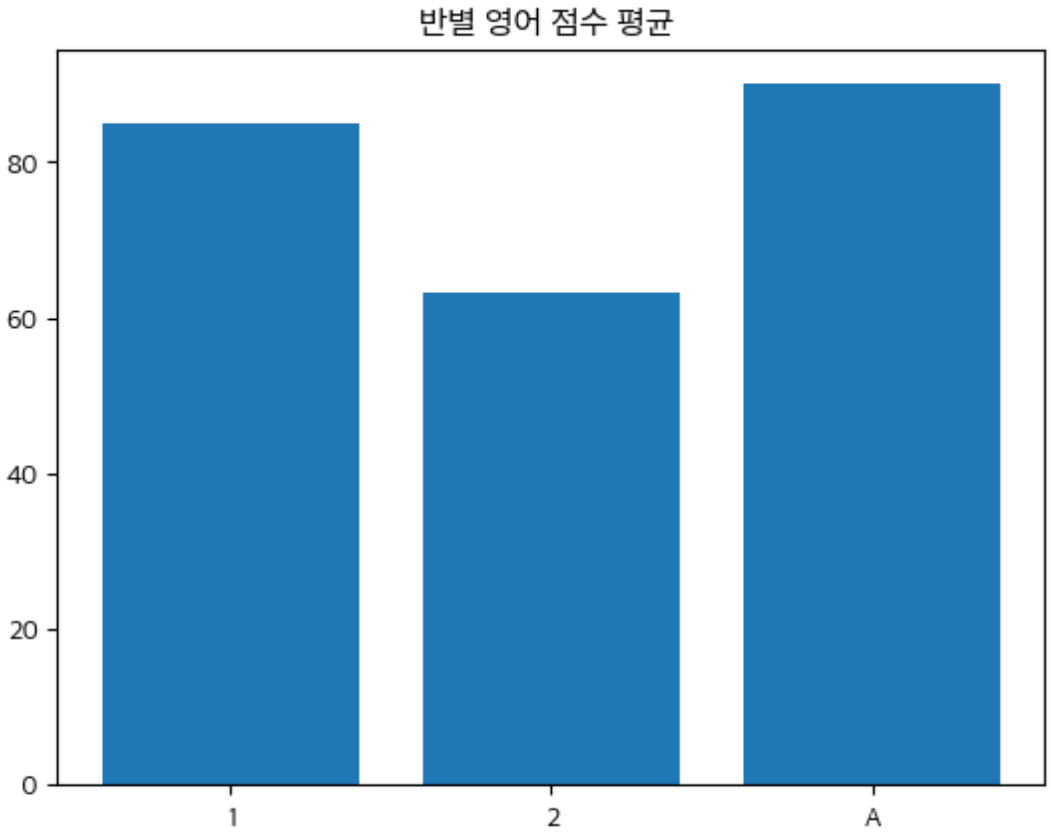
	english	math	class
0	100	999	1
1	90	90	1
2	80	80	1
3	70	20	1
4	20	90	2
5	90	100	2
6	80	80	2
7	90	99	A

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

data = students["english"].groupby(students['class'])

data_avg = data.mean()
print( data_avg    )  # 반별 영어 점수 평균
```

```
class
1     85.000000
2     63.333333
A     90.000000
Name: english, dtype: float64
```

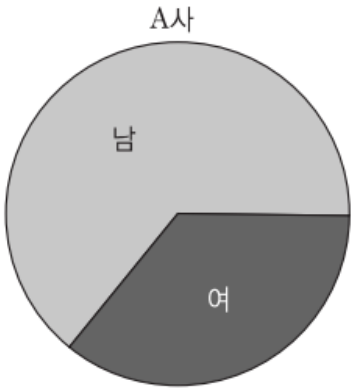

(실습) 막대 그래프

```
plt.title("반별 영어 점수 평균 ")
plt.bar( data_avg.index, data_avg)
plt.show()
```

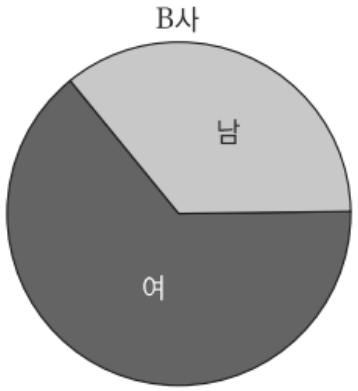


3.6 파이 그래프

- 파이 그래프 : (Pie chart, 원 그래프)는 범주별 구성 비율을 원형으로 표현한 그래프
- 분포의 시각화를 위해 사용, 범주가 몇 개 되지 않고, 차이가 확연한 경우 유용



[그림 3-9] A사 남녀 분포 파이 그래프



[그림 3-10] B사 남녀 분포 파이 그래프

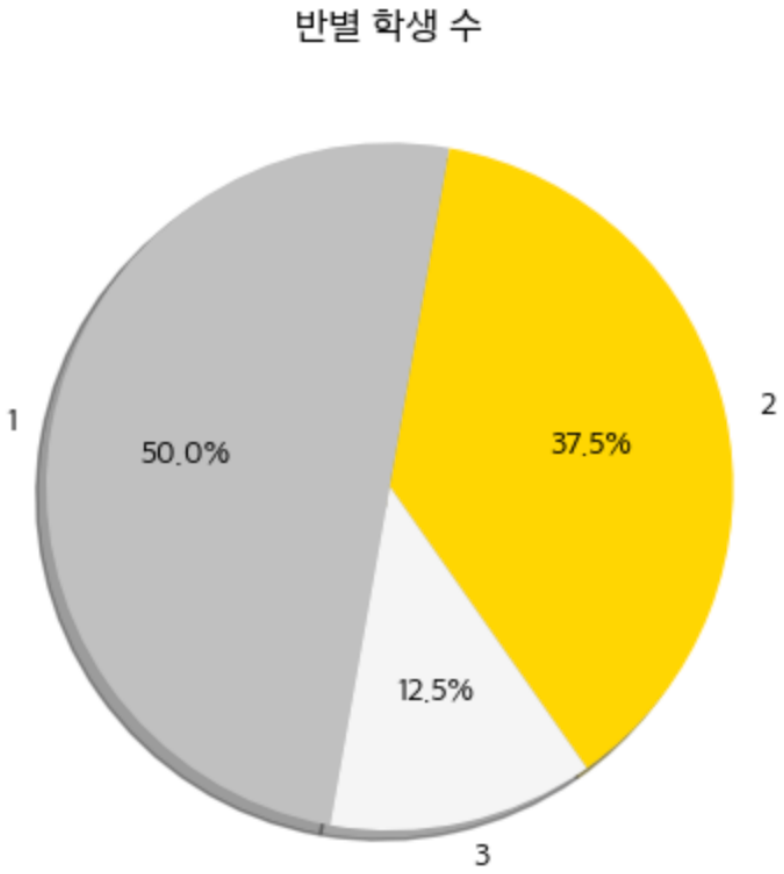
(실습) 데이터 준비

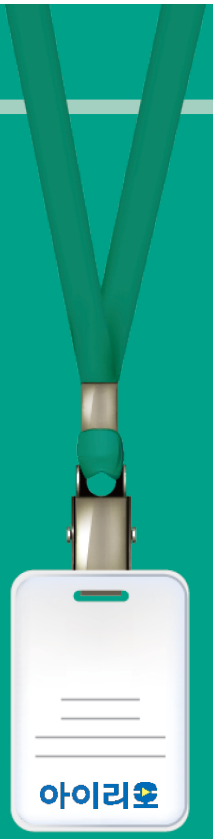
```
import matplotlib.pyplot as plt
%matplotlib inline

colors = ['silver', 'gold', 'whitesmoke']

data = students["english"].groupby(students['class'])
print( data.mean() )  # 반별 영어 점수 평균
print( data.size() )  # 반별 학생수
```

```
class
1    85.000000
2    63.333333
A    90.000000
Name: english, dtype: float64
class
1    4
2    3
A    1
Name: english, dtype: int64
```

(실습) 파이 그래프

```
plt.pie(data.size(), labels=[1,2,3], autopct='%.1f%%', startangle=260, counterclock=False,
        shadow=True, colors=colors)
plt.title("반별 학생 수")
plt.show()
```





수고하셨습니다.

문의사항 : krishnaleela@daum.net