Pandas 라이브러리 IRIS 데이터 셋 실습해보기

- 데이터 처리와 분석을 위한 파이썬 라이브러리
- R의 data.frame과 유사하게 설계한 DataFrame이라는 데이터 기반으로 만들어짐.
- 데이터 분석시에 속도도 빠르고 많은 기능을 가지고 있어, 머신러닝 데이터 분석을 수행시에 많이 사용됨.
- 데이터를 읽고, 쓰기가 비교적 용이함.

- 참조 url : https://pandas.pydata.org/ (https://pandas.pydata.org/)

학습 내용

- Iris데이터 셋을 이용한 다양한 데이터 처리를 알아보자.
  - 하나/둘이상의 열 선택
  - 조건(하나 또는 두개 이상)을 이용한 행 선택
  - [ ].unique() - 중복값 제외
  - [ ].value_counts() - 각 값의 빈도수 확인
  - 행의 index를 초기화 시키기 - [].reset_index()
  - 특징간 상관계수 확인하기 - [].corr()
  - 히트맵으로 상관계수 확인 - sns.heatmat()
  - 특징간 상관관계 확인 - sns.pairplot()

01. 데이터 준비

In [30]:

```
import pandas as pd
import seaborn as sns

print(pd.__version__)
iris = sns.load_dataset("iris")
iris
```

1.1.3

Out[30]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

02. 행,열 선택

In [31]:

```
print(iris.columns)

# sepal_length 열 선택
# sepal_width에서 petal_width열 선택
```

```
Index(['sepal_length', 'sepal_width', 'petal_length', 'petal_width',
       'species'],
      dtype='object')
```

In [32]:

```
# sepal_length 열 선택
iris['sepal_length']
```

Out[32]:

```
0      5.1
1      4.9
2      4.7
3      4.6
4      5.0
      ...
145    6.7
146    6.3
147    6.5
148    6.2
149    5.9
Name: sepal_length, Length: 150, dtype: float64
```

In [33]:

```
# sepal_width에서 petal_width열 선택 - (1) (다중 컬럼 선택)
iris[ ['sepal_length', 'petal_length', 'petal_width']   ]
```

Out[33]:

	sepal_length	petal_length	petal_width
0	5.1	1.4	0.2
1	4.9	1.4	0.2
2	4.7	1.3	0.2
3	4.6	1.5	0.2
4	5.0	1.4	0.2
...	...	...	...
145	6.7	5.2	2.3
146	6.3	5.0	1.9
147	6.5	5.2	2.0
148	6.2	5.4	2.3
149	5.9	5.1	1.8

150 rows × 3 columns

In [34]:

```
# sepal_width에서 petal_width열 선택 -
# (2) (다중 컬럼 선택) - [].loc[행전체선택 , 시작:끝]
iris.loc[ : , 'sepal_length':'petal_width'  ]
```

Out[34]:

	sepal_length	sepal_width	petal_length	petal_width
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
...	...	...	...	...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows × 4 columns

In [35]:

```
# width에 해당하는 컬럼만 반복을 통해 가져올 수 있음.
# : 전후로 생략 가능
iris.iloc[0:3, 0:2]
```

Out[35]:

	sepal_length	sepal_width
0	5.1	3.5
1	4.9	3.0
2	4.7	3.2

03. 중복값을 제외한 값 확인 - [ ].unique()

iris의 꽃의 종류 - 중복 제외하고 어떤 값이 있는지 확인할 수 있을까?

In [36]:

```
print(iris.columns)
```

```
Index(['sepal_length', 'sepal_width', 'petal_length', 'petal_width',
       'species'],
      dtype='object')
```

In [37]:

```
iris['species'].unique()
```

Out[37]:

```
array(['setosa', 'versicolor', 'virginica'], dtype=object)
```

04. 중복값을 제외한 값의 빈도수 확인 - [ ].value_counts()

In [38]:

```
print(iris.columns)
```

```
Index(['sepal_length', 'sepal_width', 'petal_length', 'petal_width',
       'species'],
      dtype='object')
```

In [39]:

```
iris['species'].value_counts()
```

Out[39]:

```
virginica     50
setosa        50
versicolor    50
Name: species, dtype: int64
```

05. 조건식을 이용하여 해당 하는 컬럼 가져오기

In [40]:

```
[x for x in iris.columns if 'sepal' in x]
```

Out[40]:

```
['sepal_length', 'sepal_width']
```

In [41]:

```
# width에 해당하는 컬럼만 반복을 통해 가져올 수 있음.
iris.loc[:3, [x for x in iris.columns if 'sepal' in x] ]
```

Out[41]:

	sepal_length	sepal_width
0	5.1	3.5
1	4.9	3.0
2	4.7	3.2
3	4.6	3.1

조건을 두고 versicolor 행만 추출해 보자.

In [42]:

```
#iris[ 조건식 ] => 조건에 만족하는 행 추출
iris[ iris['species']=='versicolor' ]
```

Out[42]:

	sepal_length	sepal_width	petal_length	petal_width	species
50	7.0	3.2	4.7	1.4	versicolor
51	6.4	3.2	4.5	1.5	versicolor
52	6.9	3.1	4.9	1.5	versicolor
53	5.5	2.3	4.0	1.3	versicolor
54	6.5	2.8	4.6	1.5	versicolor
55	5.7	2.8	4.5	1.3	versicolor
56	6.3	3.3	4.7	1.6	versicolor
57	4.9	2.4	3.3	1.0	versicolor
58	6.6	2.9	4.6	1.3	versicolor
59	5.2	2.7	3.9	1.4	versicolor
60	5.0	2.0	3.5	1.0	versicolor
61	5.9	3.0	4.2	1.5	versicolor
62	6.0	2.2	4.0	1.0	versicolor
63	6.1	2.9	4.7	1.4	versicolor
64	5.6	2.9	3.6	1.3	versicolor
65	6.7	3.1	4.4	1.4	versicolor
66	5.6	3.0	4.5	1.5	versicolor
67	5.8	2.7	4.1	1.0	versicolor
68	6.2	2.2	4.5	1.5	versicolor
69	5.6	2.5	3.9	1.1	versicolor
70	5.9	3.2	4.8	1.8	versicolor
71	6.1	2.8	4.0	1.3	versicolor
72	6.3	2.5	4.9	1.5	versicolor
73	6.1	2.8	4.7	1.2	versicolor
74	6.4	2.9	4.3	1.3	versicolor
75	6.6	3.0	4.4	1.4	versicolor
76	6.8	2.8	4.8	1.4	versicolor
77	6.7	3.0	5.0	1.7	versicolor
78	6.0	2.9	4.5	1.5	versicolor
79	5.7	2.6	3.5	1.0	versicolor
80	5.5	2.4	3.8	1.1	versicolor
81	5.5	2.4	3.7	1.0	versicolor
82	5.8	2.7	3.9	1.2	versicolor
83	6.0	2.7	5.1	1.6	versicolor

	sepal_length	sepal_width	petal_length	petal_width	species
84	5.4	3.0	4.5	1.5	versicolor
85	6.0	3.4	4.5	1.6	versicolor
86	6.7	3.1	4.7	1.5	versicolor
87	6.3	2.3	4.4	1.3	versicolor
88	5.6	3.0	4.1	1.3	versicolor
89	5.5	2.5	4.0	1.3	versicolor
90	5.5	2.6	4.4	1.2	versicolor
91	6.1	3.0	4.6	1.4	versicolor
92	5.8	2.6	4.0	1.2	versicolor
93	5.0	2.3	3.3	1.0	versicolor
94	5.6	2.7	4.2	1.3	versicolor
95	5.7	3.0	4.2	1.2	versicolor
96	5.7	2.9	4.2	1.3	versicolor
97	6.2	2.9	4.3	1.3	versicolor
98	5.1	2.5	3.0	1.1	versicolor
99	5.7	2.8	4.1	1.3	versicolor

In [43]:

```
# 조건을 만족하는 행 추출. loc 이용
iris.loc[ iris['species']=='versicolor' ]
```

Out[43]:

	sepal_length	sepal_width	petal_length	petal_width	species
50	7.0	3.2	4.7	1.4	versicolor
51	6.4	3.2	4.5	1.5	versicolor
52	6.9	3.1	4.9	1.5	versicolor
53	5.5	2.3	4.0	1.3	versicolor
54	6.5	2.8	4.6	1.5	versicolor
55	5.7	2.8	4.5	1.3	versicolor
56	6.3	3.3	4.7	1.6	versicolor
57	4.9	2.4	3.3	1.0	versicolor
58	6.6	2.9	4.6	1.3	versicolor
59	5.2	2.7	3.9	1.4	versicolor
60	5.0	2.0	3.5	1.0	versicolor
61	5.9	3.0	4.2	1.5	versicolor
62	6.0	2.2	4.0	1.0	versicolor
63	6.1	2.9	4.7	1.4	versicolor
64	5.6	2.9	3.6	1.3	versicolor
65	6.7	3.1	4.4	1.4	versicolor
66	5.6	3.0	4.5	1.5	versicolor
67	5.8	2.7	4.1	1.0	versicolor
68	6.2	2.2	4.5	1.5	versicolor
69	5.6	2.5	3.9	1.1	versicolor
70	5.9	3.2	4.8	1.8	versicolor
71	6.1	2.8	4.0	1.3	versicolor
72	6.3	2.5	4.9	1.5	versicolor
73	6.1	2.8	4.7	1.2	versicolor
74	6.4	2.9	4.3	1.3	versicolor
75	6.6	3.0	4.4	1.4	versicolor
76	6.8	2.8	4.8	1.4	versicolor
77	6.7	3.0	5.0	1.7	versicolor
78	6.0	2.9	4.5	1.5	versicolor
79	5.7	2.6	3.5	1.0	versicolor
80	5.5	2.4	3.8	1.1	versicolor
81	5.5	2.4	3.7	1.0	versicolor
82	5.8	2.7	3.9	1.2	versicolor
83	6.0	2.7	5.1	1.6	versicolor

	sepal_length	sepal_width	petal_length	petal_width	species
84	5.4	3.0	4.5	1.5	versicolor
85	6.0	3.4	4.5	1.6	versicolor
86	6.7	3.1	4.7	1.5	versicolor
87	6.3	2.3	4.4	1.3	versicolor
88	5.6	3.0	4.1	1.3	versicolor
89	5.5	2.5	4.0	1.3	versicolor
90	5.5	2.6	4.4	1.2	versicolor
91	6.1	3.0	4.6	1.4	versicolor
92	5.8	2.6	4.0	1.2	versicolor
93	5.0	2.3	3.3	1.0	versicolor
94	5.6	2.7	4.2	1.3	versicolor
95	5.7	3.0	4.2	1.2	versicolor
96	5.7	2.9	4.2	1.3	versicolor
97	6.2	2.9	4.3	1.3	versicolor
98	5.1	2.5	3.0	1.1	versicolor
99	5.7	2.8	4.1	1.3	versicolor

두개의 조건 - setosa 중에 sepal_length이 평균 이상인 것들만 추출해보기.

In [44]:

```
iris.sepal_length.mean()
```

Out[44]:

5.843333333333335

In [45]:

```
# iris[ (조건1) & (조건2) ]
iris_tmp = iris[ (iris['species']=='versicolor') &
                 (iris['sepal_length'] > 5.843 ) ]
iris_tmp
```

Out[45]:

	sepal_length	sepal_width	petal_length	petal_width	species
50	7.0	3.2	4.7	1.4	versicolor
51	6.4	3.2	4.5	1.5	versicolor
52	6.9	3.1	4.9	1.5	versicolor
54	6.5	2.8	4.6	1.5	versicolor
56	6.3	3.3	4.7	1.6	versicolor
58	6.6	2.9	4.6	1.3	versicolor
61	5.9	3.0	4.2	1.5	versicolor
62	6.0	2.2	4.0	1.0	versicolor
63	6.1	2.9	4.7	1.4	versicolor
65	6.7	3.1	4.4	1.4	versicolor
68	6.2	2.2	4.5	1.5	versicolor
70	5.9	3.2	4.8	1.8	versicolor
71	6.1	2.8	4.0	1.3	versicolor
72	6.3	2.5	4.9	1.5	versicolor
73	6.1	2.8	4.7	1.2	versicolor
74	6.4	2.9	4.3	1.3	versicolor
75	6.6	3.0	4.4	1.4	versicolor
76	6.8	2.8	4.8	1.4	versicolor
77	6.7	3.0	5.0	1.7	versicolor
78	6.0	2.9	4.5	1.5	versicolor
83	6.0	2.7	5.1	1.6	versicolor
85	6.0	3.4	4.5	1.6	versicolor
86	6.7	3.1	4.7	1.5	versicolor
87	6.3	2.3	4.4	1.3	versicolor
91	6.1	3.0	4.6	1.4	versicolor
97	6.2	2.9	4.3	1.3	versicolor

06. reset_ index로 인덱스를 초기화 시키기

In [46]:

```
### 조건식을 이용하여 데이터를 추출하고, index를 초기화 시켜 보자.
iris_versi = iris[ iris['species']=='versicolor' ].reset_index(drop=True)
iris_versi
```

Out[46]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	7.0	3.2	4.7	1.4	versicolor
1	6.4	3.2	4.5	1.5	versicolor
2	6.9	3.1	4.9	1.5	versicolor
3	5.5	2.3	4.0	1.3	versicolor
4	6.5	2.8	4.6	1.5	versicolor
5	5.7	2.8	4.5	1.3	versicolor
6	6.3	3.3	4.7	1.6	versicolor
7	4.9	2.4	3.3	1.0	versicolor
8	6.6	2.9	4.6	1.3	versicolor
9	5.2	2.7	3.9	1.4	versicolor
10	5.0	2.0	3.5	1.0	versicolor
11	5.9	3.0	4.2	1.5	versicolor
12	6.0	2.2	4.0	1.0	versicolor
13	6.1	2.9	4.7	1.4	versicolor
14	5.6	2.9	3.6	1.3	versicolor
15	6.7	3.1	4.4	1.4	versicolor
16	5.6	3.0	4.5	1.5	versicolor
17	5.8	2.7	4.1	1.0	versicolor
18	6.2	2.2	4.5	1.5	versicolor
19	5.6	2.5	3.9	1.1	versicolor
20	5.9	3.2	4.8	1.8	versicolor
21	6.1	2.8	4.0	1.3	versicolor
22	6.3	2.5	4.9	1.5	versicolor
23	6.1	2.8	4.7	1.2	versicolor
24	6.4	2.9	4.3	1.3	versicolor
25	6.6	3.0	4.4	1.4	versicolor
26	6.8	2.8	4.8	1.4	versicolor
27	6.7	3.0	5.0	1.7	versicolor
28	6.0	2.9	4.5	1.5	versicolor
29	5.7	2.6	3.5	1.0	versicolor
30	5.5	2.4	3.8	1.1	versicolor
31	5.5	2.4	3.7	1.0	versicolor
32	5.8	2.7	3.9	1.2	versicolor

	sepal_length	sepal_width	petal_length	petal_width	species
33	6.0	2.7	5.1	1.6	versicolor
34	5.4	3.0	4.5	1.5	versicolor
35	6.0	3.4	4.5	1.6	versicolor
36	6.7	3.1	4.7	1.5	versicolor
37	6.3	2.3	4.4	1.3	versicolor
38	5.6	3.0	4.1	1.3	versicolor
39	5.5	2.5	4.0	1.3	versicolor
40	5.5	2.6	4.4	1.2	versicolor
41	6.1	3.0	4.6	1.4	versicolor
42	5.8	2.6	4.0	1.2	versicolor
43	5.0	2.3	3.3	1.0	versicolor
44	5.6	2.7	4.2	1.3	versicolor
45	5.7	3.0	4.2	1.2	versicolor
46	5.7	2.9	4.2	1.3	versicolor
47	6.2	2.9	4.3	1.3	versicolor
48	5.1	2.5	3.0	1.1	versicolor
49	5.7	2.8	4.1	1.3	versicolor

(실습) iris_tmp의 행의 index를 초기화 시키고, 총 몇 행인지 확인해 보자.

In [47]:

```
iris_tmp = iris_tmp.reset_index(drop=True)
print( iris_tmp.shape )
iris_tmp.head()
```

(26, 5)

Out[47]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	7.0	3.2	4.7	1.4	versicolor
1	6.4	3.2	4.5	1.5	versicolor
2	6.9	3.1	4.9	1.5	versicolor
3	6.5	2.8	4.6	1.5	versicolor
4	6.3	3.3	4.7	1.6	versicolor

네개의 특성에 대한 상관계수 구해보기

In [48]:

```
iris.corr()
```

Out[48]:

	sepal_length	sepal_width	petal_length	petal_width
sepal_length	1.000000	-0.117570	0.871754	0.817941
sepal_width	-0.117570	1.000000	-0.428440	-0.366126
petal_length	0.871754	-0.428440	1.000000	0.962865
petal_width	0.817941	-0.366126	0.962865	1.000000

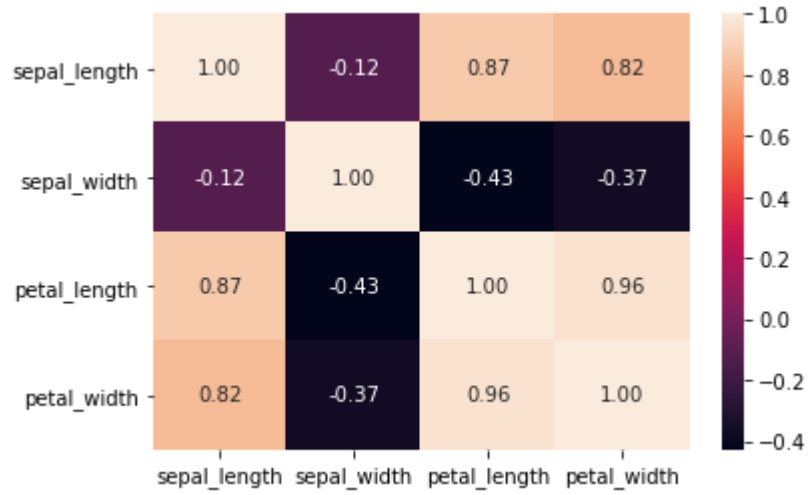
히트맵

In [49]:

```
sns.heatmap(iris.corr(), annot=True, fmt=".2f")
```

Out[49]:

<AxesSubplot:>

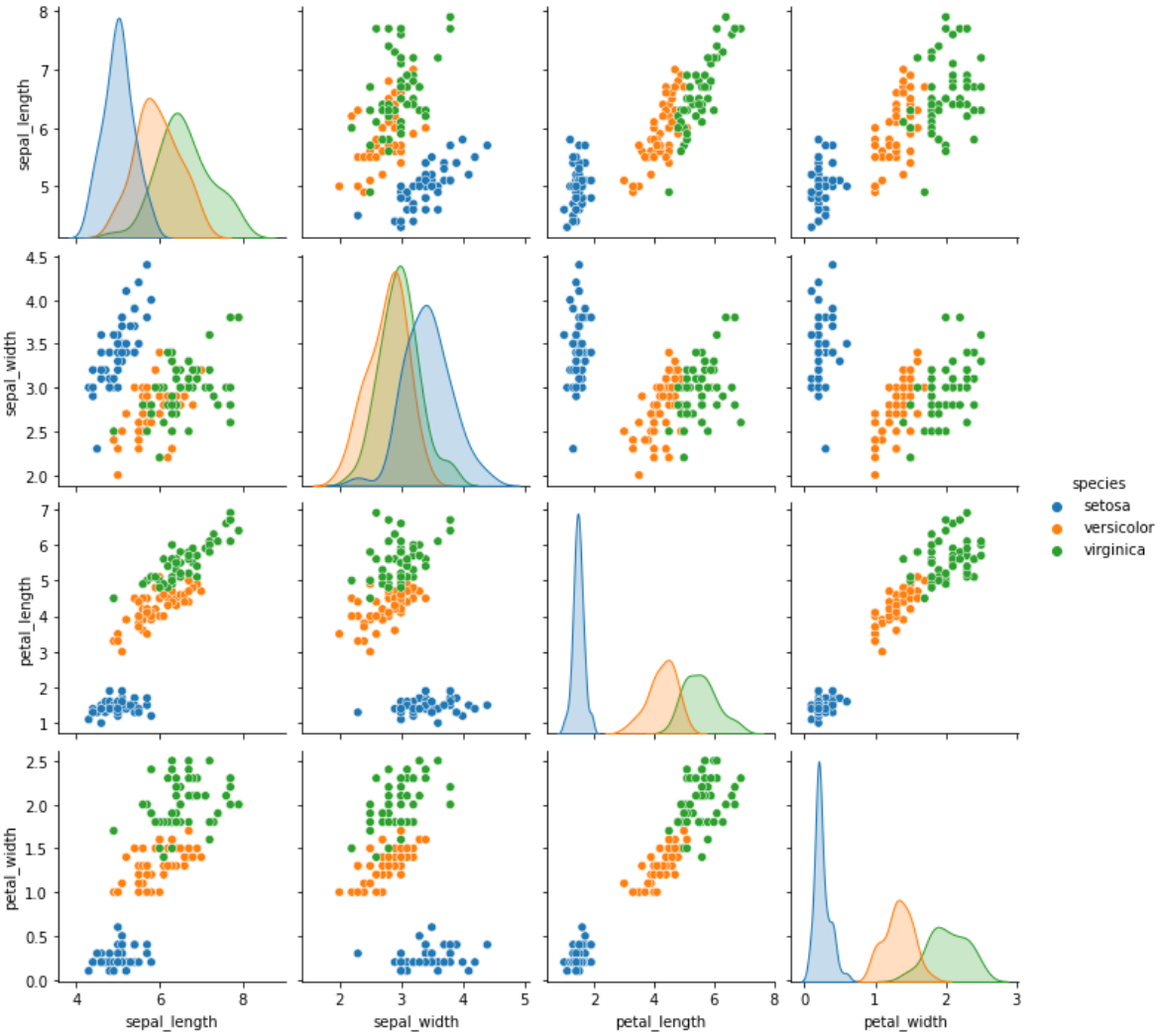


In [51]:

```
sns.pairplot(iris, hue='species')
```

Out[51]:

<seaborn.axisgrid.PairGrid at 0x1b5a02c88b0>



07. sort_values()를 이용한 정렬

In [53]:

```
iris.head()
```

Out[53]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

sepal_length로 정렬하기

In [56]:

```
iris.sort_values(by='sepal_length', ascending=False)
```

Out[56]:

	sepal_length	sepal_width	petal_length	petal_width	species
131	7.9	3.8	6.4	2.0	virginica
135	7.7	3.0	6.1	2.3	virginica
122	7.7	2.8	6.7	2.0	virginica
117	7.7	3.8	6.7	2.2	virginica
118	7.7	2.6	6.9	2.3	virginica
...	...	...	...	...	...
41	4.5	2.3	1.3	0.3	setosa
42	4.4	3.2	1.3	0.2	setosa
38	4.4	3.0	1.3	0.2	setosa
8	4.4	2.9	1.4	0.2	setosa
13	4.3	3.0	1.1	0.1	setosa

150 rows × 5 columns

In [57]:

```
iris.sort_values(by=['sepal_length', 'sepal_width'], ascending=False)
```

Out[57]:

	sepal_length	sepal_width	petal_length	petal_width	species
131	7.9	3.8	6.4	2.0	virginica
117	7.7	3.8	6.7	2.2	virginica
135	7.7	3.0	6.1	2.3	virginica
122	7.7	2.8	6.7	2.0	virginica
118	7.7	2.6	6.9	2.3	virginica
...	...	...	...	...	...
41	4.5	2.3	1.3	0.3	setosa
42	4.4	3.2	1.3	0.2	setosa
38	4.4	3.0	1.3	0.2	setosa
8	4.4	2.9	1.4	0.2	setosa
13	4.3	3.0	1.1	0.1	setosa

150 rows × 5 columns

08. astype를 이용한 데이터 자료형 변환

In [58]:

```
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
```

object를 category로 변경하기

In [59]:

```
iris['species'] = iris['species'].astype('category')
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    category
dtypes: category(1), float64(4)
memory usage: 5.1 KB
```