202001555 지은미

In [1]:

```
import pandas as pd
data=pd.Series([0.25,0.5,0.75,1.0])
data[0:3]
```

Out[1]:

```
0    0.25
1    0.50
2    0.75
dtype: float64
```

In [2]:

```
area=pd.Series({'정보기술대':7,
                '도서관':6,
                '자연과학대':5,
                '공과대':8})
depart=pd.Series({'정보기술대':'컴퓨터공학부',
                  '도서관':'열람실',
                  '자연과학대':'생명과학부',
                  '공과대':'기계공학과'})
inu=pd.DataFrame({'area':area,
                  'depart':depart})

inu
```

Out[2]:

	area	depart
정보기술대	7	컴퓨터공학부
도서관	6	열람실
자연과학대	5	생명과학부
공과대	8	기계공학과

In [3]:

```
inputdata={"name":["kim","Lim","Sung","Choi"],"year":[2012,2013,2014,2015],"month":[6,
inputFrame=pd.DataFrame(inputdata)
inputFrame
```

Out[3]:

	name	year	month	day
0	kim	2012	6	10
1	Lim	2013	7	11
2	Sung	2014	8	12
3	Choi	2015	9	13

In [4]:

```
A=pd.Series([2,4,6])
B=pd.Series([1,3,5])
A-B
```

Out[4]:

```
0    1
1    1
2    1
dtype: int64
```

In [5]:

```
A*B
```

Out[5]:

```
0     2
1    12
2    30
dtype: int64
```

In [6]:
```
C=pd.Series([1,2,3,4])
D=pd.Series([1,2,3])
C+D
```

Out[6]:
```
0    2.0
1    4.0
2    6.0
3    NaN
dtype: float64
```

In [7]:
```
C/D
```

Out[7]:
```
0    1.0
1    1.0
2    1.0
3    NaN
dtype: float64
```

In [8]:
```
A_raw_data={'col0':[1,2,3,4],'col1':[10,20,30,40],'col2':[100,200,300,400]}
B_raw_data={'col0':[1,2,3,4,5],'col1':[11,22,33,44,55],'col2':[111,222,333,444,555],'c

ADF=pd.DataFrame(A_raw_data)
BDF=pd.DataFrame(B_raw_data)
ADF-BDF
```

Out[8]:

	col0	col1	col2	col3
0	0.0	-1.0	-11.0	NaN
1	0.0	-2.0	-22.0	NaN
2	0.0	-3.0	-33.0	NaN
3	0.0	-4.0	-44.0	NaN
4	NaN	NaN	NaN	NaN

In [9]:
```
ADF.add(BDF,fill_value=0)
```

Out[9]:

	col0	col1	col2	col3
0	2.0	21.0	211.0	1111.0
1	4.0	42.0	422.0	2222.0
2	6.0	63.0	633.0	3333.0
3	8.0	84.0	844.0	4444.0
4	5.0	55.0	555.0	5555.0

In [10]:
```
C_raw_data={'col0':[1,2,3,4],'col1':[10,20,30,40],'col2':[100,200,300,400]}
D_raw_data={'col0':[1,2,3,4],'col1':[11,22,33,44],'col2':[111,222,333,444]}

CDF=pd.DataFrame(C_raw_data)
DDF=pd.DataFrame(D_raw_data)
CDF*DDF
```

Out[10]:

	col0	col1	col2
0	1	110	11100
1	4	440	44400
2	9	990	99900

	col0	col1	col2
3	16	1760	177600

In [11]:

```
# 데이터 불러오기

import pandas as pd
from sklearn.datasets import load_iris
iris=load_iris()
irisdf=pd.DataFrame(iris.data,columns=iris.feature_names)
irisdf['target']=iris.target
irisdf['target']=irisdf['target'].map({0:"setosa", 1:"versicolor", 2:"virginica"})
irisdf
```

Out[11]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
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

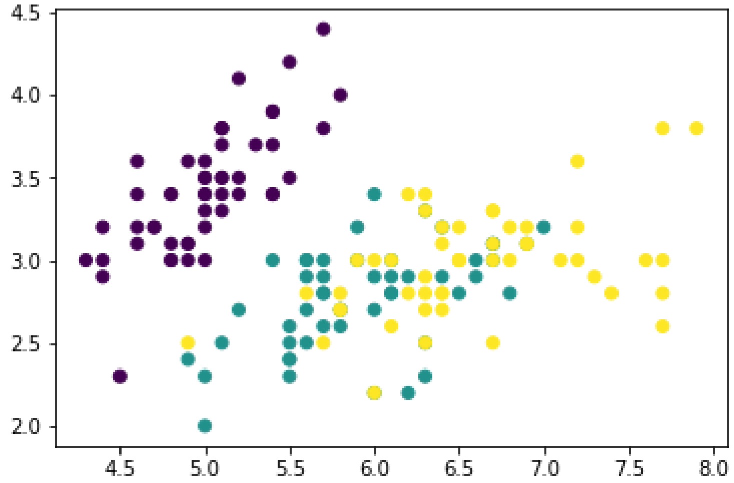
150 rows × 5 columns

In [12]:

```
import matplotlib.pyplot as plt
SL=irisdf.iloc[:,0] #꽃받침 길이
SW=irisdf.iloc[:,1] #꽃받침 너비
PL=irisdf.iloc[:,2] #꽃잎 길이
PW=irisdf.iloc[:,3] #꽃잎 너비
name=irisdf.iloc[:,4]
plt.scatter(SL,SW,c=iris.target)
```

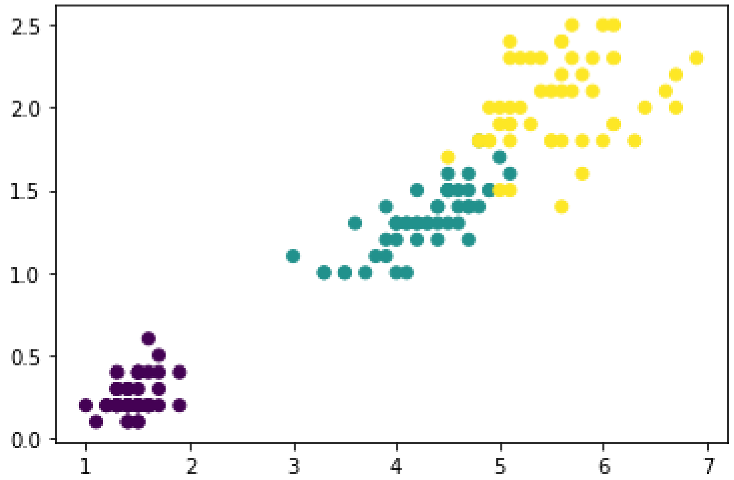
Out[12]:

```
<matplotlib.collections.PathCollection at 0x23d04a6b640>
```



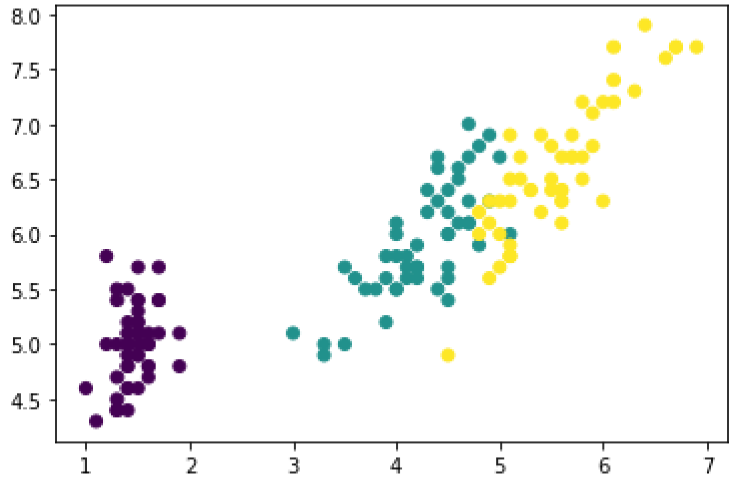
```
In [13]: plt.scatter(PL,PW,c=iris.target)

Out[13]: <matplotlib.collections.PathCollection at 0x23d04b54c70>
```



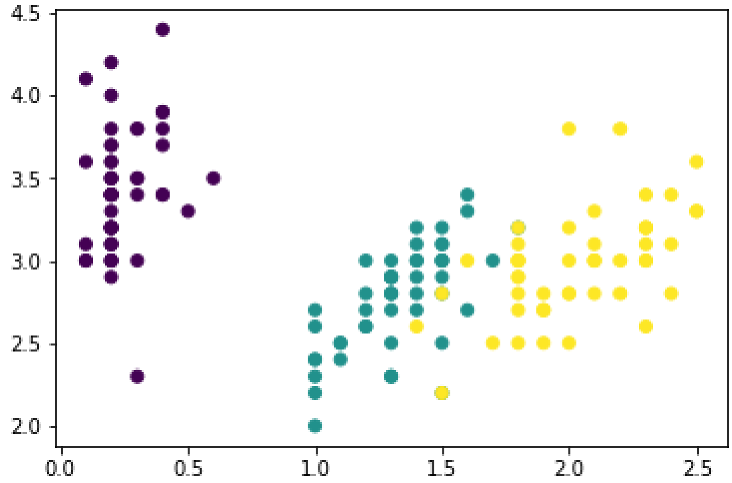
```
In [14]: plt.scatter(PL,SL,c=iris.target)

Out[14]: <matplotlib.collections.PathCollection at 0x23d04bbe100>
```



```
In [15]: plt.scatter(PW,SW,c=iris.target)

Out[15]: <matplotlib.collections.PathCollection at 0x23d04c13970>
```



In [16]:

```
#평균구하기
iris_group=irisdf.groupby('target')
iris_m=iris_group.mean()
iris_m
```

Out[16]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
target				
setosa	5.006	3.428	1.462	0.246
versicolor	5.936	2.770	4.260	1.326
virginica	6.588	2.974	5.552	2.026

In [17]:

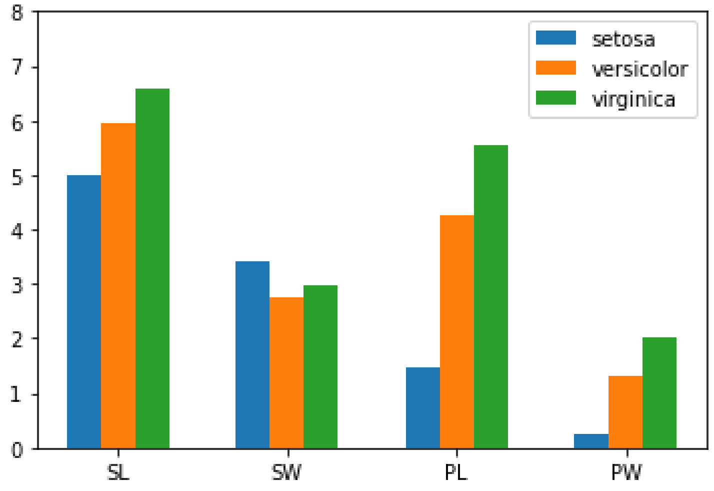
```
#표1 : SL,SW, PL, PW
setosa=iris_m.iloc[0]
versicolor=iris_m.iloc[1]
virginica=iris_m.iloc[2]

name=["SL","SW","PL","PW"]

xs1=[i + 0.1 for i, _ in enumerate(name)]
xs2=[i + 0.3 for i, _ in enumerate(name)]
xs3=[i + 0.5 for i, _ in enumerate(name)]

plt.bar(xs1,setosa,width=0.2,label="setosa")
plt.bar(xs2,versicolor,width=0.2,label="versicolor")
plt.bar(xs3,virginica,width=0.2,label="virginica")

plt.xticks([i + 0.3 for i, _ in enumerate(name)], name)
plt.ylim([0,8])
plt.legend(loc="upper right")
plt.show()
```



In [18]:

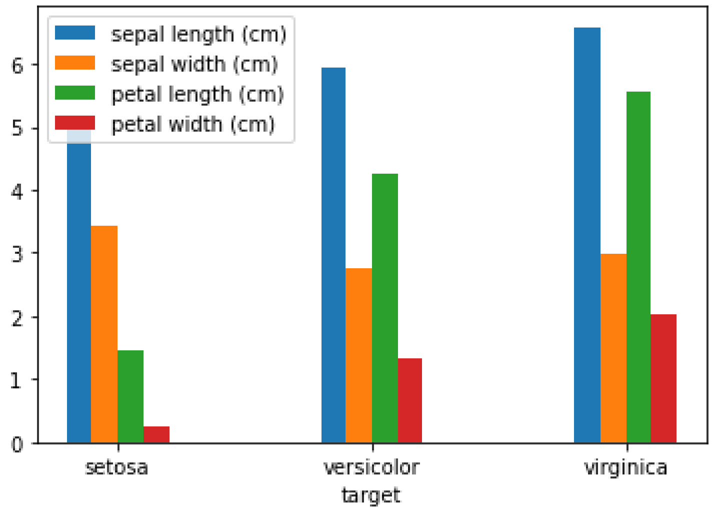
```
#표2 : setosa,versicolor, virginica
SL=iris_m.iloc[:,0]
SW=iris_m.iloc[:,1]
PL=iris_m.iloc[:,2]
PW=iris_m.iloc[:,3]

flower=["setosa","versicolor","virginica"]

xs1=[i + 0.1 for i, _ in enumerate(flower)]
xs2=[i + 0.2 for i, _ in enumerate(flower)]
xs3=[i + 0.3 for i, _ in enumerate(flower)]
xs4=[i + 0.4 for i, _ in enumerate(flower)]

plt.bar(xs1,SL,width=0.1,label="sepal length (cm)")
plt.bar(xs2,SW,width=0.1,label="sepal width (cm)")
plt.bar(xs3,PL,width=0.1,label="petal length (cm)")
plt.bar(xs4,PW,width=0.1,label="petal width (cm)")

plt.xticks([i + 0.25 for i, _ in enumerate(flower)], flower)
plt.xlabel("target")
plt.legend(loc="upper left")
plt.show()
```



이러한 데이터를 분석하는 이유는 꽃잎와 꽃받침의 너비와 길이를 판단하여 꽃의 종류를 맞추는 것이 중요합니다. 표 1을 참고하면 꽃의 종류마다 꽃잎의 너비, 길이 그리고 꽃받침의 너비, 길이를 한눈에 보기 쉽도록 표현이 되어 꽃의 종류를 판단하는데 잘 나타낸 느낌이 듭니다. 하지만 표 2를 보게 되면 표 1에 비해 상대적으로 꽃의 종류에 따라 구분되어 나타나있어 데이터를 분석하는데에 표1이 더 데이터의 특징을 잘 나타내는 것 같다고 생각합니다.

In [19]:

```
# 데이터 슬라이싱
import numpy as np

iris_setosa=iris_group.get_group('setosa')
iris_versicolor=iris_group.get_group('versicolor')
iris_virginica=iris_group.get_group('virginica')

list_setosa=list(range(50))
list_versicolor=list(range(50, 100))
list_virginica=list(range(100, 150))

np.random.shuffle(list_setosa)
np.random.shuffle(list_versicolor)
np.random.shuffle(list_virginica)


test_dataset=irisdf
train_dataset=irisdf

for i in range(50//4):
    s_setosa=list_setosa.pop()
    s_versicolor=list_versicolor.pop()
    s_virginica=list_virginica.pop()
    train_dataset=train_dataset.drop(s_setosa)
    train_dataset=train_dataset.drop(s_versicolor)
    train_dataset=train_dataset.drop(s_virginica)
test_dataset=test_dataset.drop(test_dataset.index[train_dataset.index])
train_dataset
```

Out[19]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
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
6	4.6	3.4	1.4	0.3	setosa
...	...	...	...	...	...
144	6.7	3.3	5.7	2.5	virginica
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

114 rows × 5 columns

In [20]:

```
test_dataset
```

Out[20]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
4	5.0	3.6	1.4	0.2	setosa
5	5.4	3.9	1.7	0.4	setosa
11	4.8	3.4	1.6	0.2	setosa
12	4.8	3.0	1.4	0.1	setosa

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
15	5.7	4.4	1.5	0.4	setosa
16	5.4	3.9	1.3	0.4	setosa
19	5.1	3.8	1.5	0.3	setosa
23	5.1	3.3	1.7	0.5	setosa
25	5.0	3.0	1.6	0.2	setosa
27	5.2	3.5	1.5	0.2	setosa
44	5.1	3.8	1.9	0.4	setosa
47	4.6	3.2	1.4	0.2	setosa
52	6.9	3.1	4.9	1.5	versicolor
55	5.7	2.8	4.5	1.3	versicolor
57	4.9	2.4	3.3	1.0	versicolor
59	5.2	2.7	3.9	1.4	versicolor
61	5.9	3.0	4.2	1.5	versicolor
68	6.2	2.2	4.5	1.5	versicolor
75	6.6	3.0	4.4	1.4	versicolor
89	5.5	2.5	4.0	1.3	versicolor
91	6.1	3.0	4.6	1.4	versicolor
94	5.6	2.7	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
104	6.5	3.0	5.8	2.2	virginica
107	7.3	2.9	6.3	1.8	virginica
115	6.4	3.2	5.3	2.3	virginica
118	7.7	2.6	6.9	2.3	virginica
121	5.6	2.8	4.9	2.0	virginica
125	7.2	3.2	6.0	1.8	virginica
127	6.1	3.0	4.9	1.8	virginica
129	7.2	3.0	5.8	1.6	virginica
137	6.4	3.1	5.5	1.8	virginica
139	6.9	3.1	5.4	2.1	virginica
140	6.7	3.1	5.6	2.4	virginica
149	5.9	3.0	5.1	1.8	virginica

In [21]:

```
#파일입력
train_dataset.to_csv("./train_dataset.csv")
test_dataset.to_csv("./test_dataset.csv")
```

In [22]:

```
test_dataset=pd.read_csv("./test_dataset.csv")
```

```
test_dataset=test_dataset.drop(["Unnamed: 0"],axis=1)
test_dataset
```

Out[22]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
0	5.0	3.6	1.4	0.2	setosa
1	5.4	3.9	1.7	0.4	setosa
2	4.8	3.4	1.6	0.2	setosa
3	4.8	3.0	1.4	0.1	setosa
4	5.7	4.4	1.5	0.4	setosa
5	5.4	3.9	1.3	0.4	setosa
6	5.1	3.8	1.5	0.3	setosa
7	5.1	3.3	1.7	0.5	setosa
8	5.0	3.0	1.6	0.2	setosa
9	5.2	3.5	1.5	0.2	setosa
10	5.1	3.8	1.9	0.4	setosa
11	4.6	3.2	1.4	0.2	setosa
12	6.9	3.1	4.9	1.5	versicolor
13	5.7	2.8	4.5	1.3	versicolor
14	4.9	2.4	3.3	1.0	versicolor
15	5.2	2.7	3.9	1.4	versicolor
16	5.9	3.0	4.2	1.5	versicolor
17	6.2	2.2	4.5	1.5	versicolor
18	6.6	3.0	4.4	1.4	versicolor
19	5.5	2.5	4.0	1.3	versicolor
20	6.1	3.0	4.6	1.4	versicolor
21	5.6	2.7	4.2	1.3	versicolor
22	6.2	2.9	4.3	1.3	versicolor
23	5.1	2.5	3.0	1.1	versicolor
24	6.5	3.0	5.8	2.2	virginica
25	7.3	2.9	6.3	1.8	virginica
26	6.4	3.2	5.3	2.3	virginica
27	7.7	2.6	6.9	2.3	virginica
28	5.6	2.8	4.9	2.0	virginica
29	7.2	3.2	6.0	1.8	virginica
30	6.1	3.0	4.9	1.8	virginica
31	7.2	3.0	5.8	1.6	virginica
32	6.4	3.1	5.5	1.8	virginica
33	6.9	3.1	5.4	2.1	virginica

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
34	6.7	3.1	5.6	2.4	virginica
35	5.9	3.0	5.1	1.8	virginica

In [23]:

```
train_dataset=pd.read_csv("./train_dataset.csv")
train_dataset=train_dataset.drop(["Unnamed: 0"],axis=1)
train_dataset
```

Out[23]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
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	4.6	3.4	1.4	0.3	setosa
...	...	...	...	...	...
109	6.7	3.3	5.7	2.5	virginica
110	6.7	3.0	5.2	2.3	virginica
111	6.3	2.5	5.0	1.9	virginica
112	6.5	3.0	5.2	2.0	virginica
113	6.2	3.4	5.4	2.3	virginica

114 rows × 5 columns

In [24]:

```
testdata_group=test_dataset.groupby("target")
testdata_m=testdata_group.mean()
testdata_m
```

Out[24]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
target				
setosa	5.100000	3.566667	1.541667	0.291667
versicolor	5.825000	2.733333	4.150000	1.333333
virginica	6.658333	3.000000	5.625000	1.991667

In [25]:

```
traindata_group=train_dataset.groupby("target")
traindata_m=traindata_group.mean()
traindata_m
```

Out[25]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
target				
setosa	4.976316	3.384211	1.436842	0.231579
versicolor	5.971053	2.781579	4.294737	1.323684
virginica	6.565789	2.965789	5.528947	2.036842

```
setosa1=traindata_m.iloc[0]
```

In [27]:

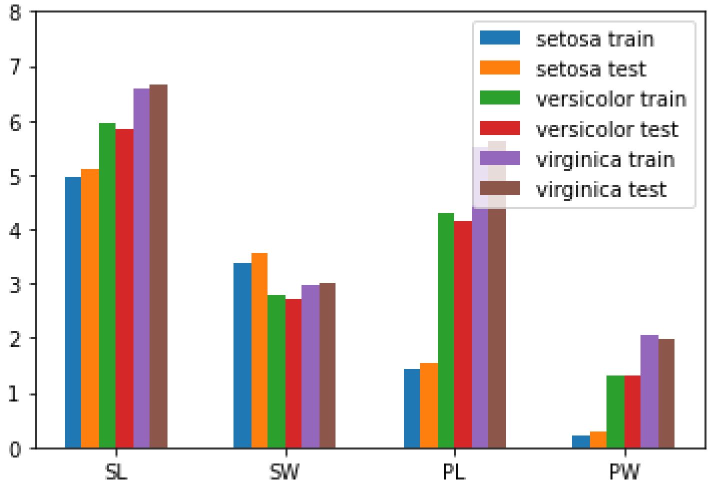
```
versicolor1=traindata_m.iloc[1]
virginica1=traindata_m.iloc[2]
setosa2=testdata_m.iloc[0]
versicolor2=testdata_m.iloc[1]
virginica2=testdata_m.iloc[2]

name=["SL","SW","PL","PW"]
flower=["setosa","versicolor","virginica"]

xs1=[i + 0.1 for i, _ in enumerate(name)]
xs2=[i + 0.2 for i, _ in enumerate(name)]
xs3=[i + 0.3 for i, _ in enumerate(name)]
xs4=[i + 0.4 for i, _ in enumerate(name)]
xs5=[i + 0.5 for i, _ in enumerate(name)]
xs6=[i + 0.6 for i, _ in enumerate(name)]

plt.bar(xs1,setosa1,width=0.1,label="setosa train")
plt.bar(xs2,setosa2,width=0.1,label="setosa test")
plt.bar(xs3,versicolor1,width=0.1,label="versicolor train")
plt.bar(xs4,versicolor2,width=0.1,label="versicolor test")
plt.bar(xs5,virginica1,width=0.1,label="virginica train")
plt.bar(xs6,virginica2,width=0.1,label="virginica test")

plt.xticks([i + 0.35 for i, _ in enumerate(name)], name)
plt.ylim([0,8])
plt.legend(loc="upper right")
plt.show()
```



202001555 지은미