파이썬 데이터 분석 실무과정

이 선 순

데이터 요약 및 표현

데이터프레임에 함수 적용

데이터프레임명.함수()	▶ pandas 데이터프레임에 함수를 사용하는 형식

■ 데이터프레임에 관한 정보를 얻기에 유용한 함수

함수	설명
info() head() tail() shape dtypes columns del(dataframe 이름)	• 지정된 데이터프레임의 기본 정보 출력 • 지정된 데이터프레임의 처음 5개의 데이터를 출력, n=3(처음 3개의 데이터 출력) • 지정된 데이터프레임의 마지막 5개의 데이터를 출력 • 데이터프레임의 차원(행, 열의 개수)출력 • 데이터프레임의 변수들의 데이터 타입 출력 • 데이터프레임의 변수들의 목록 출력 • 작업공간에서 객체 제거하기

```
In [2]: import seaborn as sns

In [3]: iris = sns.load_dataset('iris')

In [4]: iris.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   species       150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

▶ iris 데이터프레임에 대한 기본 정보 출력

```
In [5]: iris.shape
Out[5]: (150, 5)

In [6]: iris.columns
Out[6]:
Index(['sepal_length', 'sepal_width', 'petal_length', 'petal_width',
       'species'],
      dtype='object')
```

▶ iris 데이터프레임의 차원 출력 : 150행, 5열 출력

▶ iris 데이터프레임의 변수들의 목록 출력

```
In [7]: iris.dtypes
Out[7]:
sepal_length    float64
sepal_width     float64
petal_length    float64
petal_width     float64
species          object
dtype: object
```

▶ iris 데이터프레임의 변수들의 데이터 타입 출력

데이터프레임의 가공

- 통계분석시 필요에 따라 데이터프레임의 내용을 수정 또는 변경

- 예. test.xlsx는 50명 학생의 자료이다.

id	class	gender	kor	math	eng
101	1	M	10	10	10
102	2	M	30	30	30
103	4	F	30	30	10
104	4	M	30	30	20
105	4	M	10	10	10
106	5	M	20	0	10
107	3	M	10	10	0
108	5	F	10	10	10
109	5	M	10	10	30
110	2	F	20	30	20
⋮	⋮	⋮	⋮	⋮	⋮
150	2	F	20	30	20

■ 숫자형 변수를 문자형 변수로 변환

- astype함수를 이용하여 숫자로 입력된 변수를 문자로 변환 : 이것은 양적자료를 질적자료로 변환하는 것과 같아 요인변수 설정시 이용
- class의 값이 1='1반', 2='2반', 3='3반', 4='4반', 5='5반'으로 변환 : class에 해당하는 label을 딕셔너리로 생성 후 map함수를 이용하여 해당변수에 대하여 label을 지정

```
import pandas as pd
test = pd.read_excel("d:/data/test.xlsx")
test.dtypes

test["class"] = test["class"].astype("str")

class_label = {'1':'1반','2':'2반','3':'3반',
               '4':'4반','5':"5반"}
test["class"] = test["class"].map(class_label)
```

▶ test 데이터프레임의 변수들의 데이터 타입 확인

▶ class변수를 문자형(질적자료)로 변환하여 test에 다시 저장

▶ 범주 레이블 지정
- class_label을 딕셔너리로 저장
- map함수를 이용하여 class변수에 레이블을 지정
  : 1='1반', 2='2반', 3='3반', 4='4반', 5='5반'

```
In [17]: import pandas as pd
    ...:
    ...: test = pd.read_excel("c:/data/test.xlsx")
    ...:
    ...: test.dtypes
Out[17]:
id          int64
class       int64
gender     object
kor         int64
math        int64
eng         int64
dtype: object

In [18]: test["class"] = test["class"].astype("str")

In [19]: class_label = {'1':'1반','2':'2반','3':'3반','4':'4반','5':"5반"}
    ...:
    ...: test["class"] = test["class"].map(class_label)

In [20]: test.dtypes
Out[20]:
id          int64
class      object
gender     object
kor         int64
math        int64
eng         int64
dtype: object
```

test - DataFrame

Index	id	class	gender	kor	math	eng
0	101	1반	M	10	10	10
1	102	2반	M	30	30	30
2	103	4반	F	30	30	10
3	104	4반	M	30	30	20
4	105	4반	M	10	10	10
5	106	5반	M	20	0	10
6	107	3반	M	10	10	0
7	108	5반	F	10	10	10
8	109	5반	M	10	10	30
9	110	2반	F	20	30	20
10	111	1반	M	10	10	10
11	112	2반	M	30	30	30
12	113	4반	F	30	30	10
13	114	4반	M	30	30	20

Format Resize ☑ Background color ☑ Column min/max

■ 데이터프레임에 새로운 변수 추가

- loc함수를 이용하여 해당개체를 선택하고 조건식을 이용하여 기존의 변수의 범주를 새로운 범주를 갖는 문자형 변수 생성하여 추가

```
test.loc[test["class"]<='2반', "group"]="A "
test.loc[test["class"]>='3반', "group"]="B"
```

▶ loc 함수와 조건식을 이용하여 class를 A(2반이하), B(3반이상)로 구분하는 문자형 변수 "group" 생성

test - DataFrame

Index	id	class	gender	kor	math	eng	group
0	101	1반	M	10	10	10	A
1	102	2반	M	30	30	30	A
2	103	4반	F	30	30	10	B
3	104	4반	M	30	30	20	B
4	105	4반	M	10	10	10	B
5	106	5반	M	20	0	10	B
6	107	3반	M	10	10	0	B
7	108	5반	F	10	10	10	B
8	109	5반	M	10	10	30	B
9	110	2반	F	20	30	20	A
10	111	1반	M	10	10	10	A
11	112	2반	M	30	30	30	A
12	113	4반	F	30	30	10	B
13	114	4반	M	30	30	20	B

모집단과 표본

- **모집단(Population) : 관심의 대상이 되는 모든 추출단위의 특성값을 모아놓은 것**
  - 유한모집단(Finite Population) : 유한개의 추출단위로 구성된 모집단
  - 무한모집단(Infinite Population) : 무한개의 추출단위를 가지는 모집단

- **표본(sample) : 관심대상 중 자료수집으로 뽑힌 일부분, 실제로 관측한 추출단위의 특성값의 모임**

- **추출단위(sampling unit) : 전체를 구성하는 각 개체들**

- **특성값(Characteristics) : 각 추출단위의 특성을 나타내는 값**

- **관찰값(Observed values) : 표본의 특성값**

참고 : 자료(Data)의 분류

■ 양적자료(quantitative data) 또는 숫자형 자료(numerical data)

- 계측데이터(metric)
- 길이, 무게 등과 같이 양적인 수치로 측정되거나 몇 개인가를 세어 측정하는 변수
- 사칙연산 가능
  - ① 연속형 자료(Continuous data) : 무게, 키, 전구수명등과 같이 셀 수 없는 소수점을 포함하는 형태의 자료
  - ② 이산형 자료(Discrete data) : 자녀의 수, 교통사고건수, 불량품의 수등과 같이 셀 수 있는 정수 형태의 자료

■ 질적자료(qualitative data) 또는 범주형 자료(categorical data)

- 비계측자료(nonmetric)
- 학년, 성별, 등급 등
- 조사대상을 특성에 따라 범주로 구분하여 측정된 변수
- 사칙연산 불가능
  - ① 명목형 자료(Nominal data) : 성별, 직업, 지역등과 같이 자료값의 크기나 순서에 대한 의미가 없고 자료값 자체의 이름만 의미를 부여한 자료
  - ② 순서형 자료(Ordinal Data): 학년, 순위, 성적등급 등과 같이 어떤 기준에 따라 자료값들의 순서의 개념이 있는 자료

수치를 이용한 일변량 자료의 요약

- **통계량 : 표본으로부터 계산되는 표본의 특성값**
  - **중심위치 측도 : 평균, 중앙값**
  - **산포 측도 : 분산, 표준편차, 사분위수 범위**

■ 질적(범주형)자료의 요약

▸ 빈도표(frequency table)

- value_counts 함수(pandas 데이터프레임)
- groupby함수와 size함수(pandas 데이터프레임)

`import seaborn as sns`	▶ 필요한 라이브러리를 import한다.
`iris = sns.load_dataset('iris')`	
`iris.species.value_counts()` `iris.groupby("species").size()`	▶ 빈도표 작성 : value_counts 함수(pandas 데이터프레임) groupby함수, size함수(pandas 데이터프레임)

```
In [8]: import seaborn as sns
   ...: iris = sns.load_dataset('iris')

In [9]: iris.species.value_counts()
Out[9]:
species
setosa        50
versicolor    50
virginica     50
Name: count, dtype: int64

In [10]: iris.groupby("species").size()
Out[10]:
species
setosa        50
versicolor    50
virginica     50
dtype: int64
```

■ 양적자료의 요약

- 양적(숫자형)자료의 기술통계량을 구하여 수치적으로 요약

▸ 중심위치 측도

- 자료들이 어느 위치를 중심으로 자료들이 모여 있는지를 나타내는 측도

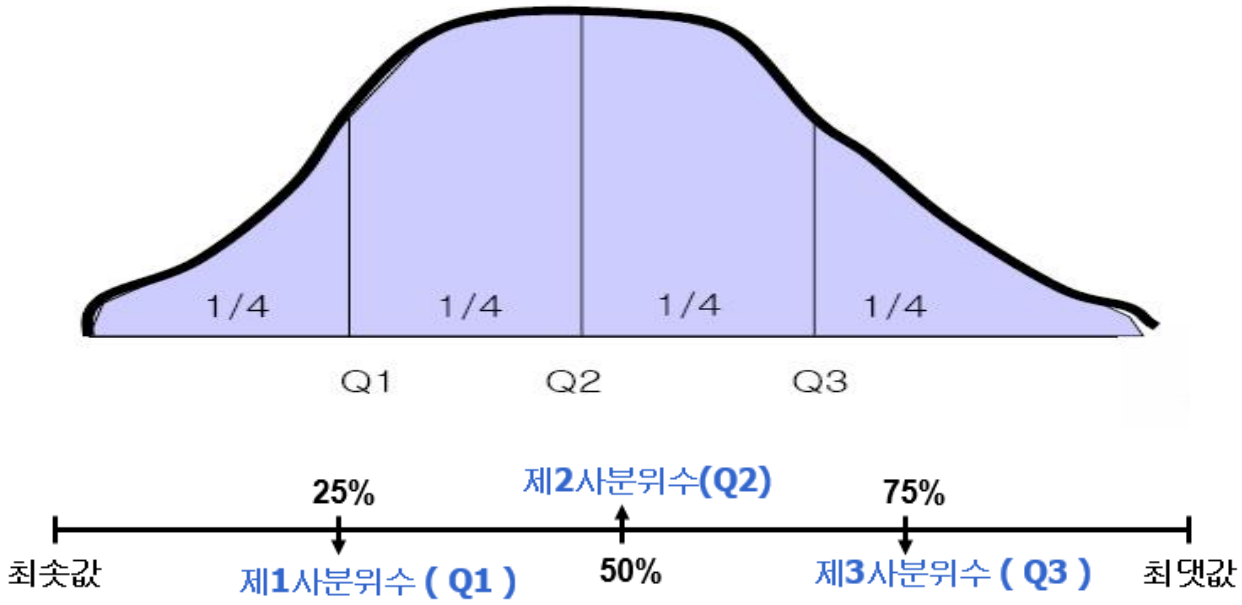
- 평균(mean) : $\bar{x} = \frac{1}{n}(x_1 + x_2 + ... + x_n)$
  - 중심위치의 측도로 가장 많이 사용
  - 모든 관측값들을 반영하여 계산
  - 비대칭자료, 이상점(아주 크거나 작은값)을 포함하는 자료의 대표값으로 부적절함 (why?)

- 중앙값(median) : 자료를 작은 값부터 크기 순서로 나열했을 때, 50%에 위치하는 값으로 전체 자료를 반으로 나누는 경계값
  - 자료의 개수(n)가 홀수 : n=11이면 6번째 값
  - 자료의 개수(n)가 짝수 : n=10이면 5번째값과 6번째 값의 평균

※ 평균, 중앙값의 비교

표본평균은 가장 많이 쓰이는 중심위치 측도이지만, 이상치(크거나 작은값)에 민감하게 반응함.
반면, 중앙값은 이상치에는 강하나 자료 전체를 이용하지 않음. 따라서, 전체의 경향을 볼 때 극단적인 관측값의 영향을 배제하고 싶으면 중앙값이 바람직하고, 전체 관측값을 모두 포함하고 싶으면 평균을 사용하는 것이 바람직함.

- 사분위수(quartile)



- Q_1 : 제 25 백분위수, 제 1 사분위수 (first quartile)
- Q_2 : 제 50 백분위수, 제 2 사분위수 (second quartile) (= 중앙값)
- Q_3 : 제 75 백분위수, 제 3 사분위수 (third quartile)

▸ 산포 측도

- 산포(Spread), 변동성(Variability) : 자료가 얼마나 퍼져 있는가를 나타내주는 측도

- 분산(variance) : $s^2 = \frac{\text{편차의 제곱들의 합}}{(\text{자료의 개수})-1} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i-\bar{x})^2$

- 표준편차(standard deviation) : $s = \sqrt{s^2}$ (자료의 단위와 일치함)

※ 분산과 표준편차의 특징

- 퍼진 정도를 나타내는 가장 일반적인 측도이고 평균과 같이 모든 자료를 사용하기 때문에 이상치에 민감함
- 분산이 크면 자료가 평균값을 중심으로 광범위하게 분포되어 있다는 뜻이고, 분산이 작으면 평균값을 중심으로 조밀하게 분포되어 있다는 것을 의미

- 범위(range) : $R = x_{\max} - x_{\min}$
  - 계산은 편리하나 이상치에 민감

- 사분위수범위(IQR : Inter quartile Range) : IQR = 제 3 사분위수 - 제 1 사분위수 = $Q_3 - Q_1$
  - 중간 50%의 자료값들의 범위이며 이상치에 영향을 받지 않으나 이론적 전개의 어려움 때문에, 널리 쓰이지 않음.

▸ 요약통계량 : describe 함수(pandas 데이터프레임)

- 데이터프레임의 변수들 중 양적(숫자형)자료의 평균, 표준편차, 최소값, 최대값, 사분위수등 간단한 기초통계량을 요약하고 질적(문자형)자료에 대해서는 범주별 빈도수 등의 결과를 요약

`import seaborn as sns` `iris = sns.load_dataset('iris')`	
`iris.describe()`	▶ 데이터프레임의 양적(숫자형) 변수들에 대한 간단한 요약통계량 값 출력
`iris.describe(include='object')`	▶ include='object ': 데이터프레임의 질적(문자형) 변수에 대한 요약통계량 값 출력
`iris.describe(include='all')`	▶ include='all' : 양적(숫자형)변수, 질적(문자형) 변수 모두에 대한 요약통계량 값 출력
`iris.sepal_length.describe()`	▶ 지정된 변수(양적, 질적)에 대한 요약통계량 값 출력
`iris.species.describe()`	▶ 지정된 여러 개의 변수(양적, 질적)에 대하여 대한 요약통계량 값 출력
`iris[["sepal_length","petal_length"]].describe()`	

```
In [7]: import seaborn as sns
   ...: iris = sns.load_dataset('iris')

In [8]: iris.describe()
Out[8]:
       sepal_length  sepal_width  petal_length  petal_width
count    150.000000   150.000000    150.000000   150.000000
mean       5.843333     3.057333      3.758000     1.199333
std        0.828066     0.435866      1.765298     0.762238
min        4.300000     2.000000      1.000000     0.100000
25%        5.100000     2.800000      1.600000     0.300000
50%        5.800000     3.000000      4.350000     1.300000
75%        6.400000     3.300000      5.100000     1.800000
max        7.900000     4.400000      6.900000     2.500000

In [9]: iris.describe(include='object')
Out[9]:
        species
count       150
unique        3
top      setosa
freq         50
```

```
In [10]: iris.describe(include='all')
Out[10]:
        sepal_length  sepal_width  petal_length  petal_width species
count     150.000000   150.000000    150.000000   150.000000     150
unique           NaN          NaN           NaN          NaN       3
top              NaN          NaN           NaN          NaN  setosa
freq             NaN          NaN           NaN          NaN      50
mean        5.843333     3.057333      3.758000     1.199333     NaN
std         0.828066     0.435866      1.765298     0.762238     NaN
min         4.300000     2.000000      1.000000     0.100000     NaN
25%         5.100000     2.800000      1.600000     0.300000     NaN
50%         5.800000     3.000000      4.350000     1.300000     NaN
75%         6.400000     3.300000      5.100000     1.800000     NaN
max         7.900000     4.400000      6.900000     2.500000     NaN

In [11]: iris.sepal_length.describe()
Out[11]:
count    150.000000
mean       5.843333
std        0.828066
min        4.300000
25%        5.100000
50%        5.800000
75%        6.400000
max        7.900000
Name: sepal_length, dtype: float64
```

```
In [12]: iris.species.describe()
Out[12]:
count         150
unique          3
top        setosa
freq           50
Name: species, dtype: object

In [13]: iris[["sepal_length","petal_length"]].describe()
Out[13]:
       sepal_length  petal_length
count    150.000000    150.000000
mean       5.843333      3.758000
std        0.828066      1.765298
min        4.300000      1.000000
25%        5.100000      1.600000
50%        5.800000      4.350000
75%        6.400000      5.100000
max        7.900000      6.900000
```

▸ 다양한 기술통계량 함수(pandas 데이터프레임)

- 데이터프레임명.변수명.함수()를 사용하여 양적자료의 기술통계량을 구할 수 있다.

함수	기능	예
sum()	합계	In [59]: iris.sepal_length.sum() Out[59]: 876.5
mean()	평균	In [60]: iris.sepal_length.mean() Out[60]: 5.843333333333334
median()	중앙값	In [61]: iris.sepal_length.median() Out[61]: 5.8
min()	최소값	In [62]: iris.sepal_length.min() Out[62]: 4.3
max()	최대값	In [63]: iris.sepal_length.max() Out[63]: 7.9
quantile()	사분위수	In [64]: iris.sepal_length.quantile() Out[64]: 5.8
var()	분산	In [65]: iris.sepal_length.var() Out[65]: 0.6856935123042505
std()	표준편차	In [66]: iris.sepal_length.std() Out[66]: 0.8280661279778629
quantile(0.75)-quantile(0.25)	사분위수 범위	In [68]: iris.sepal_length.quantile(0.75) - iris.sepal_length.quantile(Out[68]: 1.3000000000000007

▸ 그룹별 요약통계량 계산

- groupby함수와 sum함수, mean함수, agg(aggregation)함수를 같이 사용하여 그룹(질적자료)별 일부 변수(양적자료)들에 대한 요약통계량 계산

`iris.groupby("species")["sepal_length"].sum()`	▶ iris 품종(species)별로 sepal_length의 합계를 출력
`iris.groupby("species")["sepal_width"].mean()`	▶ iris 품종(species)별로 sepal_width의 평균을 출력
`iris.groupby("species")["petal_width"].std()`	▶ iris 품종(species)별로 petal_width의 표준편차를 출력

`iris.groupby("species").agg({"sepal_length":"sum",`
`"sepal_width":"mean",`
`"petal_width":["min","max"]})`
	▶ species의 범주별로 sepal_length의 합계,
	sepal_width의 평균, petal_width의 최소값, 최대값을 출력

```
In [55]: iris.groupby("species")["sepal_length"].sum()
Out[55]:
species
setosa        250.3
versicolor    296.8
virginica     329.4
Name: sepal_length, dtype: float64

In [56]: iris.groupby("species")["sepal_width"].mean()
Out[56]:
species
setosa        3.428
versicolor    2.770
virginica     2.974
Name: sepal_width, dtype: float64

In [57]: iris.groupby("species")["petal_width"].std()
Out[57]:
species
setosa        0.105386
versicolor    0.197753
virginica     0.274650
Name: petal_width, dtype: float64
```

```
In [58]: iris.groupby("species").agg({"sepal_length":"sum",
    ...:                              "sepal_width":"mean",
    ...:                              "petal_width":
["min","max"]})
Out[58]:
           sepal_length sepal_width petal_width
                    sum        mean         min  max
species
setosa            250.3       3.428         0.1  0.6
versicolor        296.8       2.770         1.0  1.8
virginica         329.4       2.974         1.4  2.5
```

▸ 양적자료의 수치적 요약 : 넘파이(numpy) 함수 이용

함수	기능
`np.mean(x)`	평균
`np.var(x, ddof=1)`	분산
`np.std(x, ddof=1)`	표준편차
`np.median(x)`	중앙값
`np.percentile(x, p)`	p-백분위수, percentile(x, 50) : 50분위수
`np.quantile(x, q)`	q-분위수, quantile(x, 0.25) : 제1사분위수
`np.sum(x)`	합계
`np.min(x)`	최소값
`np.max(x)`	최대값

표와 그래프를 이용한 양적자료의 요약

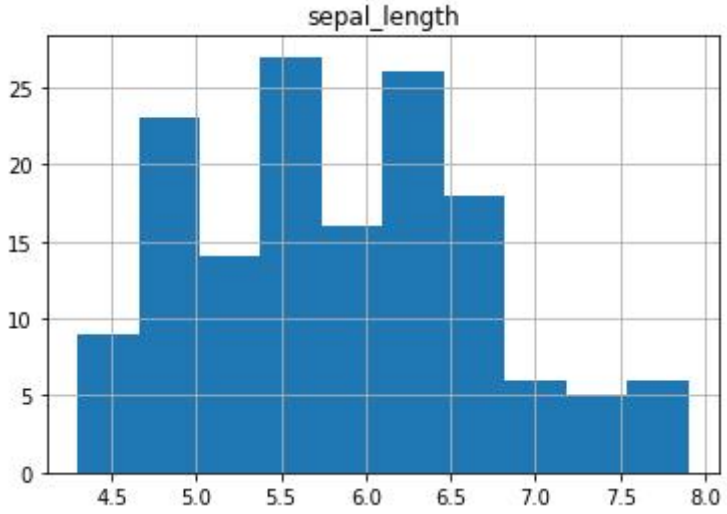
- **자료의 특성을 표, 그림, 통계량 등을 사용하여 쉽게 파악할 수 있도록 정리, 요약**

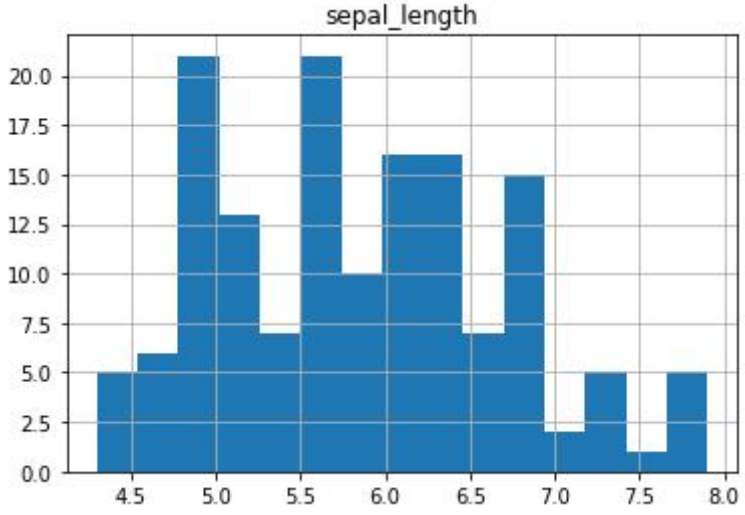
▸ 히스토그램(Histogram)

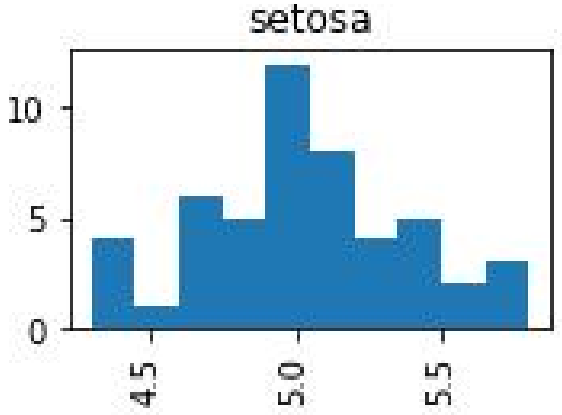
- hist 함수(pandas 데이터프레임)
- 일변량 자료의 분포를 알아보는데 유용한 그래프
- 양적(숫자형)자료에 대하여 적절한 구간으로 나누어 빈도와 퍼센트를 계산하고 이를 막대그래프의 형태로 표현

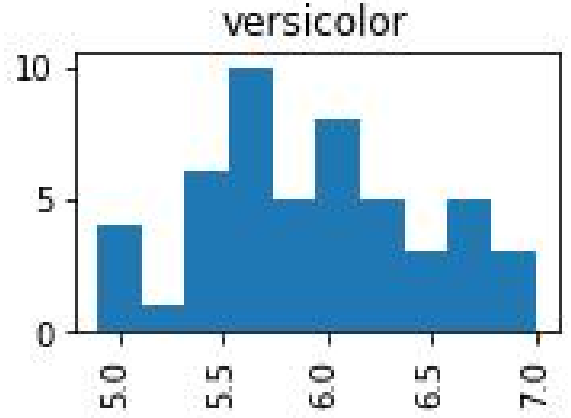
```
iris.hist(column="sepal_length")
iris.hist(column="sepal_length", bins=15)
iris.hist(column="sepal_length", by="species",
          grid=False)
```

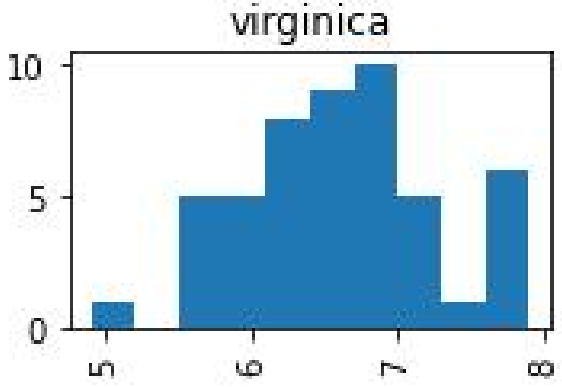
▶ iris의 sepal_length 변수에 대하여 히스토그램 출력
- column= : 히스토그램으로 출력할 변수 또는 변수 리스트 지정
- by = : 그룹변수 지정하여 그룹별 히스토그램을 그림
- bins= : 구간의 개수를 지정, 기본값은 bins=10
- grid=False : grid를 출력하지 않음


sepal_length
25
20
15
10
5
0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0


sepal_length
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0


setosa
10
5
0
4.5
5.0
5.5


versicolor
10
5
0
5.0
5.5
6.0
6.5
7.0


virginica
10
5
0
5
6
7
8

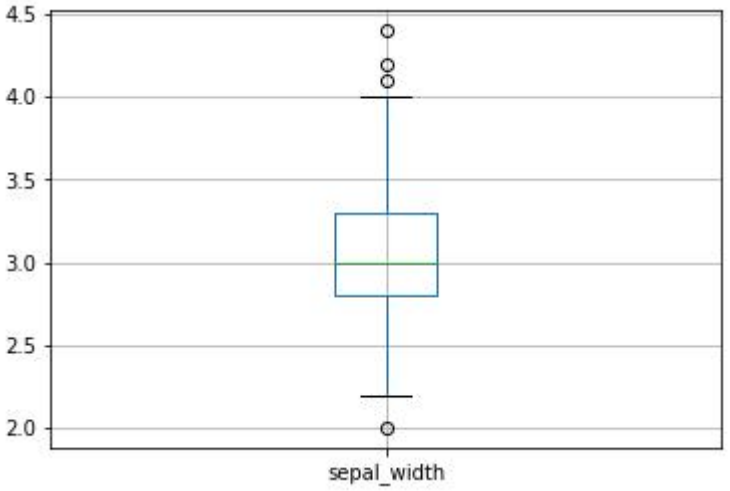
▸ 상자그림(Box plot)

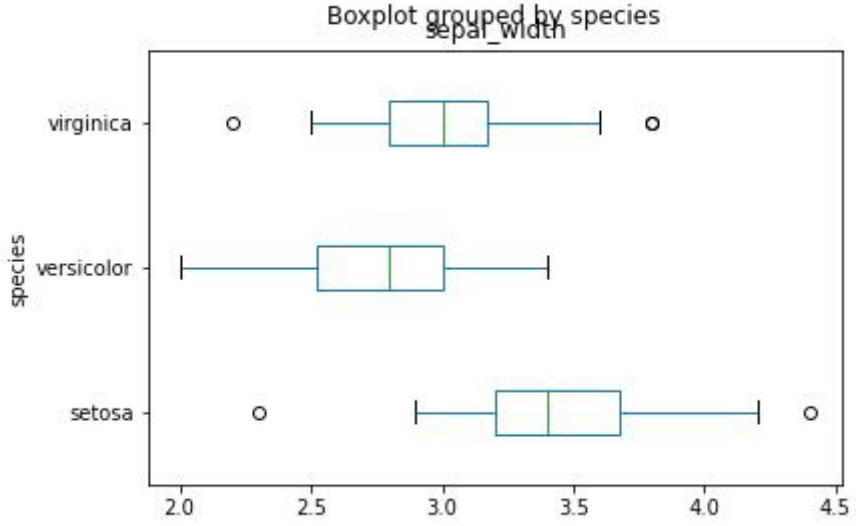
- boxplot 함수(pandas 데이터프레임)
- 다섯숫자요약그림 : 최소값, 제1사분위수, 제2사분위수(중앙값), 제3사분위수, 최대값
- 상자그림은 데이터의 분포를 보여주는 그림, 가운데 상자는 제 1사분위수, 중앙값, 제 3사분위수로 그림
- 자료의 산포도 및 대칭성의 정도를 눈으로 쉽게 파악할 수 있고 이상치의 유무도 알 수 있다.

```
iris.boxplot(column="sepal_width")

iris.boxplot(column="sepal_width", by="species",
             vert=False, grid =False)
```

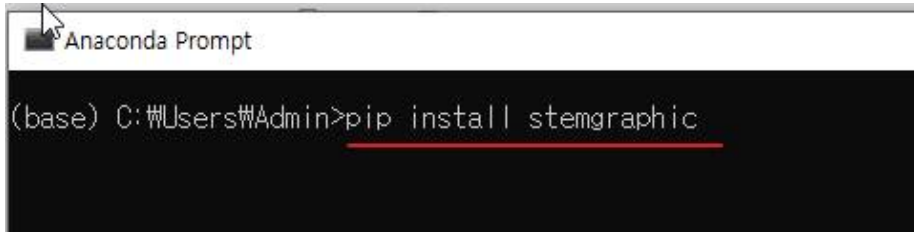
▶iris의 sepal_width 변수에 대하여 상자그림 출력
- column= : 변수 이름이나 변수들의 리스트 저장
- by= : 지정된 변수별로 상자그림을 그릴 경우 사용
- grid= : 격자의 출력 여부 지정 기본값 : grid = True
- vert=False : 수평상자그림 출력



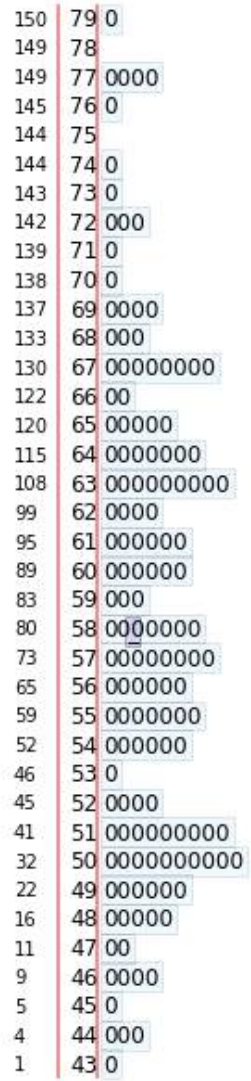


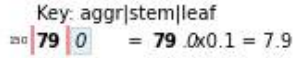
▸ 줄기-잎 그림(stem and leaf plot)

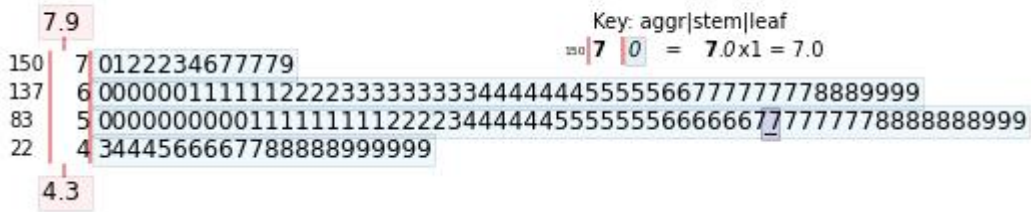
- stem_graphic함수(stemgraphic 라이브러리)
- 전체 자료를 display하는 그래프
- 줄기(stem) 부분에 일부 정보 제시, 잎(leaf) 부분에 나머지의 정보 제공
- anaconda 명령프롬프트 상에서 "pip install stemgraphic"을 실행



`from stemgraphic import stem_graphic`
 	
`stem_graphic(iris.sepal_length)`
 	
`stem_graphic(iris.sepal_length, scale=1)`
	▶ stemgraphic 라이브러리로부터 stem_graphic 함수를 import한다
	
	▶ iris.sepal_length의 줄기-잎 그림 그리기
	- array, list, df 등
	- scale=줄기의 크기


150 79 0
149 78
149 77 0000
145 76 0
144 75
144 74 0
143 73 0
142 72 000
139 71 0
138 70 0
137 69 0000
133 68 000
130 67 00000000
122 66 00
120 65 00000
115 64 0000000
108 63 000000000
99 62 0000
95 61 000000
89 60 000000
83 59 000
80 58 0000000
73 57 00000000
65 56 000000
59 55 0000000
52 54 000000
46 53 0
45 52 0000
41 51 000000000
32 50 0000000000
22 49 000000
16 48 00000
11 47 00
9 46 0000
5 45 0
4 44 000
1 43 0


Key: aggr|stem|leaf
150 79 0 = 79.0x0.1 = 7.9


7.9
150 7 0122234677779
137 6 000000111111222233333333344444445555566777777778889999
83 5 00000000001111111112222344444455555556666667777777888888999
22 4 3444566667788888999999
4.3
Key: aggr|stem|leaf
150 7 0 = 7.0x1 = 7.0

이변량 자료의 요약

■ 질적자료 요약

▸ 빈도표(분할표)

- crosstab함수(pandas 데이터프레임)
- 질적(문자형) 자료가 가지는 분포상의 특징이나 두 질적 자료간의 연관관계를 살펴보기 위하여 빈도표 사용

```
import pandas as pd
cars93 = pd.read_csv("d:/data/cars93.csv")

pd.crosstab(index=cars93["Origin"],columns=cars93["Type"])

pd.crosstab(index=cars93["Origin"],columns=cars93["Type"],
            margins=True, margins_name = "합계")

pd.crosstab(index=cars93["Origin"],columns=cars93["Type"],
            margins=True, margins_name="합계",
            normalize="all")
```

▶ cars93의 "Origin"변수와 "Type"변수에 대한 2차원 빈도표 작성

- index= : 행(row) 변수 지정
- columns= : 열(column) 변수 지정,
- margins= : 주변합 계산 출력 여부 지정, 기본값은 False
- normalize=: 비율 출력 옵션지정 ("all": 전체비율, index" : 행비율, "columns": 열비율) , 기본값은 False

```
In [77]: import pandas as pd
    ...: cars93 = pd.read_csv("d:/data/cars93.csv")

In [78]: pd.crosstab(index=cars93["Origin"],
    ...:             columns=cars93["Type"])
Out[78]:
Type     Compact  Large  Midsize  Small  Sporty  Van
Origin
USA            7     11       10      7       8    5
non-USA        9      0       12     14       6    4
```

```
In [79]: pd.crosstab(index=cars93["Origin"],
    ...:             columns=cars93["Type"],
    ...:             margins=True, margins_name="합계")
Out[79]:
Type     Compact  Large  Midsize  Small  Sporty  Van  합계
Origin
USA            7     11       10      7       8    5  48
non-USA        9      0       12     14       6    4  45
합계           16     11       22     21      14    9  93
```

```
In [80]: pd.crosstab(index=cars93["Origin"],
    ...:             columns=cars93["Type"],
    ...:              margins=True, margins_name="합계",
    ...:              normalize="all")
Out[80]:
Type      Compact    Large   Midsize     Small    Sporty       Van        합계
Origin
USA      0.075269  0.11828  0.107527  0.075269  0.086022  0.053763  0.516129
non-USA  0.096774  0.00000  0.129032  0.150538  0.064516  0.043011  0.483871
합계       0.172043  0.11828  0.236559  0.225806  0.150538  0.096774  1.000000
```

▸ 모자이크 도표

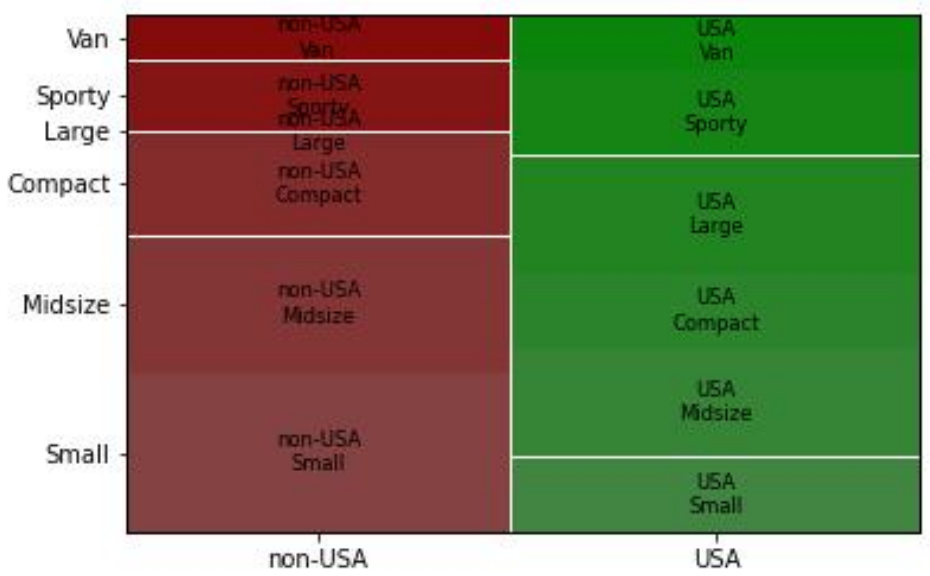
- mosaic 함수
- 다차원분할표에 대하여 각 cell별 크기를 그래프로 나타낸 것

```
from statsmodels.graphics.mosaicplot import mosaic

mosaic(cars93,['Origin','Type'])
```

▶ statsmodels.graphics.mosaicplot에서 mosaic 함수 import

▶ cars93의 'Origin', 변수와 'Type' 변수에 대하여 모자이크 도표 작성
- data : 대상객체(table, dict, Series, ndarray, DataFrame)지정
- index = : 범주형 변수의 순서 지정, 데이터프레임의 열 지정
- title = : 축의 제목 지정
- axes_label= : 축에서 각 범주 이름의 출력여부 지정



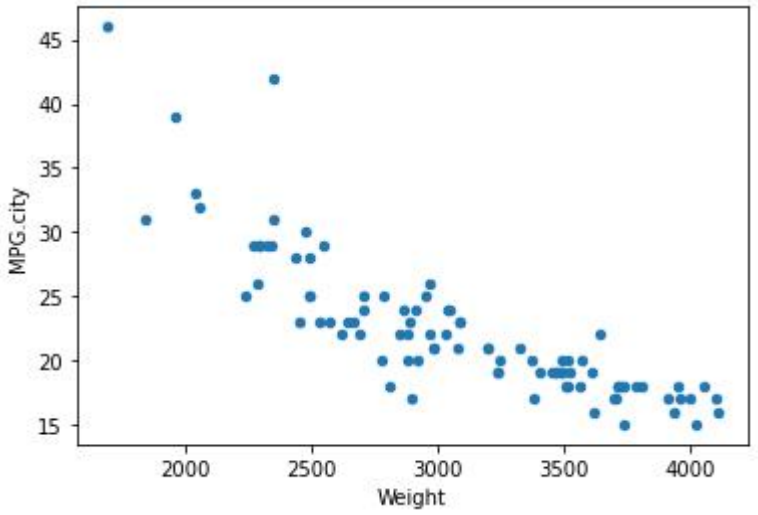
■ 양적자료 요약

▸ 산점도(scatter plot)

- DataFrame.plot함수(pandas 데이터프레임)

```
cars93.plot(x='Weight',y='MPG.city',kind='scatter')
```

▶ cars93의 Weight와 MPG.city 간의 산점도 작성
- x : x축 변수, y : y축 변수
- kind = : 생성할 플롯의 종류를 지정
  - 'line' : line plot(기본값, 선도표)
  - 'bar' : vertical bar chart(막대도표)
  - 'hist' : histogram(히스토그램)
  - 'scatter' : scatter plot(산점도)



▸ 상관계수를 이용한 요약

- corrcoef함수(numpy패키지)

- 상관계수는 두 변수간의 선형관계를 나타내는 측도 : $r = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}}$

- $-1 \le r \le 1$, 상관계수의 값이 0에 가까우면 두 변수 사이의 선형관계는 성립하지 않음
  - 상관계수>0 : 양의 상관관계, 상관계수<0 : 음의 상관관계
- 두 변수를 이용하여 산점도를 그리면 기울기가 0이 아닌 한 직선 주위에 모집단의 분포가 집중될수록 상관계수의 값은 그 직선의 방향에 따라 양극단값 -1, +1에 가까움

```
import numpy as np

np.corrcoef(cars93["Weight"],cars93["MPG.city"])
```

▶ cars93의 Weight와 MPG.city 사이의 상관계수 구하기

```
In [61]: import numpy as np
    ...:
    ...: np.corrcoef(cars93["Weight"],cars93["MPG.city"])
Out[61]:
array([[ 1.        , -0.84313855],
       [-0.84313855,  1.        ]])
```

통계적 추론

추정과 가설검정

■ 점추정

▸ 점추정(point estimation) : 하나의 모수를 한 개의 값으로 추정하는 것

▸ 구간추정(interval estimation)

- 모수를 하나의 값으로 추정하는 것이 아니라 추정량의 표준오차의 크기를 반영한 구간으로 나타내는 것
- 신뢰구간의 예 : 정부의 이번 정책에 대해 표본조사를 실시한 결과 찬성한다는 비율이 74%이었다. 이번 조사의 신뢰수준은 95%이고 오차 한계는 ±4%이다. ⇒ 점추정값 : 74%, 신뢰구간 : (70%, 78%)

■ 가설검정의 용어

- 대립가설 : 표본으로부터 입증하고자 하는 가설
- 귀무가설 : 대립가설에 대한 확실한 근거가 없을 때 받아들이는 가설
- 검정통계량 : 검정에 사용하는 통계량
- 유의수준(significance level) : 귀무가설 H_0가 참일 때 대립가설 H_1을 채택하는 오류를 범할 최대 허용 확률

  (즉, 제1종의 오류를 범할 확률)
- 유의확률(significance probability) : 유의확률이 작을수록 대립가설 H_1이 참이라는 증거가 강함을 뜻하며 유의확률이 유의수준보다 작으면 귀무가설을 기각

▸ 검정오류(test error)

- 모수는 미지의 수이므로, 귀무가설 H_0, 대립가설 H_1중 어느 하나가 옳은가를 수리적으로 증명할 수 없음
- 가설검정의 결과, 귀무가설H_0와 대립가설 H_1중 한 가설을 선택하면 반드시 그 결정에는 두 가지 오류를 범할 수 있음

<표>제1종 오류와 제2종 오류

미지의 실제 현상 / 검정결과	H_0이 맞다	H_1이 맞다
H_0 채택(H_1기각)	옳은 결정	잘못된 결론(제2종 오류)
H_0 기각(H_1채택)	잘못된 결론 (제1종 오류)	옳은 결정

한 모평균에 대한 추론 : 일표본 t-검정

▸ **주어진 표본의 정보를 이용하여 모집단의 평균에 대한 검정을 통해 모집단의 분포에 대하여 추론**

▸ **귀무가설 $H_0 : \mu = \mu_0$ 에 대하여**

- 검정통계량 $T = \dfrac{\overline{X} - \mu_0}{S/\sqrt{n}} (\sim t(n-1) \ \ under \ \ H_0)$

  검정통계량의 관측값 계산 $t = \dfrac{\overline{X} - \mu_0}{S/\sqrt{n}} (\sim t(n-1) \ \ under \ \ H_0)$

- 유의수준 α에서 기각역과 유의확률

대립가설	유의수준 α에서 기각역	유의확률
$H_1 : \mu > \mu_0$ (오른쪽 단측검정)	$t \geq t_\alpha(n-1)$	$P = P(T \geq t)$
$H_1 : \mu < \mu_0$ (왼쪽 단측검정)	$t \leq -t_\alpha(n-1)$	$P = P(T \leq t)$
$H_1 : \mu \neq \mu_0$ (양측검정)	$\lvert t \rvert \geq t_{\alpha/2}(n-1)$	$P = P(\lvert T \rvert \geq \lvert t \rvert)$

- 검정통계량의 관측값 t가 주어진 유의수준 α에서의 기각역에 포함되거나, 유의확률(P-값)이 유의수준 α보다 작으면 귀무가설 H_0를 기각한다.(즉, 대립가설 H_1을 채택한다.)

- 다음과 같이 시본(seaborn)패키지에 있는 tips 데이터프레임의 tip의 모평균이 2.8달러 정도 될 것인지 가설검정을 통해 유의수준 $\alpha = 0.05$에서 확인하시오.

변수	설명
total_bill	총매출액(달러)
tip	팁(달러)
sex	성별(Female, Male)
smoker	손님의 흡연여부 (No, Yes)_
day	요일(Thur, Fri, Sat, Sun)
time	식사시간(Lunch, Dinner)
size	식사인원

- 귀무가설(H_0) : $\mu = 2.8$(tip의 모평균은 2.8달러이다) vs 대립가설(H_1) : $\mu \neq 2.8$(tip의 모평균은 2.8달러가 아니다)
- ttest_1samp 함수(scipy 패키지의 stats 모듈) : 일표본 t-검정

```
import seaborn as sns
tips = sns.load_dataset("tips")

from scipy.stats import ttest_1samp
ttest_1samp(tips["tip"], popmean = 3)
```

▶ tips 데이터프레임의 tip변수에 대한 양측검정 실시
- a : 검정을 원하는 자료
- popmean : 귀무가설에서 지정된 모수값
- alternative= : "two-sided"(양측검정, 기본값),
  "less" (왼쪽 단측검정),
  "greater" (오른쪽 단측검정)
- <실행결과> statistic= : t-검정통계량 값,
  pvalue= : 유의확률, df= : 자유도

```
In [69]: import seaborn as sns
    ...: tips = sns.load_dataset("tips")

In [70]: from scipy.stats import ttest_1samp
    ...: ttest_1samp(tips["tip"], popmean = 2.8)
Out[70]: TtestResult(statistic=2.2384552191440736, pvalue=0.02609755920765946, df=243)
```

▶ 가설 : H_0 : $\mu = 2.8$(tip의 모평균은 2.8달러이다) vs H_1 : $\mu \neq 2.8$(tip의 모평균은 2.8달러가 아니다)

검정통계량 : $T = \dfrac{\overline{X} - \mu_0}{S/\sqrt{n}} \quad (\sim t(243)\ under\ H_0)$

검정통계량의 관측값 계산 : $t = \dfrac{\overline{X} - \mu_0}{S/\sqrt{n}} = 2.2385$

유의확률$(p-value)$: $P = P(|T| \geq 2.2385) = 0.026$

유의확률 0.026은 주어진 유의수준 $\alpha = 0.05$보다 작기 때문에 귀무가설을 기각할 수 있다.
따라서 유의수준 $\alpha = 0.05$에서 tip의 모평균은 2.8달러 정도라고 말할 수 없다.

- 예. 어느 대학의 신입생 가운데 랜덤하게 15명을 뽑아 심리검사를 실시한 결과 책임감에 대한 점수가 다음 표와 같았다. 이 대학 신입생의 책임감에 대한 평균점수가 40보다 높다라고 할 수 있는지 유의수준 $\alpha = 0.05$에서 검정해보자.

22	25	34	35	41	41	46	46	46	47	49	54	54	59	60

주어진 문제에 대한 가설을 다음과 같다. μ : 책임감 점수의 모평균이라 하면,

$$H_0 : \mu = 40 \qquad H_1 : \mu > 40$$

```
import numpy as np
from scipy.stats import ttest_1samp

score = np.array([22, 25, 34, 35, 41, 41, 46,
                  46, 46, 47, 49, 54, 54, 59, 60])

ttest_1samp(score, popmean = 40, alternative="greater")
```

두 모집단에 관한 비교 : 독립표본 t-검정

▸ **주어진 표본의 정보를 이용하여 두 모집단의 평균의 차이($\mu_1 - \mu_2$)에 대한 검정을 통해 두 모집단의 분포 비교**

① 귀무가설 $H_0 : \mu_1 - \mu_2 = \delta_0$의 검정법(공통분산 가정)

- 검정통계량 : $T = \dfrac{(\overline{X_1} - \overline{X_2}) - (\mu_1 - \mu_2)}{\sqrt{S_p^2(\dfrac{1}{n_1} + \dfrac{1}{n_2})}}$ $(\sim t(n_1 + n_2 - 2)\ under\ H_0)$

  검정통계량의 관측값 계산 : $t = \dfrac{(\overline{x_1} - \overline{x_2}) - (\mu_1 - \mu_2)}{\sqrt{s_p^2(\dfrac{1}{n_1} + \dfrac{1}{n_2})}}$, $S_p^2 = \dfrac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2}{n_1 + n_2 - 2}$

- 유의수준 α에서 기각역과 유의확률

대립가설	유의수준 α에서 기각역	유의확률
$H_1 : \mu_1 - \mu_2 > 0$ (오른쪽 단측검정)	$t \geq t_\alpha(n_1 + n_2 - 2)$	$P = P(T \geq t)$
$H_1 : \mu_1 - \mu_2 < 0$ (왼쪽 단측검정)	$t \leq -t_\alpha(n_1 + n_2 - 2)$	$P = P(T \leq t)$
$H_1 : \mu_1 - \mu_2 \neq 0$ (양측검정)	$\lvert t \rvert \geq t_{\alpha/2}(n_1 + n_2 - 2)$	$P = P(\lvert T \rvert \geq \lvert t \rvert)$

- 검정통계량의 관측값 t가 주어진 유의수준 α에서의 기각역에 포함되거나, 유의확률(P-값)이 유의수준 α보다 작으면 귀무가설 H_0를 기각한다.(즉, 대립가설 H_1을 채택한다.)

② 귀무가설 $H_0 : \mu_1 - \mu_2 = \delta_0$의 검정법(이분산 가정)

- 검정통계량 : $T = \dfrac{(\overline{X_1} - \overline{X_2}) - (\mu_1 - \mu_2)}{\sqrt{\dfrac{S_1^2}{n_1} + \dfrac{S_2^2}{n_2}}}$ $(\sim t(df)\ \ under\ \ H_0)$

  검정통계량의 관측값 계산 : $t = \dfrac{(\overline{x_1} - \overline{x_2}) - (\mu_1 - \mu_2)}{\sqrt{\dfrac{s_1^2}{n_1} + \dfrac{s_2^2}{n_2}}}$

- 유의수준 α에서 기각역과 유의확률

대립가설	유의수준 α에서 기각역	유의확률
$H_1 : \mu_1 - \mu_2 > 0$ (오른쪽 단측검정)	$t \geq t_\alpha(df)$	$P = P(T \geq t)$
$H_1 : \mu_1 - \mu_2 < 0$ (왼쪽 단측검정)	$t \leq -t_\alpha(df)$	$P = P(T \leq t)$
$H_1 : \mu_1 - \mu_2 \neq 0$ (양측검정)	$\lvert t \rvert \geq t_{\alpha/2}(df)$	$P = P(\lvert T \rvert \geq \lvert t \rvert)$

- 검정통계량의 관측값 t가 주어진 유의수준 α에서의 기각역에 포함되거나, 유의확률(P-값)이 유의수준 α보다 작으면 귀무가설 H_0를 기각한다.(즉, 대립가설 H_1을 채택한다.)

- 예.(paint.txt) 한 페인트 제조회사에서는 새로운 유성페인트를 개발하여 기존의 페인트와의 건조속도를 비교하고자 한다. 이를 확인하기 위해 시중에서 가장 인기 있는 제품과 새 제품을 각각 5종류의 벽에 칠한 후 건조시간을 측정하였다. 이 자료들은 서로 독립된 두 모집단에서 추출된 표본이다. paint.txt는 그룹을 나타내는 변수(group 1=인기제품, 2=새 제품)와 검정 대상이 되는 변수 (time, 건조시간)로 구성되어 있다.
  새로 개발한 페인트 제품은 기존의 제품보다 건조시간이 더 빠르다고 할 수 있는가? 유의수준 $\alpha = 0.05$에서 검정해보자.

	건조시간(단위 : 분)									
기존 제품	49	44	47	44	46	40	48	45	45	42
새 제품	44	41	45	44	43	39	42	40	40	42

- 가설설정 : 기존 제품의 건조시간 모평균을 μ_1, 새 제품의 건조시간 모평균을 μ_2라고 하자

$$H_0 : \mu_1 - \mu_2 = 0 \qquad H_1 : \mu_1 - \mu_2 > 0$$

- 검정하기 전에 두 표본 자료의 평균과 분산을 구해서 비교

```
import pandas as pd
paint = pd.read_table("d:/data/paint.txt", sep = " ")

paint.groupby("group")["time"].mean()
paint.groupby("group")["time"].std()
```

▶ groupby() 함수를 이용하여 기존페인트와 새페인트의 건조시간의 평균과 분산 비교
-() 안에는 범주형 변수

```
In [71]: import pandas as pd
    ...: paint = pd.read_table("d:/data/paint.txt", sep = " ")

In [72]: paint.groupby("group")["time"].mean()
Out[72]:
group
1    45.0
2    42.0
Name: time, dtype: float64

In [73]: paint.groupby("group")["time"].std()
Out[73]:
group
1    2.708013
2    2.000000
Name: time, dtype: float64
```

▶ 기존 페인트(1)와 새페인트(2) 건조시간의 평균비교
비교 : 새 페인트의 건조시간이 좀 더 짧다.

▶ 기존 페인트(1)와 새페인트(2) 건조시간의 표준편차 비교 : 두 제품간의 차이가 크게 없다.

- ttest_ind 함수(scipy패키지 stats 모듈) : 독립 이표본 검정

```
group1 = paint[paint['group'] == 1]
group2 = paint[paint['group'] == 2]


from scipy.stats import ttest_ind

ttest_ind(group1.time,group2.time,  alternative="greater",
          equal_var = True)
```

▶ 기존페인트와 새페인트 데이터를 분리

▶ ttest_ind(a, b, equal_var=, alternative=)
- 두 독립표본에 대한 t-검정 수행,
- a=, b= : 검정을 수행할 두 변수 값을 지정
- equal_val= : True (공통분산 가정), False(이분산 가정)
- alternative= : "two-sided"(양측검정, 기본값),
  "less" (왼쪽 단측검정),
  "greater" (오른쪽 단측검정)
- <실행결과> statistic= : t-검정통계량 값,
  pvalue= : 유의확률, df= : 자유도

서로 독립인 두 표본의 평균 검정의 결과는 다음과 같다.

```
In [16]: group1 = paint[paint['group'] == 1]
    ...: group2 = paint[paint['group'] == 2]

In [17]: from scipy.stats import ttest_ind

In [18]: ttest_ind(group1.time, group2.time,
    ...:               alternative="greater",equal_var = True)
Out[18]: TtestResult(statistic=2.8180093098831724, pvalue=0.00569415182464645, df=18.0)
```

▶ μ_1 : 기존 페인트의 건조시간이 평균 , μ_2 : 새 페인트의 건조시간의 평균

가설 $H_0 : \mu_1 - \mu_2 = 0 \quad H_1 : \mu_1 - \mu_2 > 0$

검정통계량 : $T = \dfrac{(\overline{X_1} - \overline{X_2}) - (\mu_1 - \mu_2)}{S_p \sqrt{\dfrac{1}{n_1} + \dfrac{1}{n_2}}} \quad (\sim t(18) \ under \ H_0)$

검정통계량의 관측값은 $t = 2.818$이고 pvalue=0.005694이다.

검정결과, 유의확률이 0.005694로 유의수준 $\alpha = 0.05$보다 작으므로 귀무가설을 기각할 수 있다.
따라서 새 페인트의 건조시간은 기존 페인트의 건조시간보다 짧다고 할 수 있다.

- 예. 다음은 두 집단에서 조사한 체질량 지수의 자료이다. 집단별로 체질량지수는 차이가 있다고 볼 수 있는가? 유의수준 $\alpha = 0.05$에서 검정해보자.

	체질량 지수										
그룹1	22	23	25	26	27	19	22	28	33	24	
그룹2	21	25	36	24	33	28	29	31	30	32	33	35

```
import numpy as np

group1 = np.array([22, 23, 25, 26, 27, 19, 22, 28, 33, 24])
group2 = np.array([21, 25, 36, 24, 33, 28, 29, 31, 30, 32, 33, 35])

group1.mean()
group2.mean()
group1.std()
group2.std()


from scipy.stats import ttest_ind

ttest_ind(group1, group2, equal_var = True )
```

대응비교에 의한 모평균의 비교

▸ **대응비교 또는 쌍체비교 (paired comparison) : 실험단위를 동질적인 쌍으로 묶고, 각 쌍에 두 처리를 랜덤하게 적용한 다음, 각 쌍에서 얻은 관측값의 차로 두 모평균의 차($\mu_1 - \mu_2 = \mu_D$)에 대하여 추론**

- 귀무가설 $H_0 : \mu_D = 0$에 대하여
- 검정통계량 : $T = \dfrac{\overline{D} - \delta_0}{S_D/\sqrt{n}}$ $(\sim t(n-1)\ under\ H_0)$

  검정통계량의 관측값 계산 : $t = \dfrac{\overline{d} - \mu_0}{s_D/\sqrt{n}}$
- 유의수준 α에서 기각역과 유의확률

대립가설	유의수준 α에서 기각역	유의확률
$H_1 : \mu_D > 0$	$t \geq t_\alpha(n-1)$	$P = P(T \geq t)$
$H_1 : \mu_D < 0$	$t \leq -t_\alpha(n-1)$	$P = P(T \leq t)$
$H_1 : \mu_D \neq 0$	$\lvert t \rvert \geq t_{\alpha/2}(n-1)$	$P = P(\lvert T \rvert \geq \lvert t \rvert)$

- 검정통계량의 관측값 t가 주어진 유의수준 α에서의 기각역에 포함되거나, 유의확률(P-값)이 유의수준 α보다 작으면 귀무가설 H_0를 기각한다.(즉, 대립가설 H_1을 채택한다.)

- 예. 어떤 약의 부작용으로 혈압강하의 효과가 있는지 알아보기 위하여 15명의 환자를 대상으로 약의 복용 전후의 이완기 혈압을 측정하였더니 그 결과가 다음과 같았다.

환자	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
복용전	70	80	72	76	76	76	72	78	82	64	74	92	74	68	84
복용후	68	72	62	70	58	66	68	52	64	72	74	60	74	72	74

이 자료로부터 약이 혈압을 내린다는 주장을 할 수 있는지 유의수준 $\alpha = 0.05$에서 검정하시오.

μ_1 : 약 복용 전 이완기 혈압의 모평균, μ_2 : 약 복용 후 이완기 혈압의 모평균, $\mu_1 - \mu_2 = \mu_D$라 하면

- 가설설정 $H_0 : \mu_D = 0$ $H_1 : \mu_D > 0$

- ttest_rel 함수(scipy 패키지의 stats 모듈) : 대응비교

```
import numpy as np
from scipy.stats import ttest_rel

before = np.array([70,80,72,76,76,76,72,78,82,
                   64,74,92,74,68,84])
after=  np.array([68,72,62,70,58,66,68,52,
                   64,72,74,60,74,72,74])
ttest_rel(before, after, alternative="greater" )
```

▶ ttest_rel(a, b, alternative=)(scipy패키지)
- 대응표본에 대한 t-검정 수행, 검정통계량과 유의확률이 출력
- a, b,: 검정을 수행할 두 변수 값을 지정

- alternative= : "two-sided"(양측검정, 기본값)
  "less"(왼쪽단측검정)
  "greater"(오른쪽단측검정),

- <실행결과> t-검정통계량 값과 유의확률 출력

```
In [29]: from scipy.stats import ttest_rel

In [30]: before = np.array([70,80,72,76,76,76,72,78,82,
    ...:                    64,74,92,74,68,84])
    ...: after=  np.array([68,72,62,70,58,66,68,52,
    ...:                    64,72,74,60,74,72,74])

In [31]: ttest_rel(before, after, alternative="greater" )
Out[31]: TtestResult(statistic=3.105360487466109, pvalue=0.0038747180533270594, df=14)
```

▶ 가설 : $H_0 : \mu_D = 0$ $H_1 : \mu_D \neq 0$

검정통계량 : $T = \dfrac{\overline{D} - \delta_0}{S_D/\sqrt{n}}$ $(\sim t(14)\ under\ H_0)$

검정 결과, $t-$검정통계량의 관측값은 3.1054이고 유의확률은 0.003875이다.

유의확률 유의확률은 0.003875는 주어진 유의수준 $\alpha = 0.05$보다 작으므로 귀무가설을 기각할 수 있다.

따라서 유의수준 $\alpha = 0.05$에서 약의 복용 후의 이완기 혈압의 평균이 낮아졌다. 따라서 약이 혈압을 내린다고 주장할 근거가 충분하다.

- 예.(textbooks.txt) 주어진 자료는 UCLA 내의 서점과 Amazon.com에서 판매되는 교재들의 가격에 대한 자료이다. 2010년 봄학기에 개설된 UCLA의 강의 중에서 73개를 선택하여 각 강의에 쓰이는 교재의 온라인 판매 가격(amazNew) 과 오프라인의 판매 가격(uclaNew)을 조사하였다. 교재의 판매가격은 판매 장소(온라인또는 오프라인)에 따라 차이가 난다고 볼 수 있는가? 적절한 가설을 세우고 유의수준 $\alpha = 0.05$에서 이를 검정하시오.

```
iimport pandas as pd
textbooks=pd.read_table("d:/data/textbooks.txt", sep = " ")

ttest_rel(textbooks.uclaNew, textbooks.amazNew)
```

세 개이상의 모평균에 관한 비교 : 분산분석

▸ **두 모평균의 차에 대한 검정의 확장으로서, 3개 이상의 처리효과나 모평균의 비교를 위한 검정 방법**
: 어떤 속성이 여러 가지 조건에 처해 있을 때 특성값이 어떻게 차이가 나는지 관심

- 가설 $H_0 : a_1 = a_2 = \ldots = a_k = 0$ vs H_1 : 적어도 한 a_i는 0은 아니다.
- 검정통계량 $F = \dfrac{MS_{tr}}{MSE}$ $\quad (\sim F(k-1, n-k)\ under\ H_0)$

  검정통계량의 관측값 : $f = \dfrac{MS_{tr}}{MSE}$
- 유의수준 α의 기각역과 유의확률

유의수준 α의 기각역	유의확률
$f \geq F_\alpha(k-1, N-k)$	$P = P(F \geq f),\ F \sim F(k-1, N-k)$

- 유의확률(P-값)이 유의수준 α보다 작거나, 검정통계량의 관측값 f가 주어진 유의수준 α에서 기각역에 포함되면 귀무가설 H_0를 기각한다.(즉, 대립가설 H_1을 채택한다.)

- 예. tips 데이터프레임에서 요일(day)에 따라 레스토랑의 총매출액(total_bill)에 차이가 있는지 유의수준 $\alpha = 0.05$에서 검정하시오.

  - 주어진 자료는 일원배치분산분석을 적용할 수 있으며 검정하고자 하는 가설은 다음과 같다.
    α_i : 총매출액에 대한 요일(day)별 효과

    $H_0 : \alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = 0$(요일별 효과가 없다) vs H_1 : 적어도 한 α_i는 0은 아니다.

  - ols.fit(statmodels패키지) : 분산분석 시행
    만약 인자변수가 숫자형이라면 분산분석을 시행하기 전에 숫자형 변수를 요인(factor)으로 변화 : ols 함수 안에서 'C'명령어를 사용하여 변환
    ols.fit()결과를 model1에 저장하고 분산분석표는 statmodels.api.stats 모듈에서 anova_lm 함수를 사용하여 작성

```
from statsmodels.formula.api import ols
import statsmodels.api as sm

model = ols('total_bill ~ C(day)', data = tips).fit()


table = sm.stats.anova_lm(model)
print(table)
```

▶ 요일별로 총매출액에 차이가 있는지 분산분석 실시
ols(formula).fit()
- 설명변수는 범주형 자료여야 함

▶ anova_lm() : (statsmodel패키지)
적합된 선형모형에 대한 분산분석표를 출력

```
In [44]: from statsmodels.formula.api import ols
    ...: import statsmodels.api as sm

In [45]: model1 = ols('total_bill ~ day', data = tips).fit()
    ...: table1 = sm.stats.anova_lm(model1)

In [46]: print(table1)
               df        sum_sq     mean_sq          F    PR(>F)
day           3.0    643.941362  214.647121   2.767479  0.042454
Residual    240.0  18614.522721   77.560511        NaN       NaN
```

▶ 처리효과의 유의성 검정

α_i : 총매출액에 대한 요일(day)별 효과

$H_0 : a_1 = a_2 = a_3 = a_4 = 0$ vs H_1 : 적어도 한 a_i는 0은 아니다.

검정통계량 $F = \frac{MS_{tr}}{MSE} \sim F(3,240)\ under\ H_0$

검정 결과, 검정통계량의 관측값 $f = 2.7675$이고 유의확률은 0.042454로 유의수준 $\alpha = 0.05$보다 작으므로 귀무가설을 기각할 수 있다.
즉, 요일별로 총매출액은 차이가 있다고 할 수 있다.

- 예. 어떤 직물의 가공시 처리액의 농도가 직물의 인장강도에 영향을 미치는 지의 여부를 조사하기 위해, 네 가지 농도 $A_1 : 3.0\%,\ A_2 : 3.5\%,\ A_3 : 4.0\%,\ A_4 : 4.5\%$에서 반복 각 5회, 총 20회를 핸덤하게 처리한 후 인장강도를 측정한 결과가 아래 표와 같다. 이 자료에 대하여 일원배치법의 모형을 적용할 때, 농도에 따른 인장강도에 차가 있는지를 유의수준 $\alpha = 0.05$에서 검정해보자.

A_1	A_2	A_3	A_4
47	51	50	22
58	62	38	23
51	31	47	28
61	46	27	42
46	49	23	25

```
rawdata = {'type': [1,1,1,1,1,2,2,2,2,2,3,3,3,3,3,4,4,4,4,4],
           'strength':[47,58,51,61,46,51,62,31,46,49,50,38,47,27,23,22,23,28,42,25]
           }

from pandas import DataFrame
y = DataFrame(rawdata)

from statsmodels.formula.api import ols
import statsmodels.api as sm

model1 = ols('strength ~ C(type)', data =y).fit()
table1 = sm.stats.anova_lm(model1)

print(table1)
```

감사합니다