

데이터 시각화 (2024)

데이터과학부 정진명

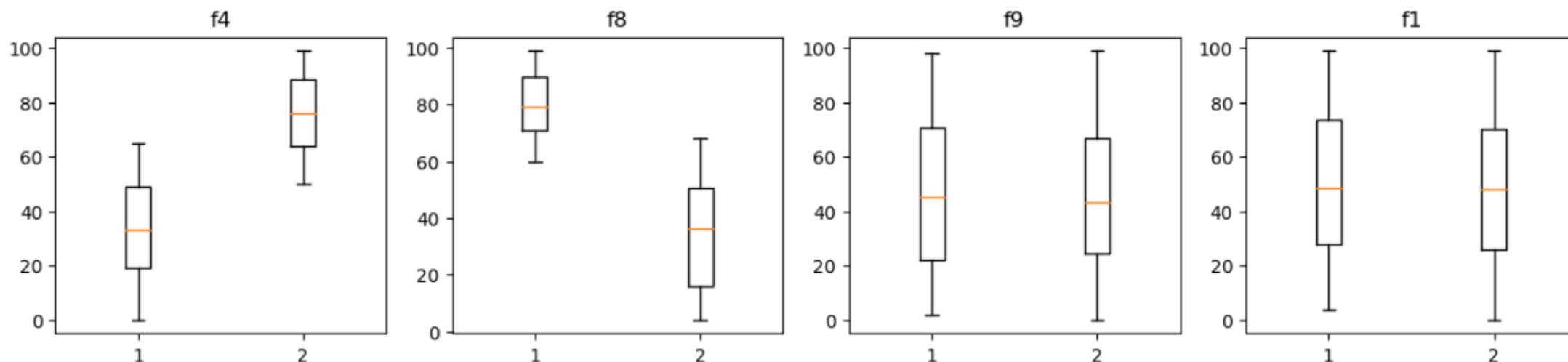
(jmjung@suwon.ac.kr, 글로벌경상관 918호)

6 주차

실습1

- 10개의 feature와 class column을 갖는 (총 11개의 column) dat_class.txt 파일을 있다.
- Class column은 0과 1의 값을 갖는다.
- 10개의 feature 중에, 0인 class 와 1인 class를
 - 구분을 제일 잘하는 feature 2개와,
 - 구분을 제일 못하는 feature 2개에 대해서class 별로 box plot을 그리시오.
- 잘 구분하는 정도는 class 0 인 feature의 평균과 class 1의 feature의 평균 차이로 정한다.

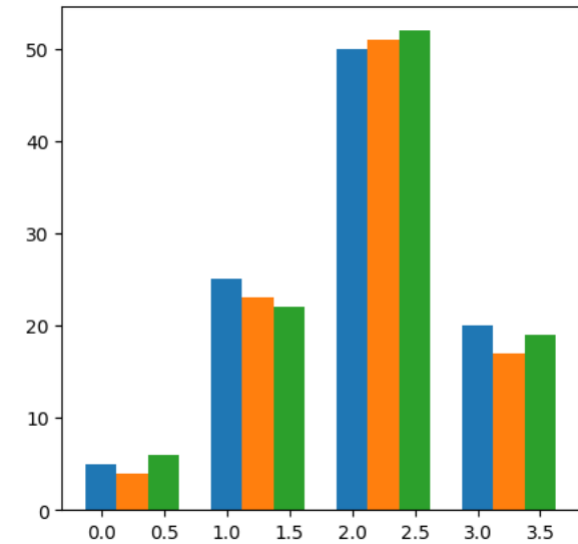
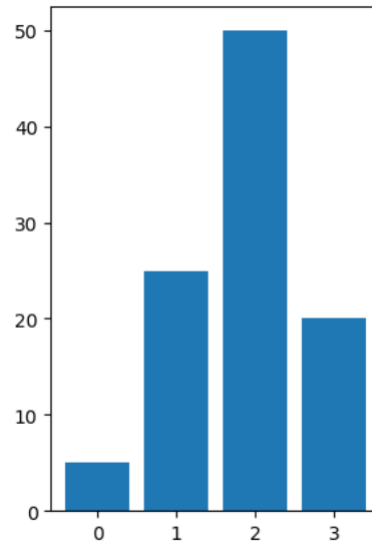
f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	class
91	43	76	57	92	18	43	65	80	81	1
23	57	73	79	79	75	41	50	64	60	1
19	7	76	71	7	97	37	61	81	50	1
73	83	70	57	28	60	82	77	4	36	0
62	54	50	71	19	90	92	65	20	92	1
98	86	1	68	20	1	7	43	12	64	1
35	61	94	62	19	79	62	39	96	70	1
52	43	17	87	33	23	14	13	47	24	1
52	14	93	97	22	42	49	6	58	72	1
44	29	88	16	87	61	36	80	88	94	0
93	35	71	13	35	5	29	67	15	24	0



color, alpha 파라미터 사용

자신만의 컬러 정의

- matplotlib에서 제공되는 기본 컬러는 단조로움



- 자신이 원하는 컬러 선택 가능

자신만의 컬러 정의

■ 컬러 사용 방법

1) 삼중 (triplet):

- 빨강(r), 초록(g), 파랑(b) 성분을 (r,g,b)로 표현. r,g,b는 0,1 사이의 float 값

2) 컬러 이름 사용

3) 컬러 이름의 축약어 사용:

- 많이 사용하는 몇 개만 존재

character	color
'b'	blue
'g'	green
'r'	red
'c'	cyan
'm'	magenta
'y'	yellow
'k'	black
'w'	white

4) 투명도 (transparency):

- alpha parameter: 0 (흐림) → 1 (진함)
- default: 1

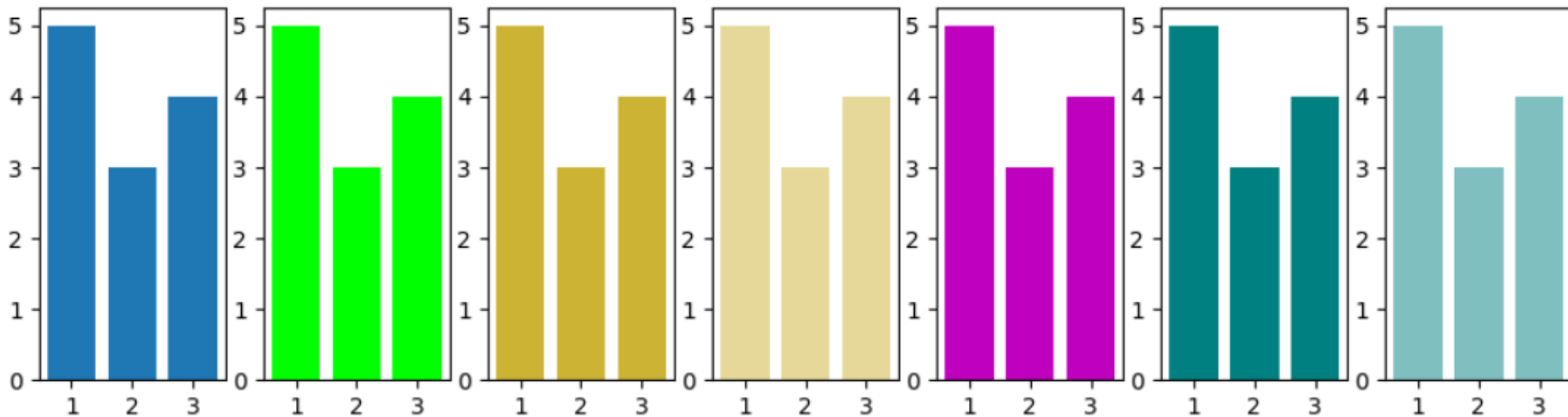


자신만의 컬러 정의 예제1

```
fig=plt.figure(figsize=(12,3), dpi=100)
axs=fig.subplots(1,7)

X=[1,2,3]
Y=[5,3,4]

_ =axs[0].bar(X,Y)
_ =axs[1].bar(X,Y,color=(0,1,0))
_ =axs[2].bar(X,Y,color=(0.8,0.7,0.2))
_ =axs[3].bar(X,Y,color=(0.8,0.7,0.2), alpha=0.5)
_ =axs[4].bar(X,Y,color='m')
_ =axs[5].bar(X,Y,color='teal')
_ =axs[6].bar(X,Y,color='teal',alpha=0.5)
```



자신만의 컬러 정의 예제2

```
array, scalar, scalar
def pdf(X, mu, sigma):
    a = 1/(sigma * np.sqrt(2*np.pi))
    b = -1/(2*(sigma**2))
    return a * np.exp(b * ((X - mu)** 2))

fig=plt.figure(figsize=(7,7), dpi=100)

ax=fig.subplots()

X = np.linspace(-6, 6, 1000)

for i in range(5):
    samples = np.random.normal(size=50) 표준정규분포에서 50개 값 sampling
    mu, sigma = np.mean(samples), np.std(samples)
    mu, sigma
    _=ax.plot(X, pdf(X, mu, sigma), color = 'k', alpha=0.4)

_=ax.plot(X, pdf(X, 0, 1), color = 'k')
```

정규분포 확률밀도함수 식

$$\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

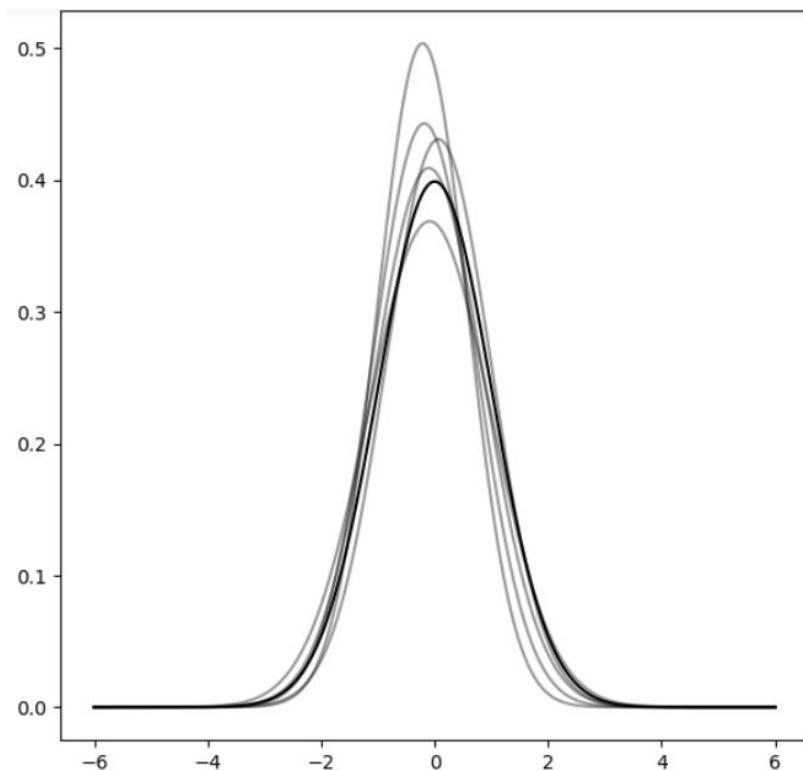
(-0.10235245790114064, 0.9749570620041083)

(-0.08980464598908616, 1.0819454737473644)

(0.06743394927728302, 0.9255415640715187)

(-0.21290740681421882, 0.7919418535429479)

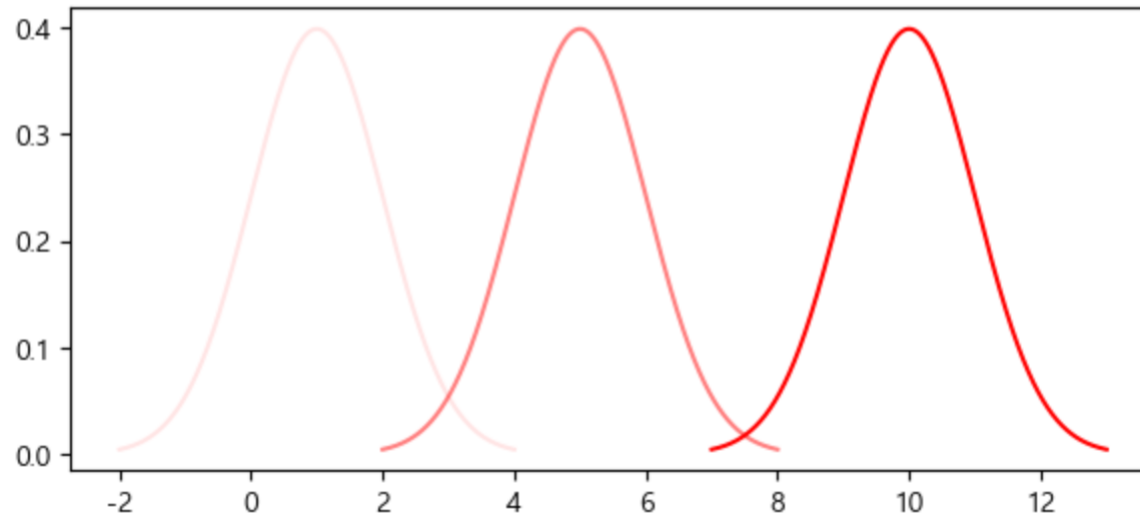
(-0.18299530734917305, 0.9005326316538114)



실습 2

- k 가 1 ~ 10 의 정수 중 하나라고 할 때,
평균 k , 표준편차 1 의 정규분포 확률밀도 함수를 그리는 함수 `draw_pdf_with_k`를 구현하시오
- 1) k 가 클수록 진하게 그리시오 ($\alpha = k * 0.1$)
- 2) 각 확률밀도 함수의 x 의 범위는 k 의 $-3 \sim +3$ 까지로 정하시오

```
draw_pdf_with_k(ax, 1)
draw_pdf_with_k(ax, 5)
draw_pdf_with_k(ax, 10)
```



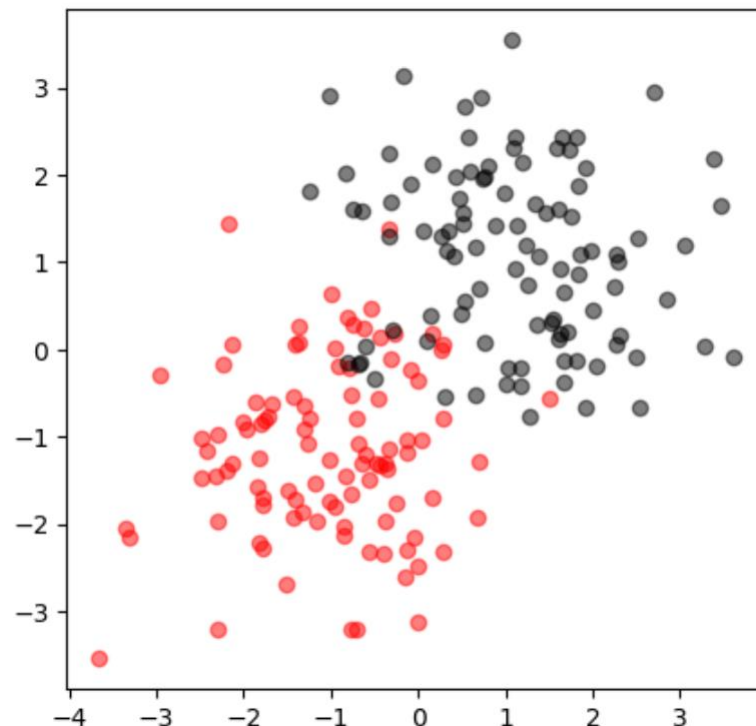
scatter 사용자 지정 컬러

```
fig=plt.figure(figsize=(5,5), dpi=100)
ax=fig.subplots()

A = np.random.normal(size=(100, 2))
A += np.array([-1, -1]) numpy broadcasting

B = np.random.normal(size=(100, 2))
B += np.array([1, 1]) numpy broadcasting

ax.scatter(A[:,0], A[:,1], color = 'r', alpha=0.5)
ax.scatter(B[:,0], B[:,1], color = 'k', alpha=0.5)
```

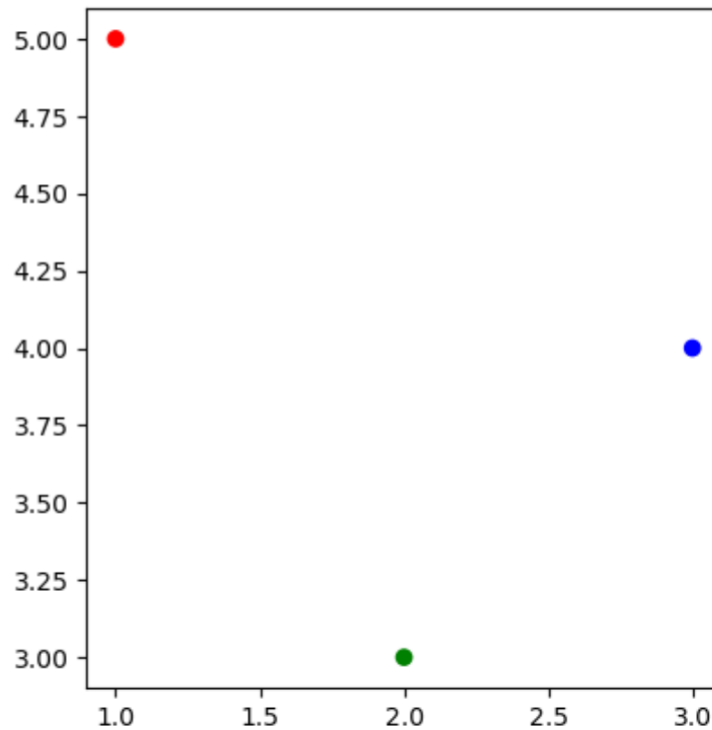
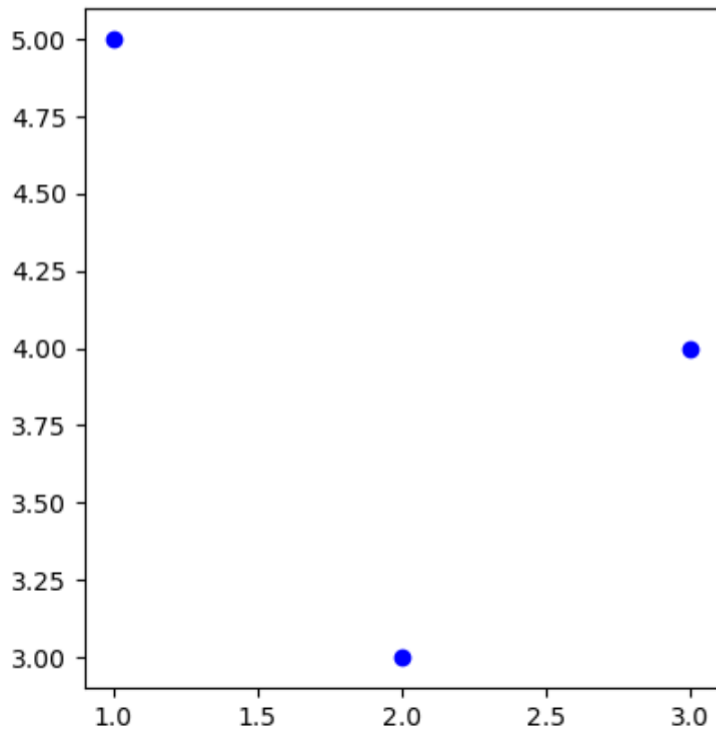


시퀀스 형태의 Color 파라미터 사용

시퀀스 형태의 color parameter

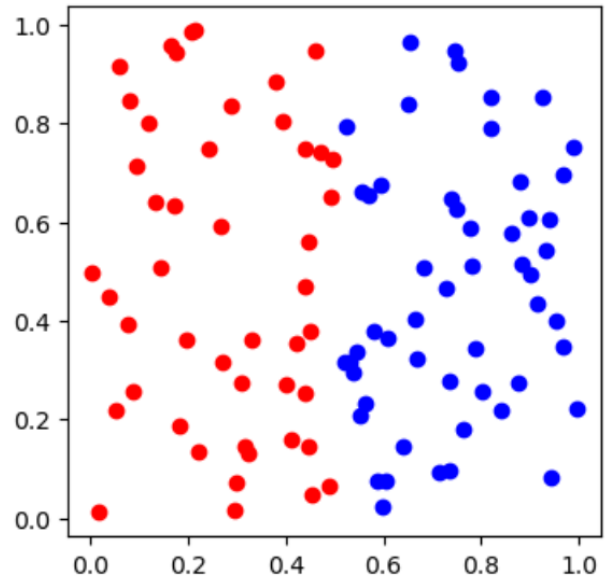
```
fig=plt.figure(figsize=(10,5), dpi=100)
ax1,ax2=fig.subplots(1,2)

_ax1.scatter([1,2,3],[5,3,4], color='b')
_ax2.scatter([1,2,3],[5,3,4], color=['r','g','b'])
```



실습 3 (3w 실습 revisit)

- 아래 그림과 같이 100개의 점을 두 가지 색깔로 scatter plot 하시오
 - 각 점의 x, y 좌표는 0~1에서 uniform하게 sampling한 값
 - 빨강 그룹의 x 좌표는 0~0.5, 파랑 그룹의 x 좌표는 0.5~1
 - color parameter를 사용하여 수행하시오



Iris scatter plot

- 오른쪽의 iris data의 'sep_len', 'sep_wid' column을 x,y로 scatter 하는데, 'name' column의 꽃 이름 별로 색깔을 다르게 하시오
- 꽃 이름과 그 색깔:
 - 1) Iris-setosa -> red
 - 2) Iris-versicolor -> green
 - 3) Iris-virginica -> blue

Iris data

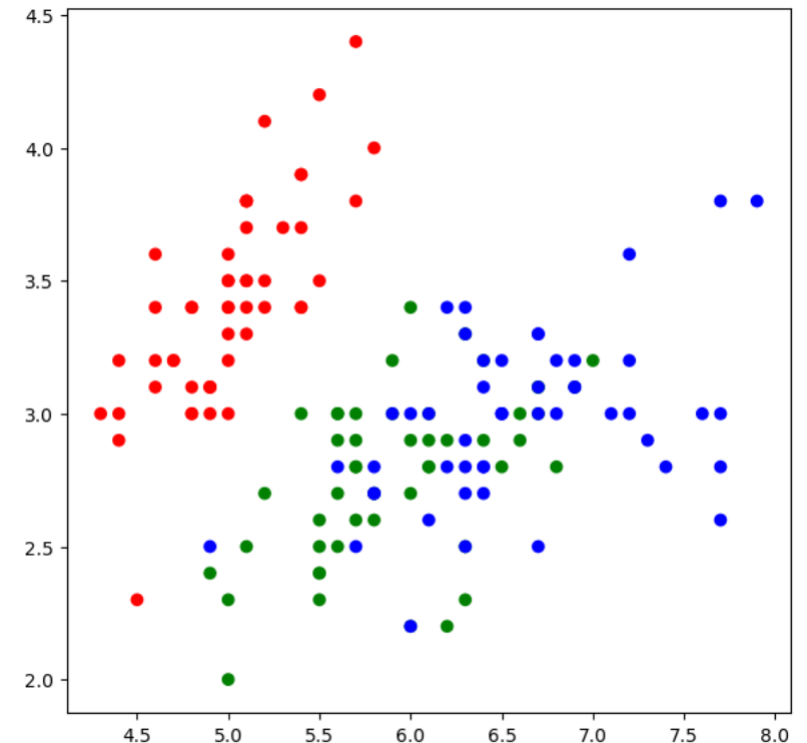
```
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa
5.4,3.9,1.7,0.4,Iris-setosa
4.6,3.4,1.4,0.3,Iris-setosa
```

```
fig=plt.figure(figsize=(7,3), dpi=100)
ax1,ax2=fig.subplots(1,2)

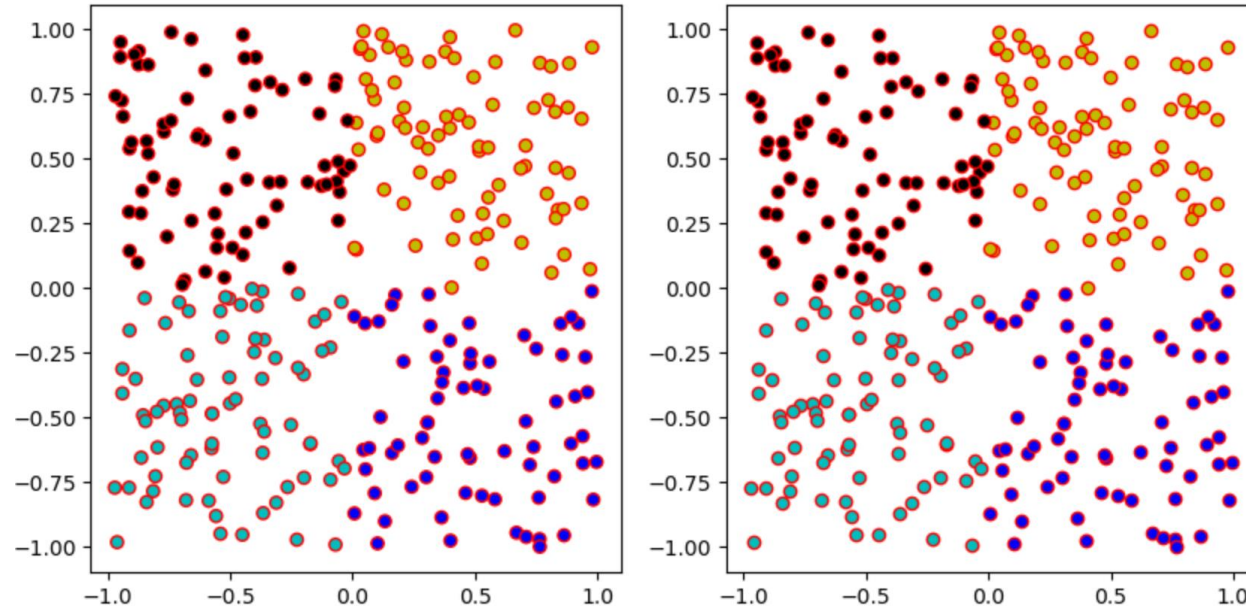
## data load
data = pd.read_table('data/dat_iris.txt', sep=',')
data.head()
color_dic={'Iris-setosa':'r','Iris-versicolor':'g','Iris-virginica':'b'}

## ax1 (시퀀스 형태의 color parameter)
color=data['name'].replace(color_dic)
color
ax1.scatter(data['sep_len'], data['sep_wid'], c = color, s=10)

## ax2 (값 하나의 color parameter)
for n1 in color_dic.keys():
    d1=data.loc[data['name']==n1]
    ax2.scatter(d1['sep_len'], d1['sep_wid'], c = color_dic[n1], s=10)
```



실습 4



- 위의 실습 5와 동일하게,
-1 < x < 1, -1 < y < 1 영역에 300개의 점을 랜덤하게 scatter 하는데,
1~4분면에 각각 다른 색깔로 scatter 하시오.
- ax1은 scatter 함수 한번 호출 (color_list 사용)
 - ax2는 scatter 함수 네 번 호출 (color_list 사용 x)

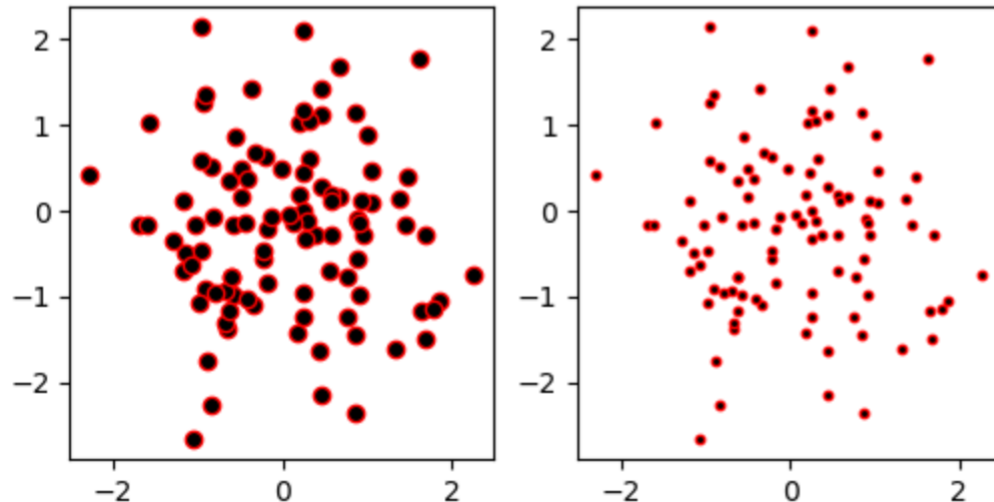
edgecolor and size

```
fig=plt.figure(figsize=(6,3), dpi=100)
ax1,ax2=fig.subplots(1,2)

data = np.random.normal(size=(100, 2))
ax1.scatter(data[:,0], data[:,1], color = 'k', edgecolor='r')
ax2.scatter(data[:,0], data[:,1], color = 'k', edgecolor='r', s=10)
```

<matplotlib.collections.PathCollection at 0x202c166c880>

<matplotlib.collections.PathCollection at 0x202c16b9100>



bar 다양한 컬러 사용

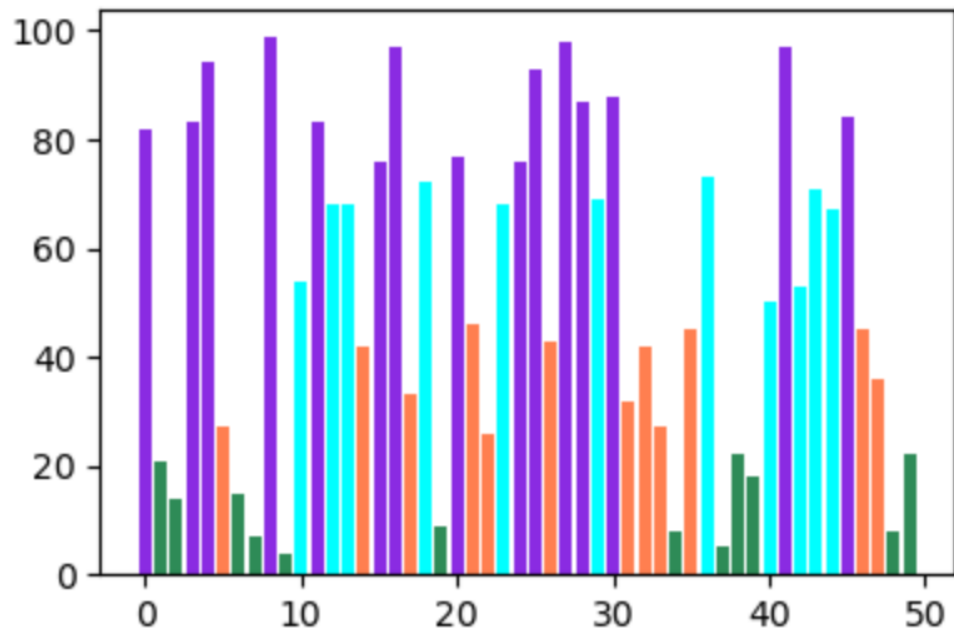
```
fig=plt.figure(figsize=(15,3), dpi=100)
ax1, ax2, ax3 =fig.subplots(1,3)

values=np.random.randint(100, size=50)
X=np.arange(len(values))
color_set = ['seagreen','coral','aqua','blueviolet']

## ax1
color_list=[]
for val in values:
    if val//25==0:
        color_list.append('seagreen')
        #color_list.append(color_set[0])
    elif val//25==1:
        color_list.append('coral')
        #color_list.append(color_set[1])
    elif val//25==2:
        color_list.append('aqua')
        #color_list.append(color_set[2])
    else:
        color_list.append('blueviolet')
        #color_list.append(color_set[3])

ax1.bar(X, values, color = color_list)

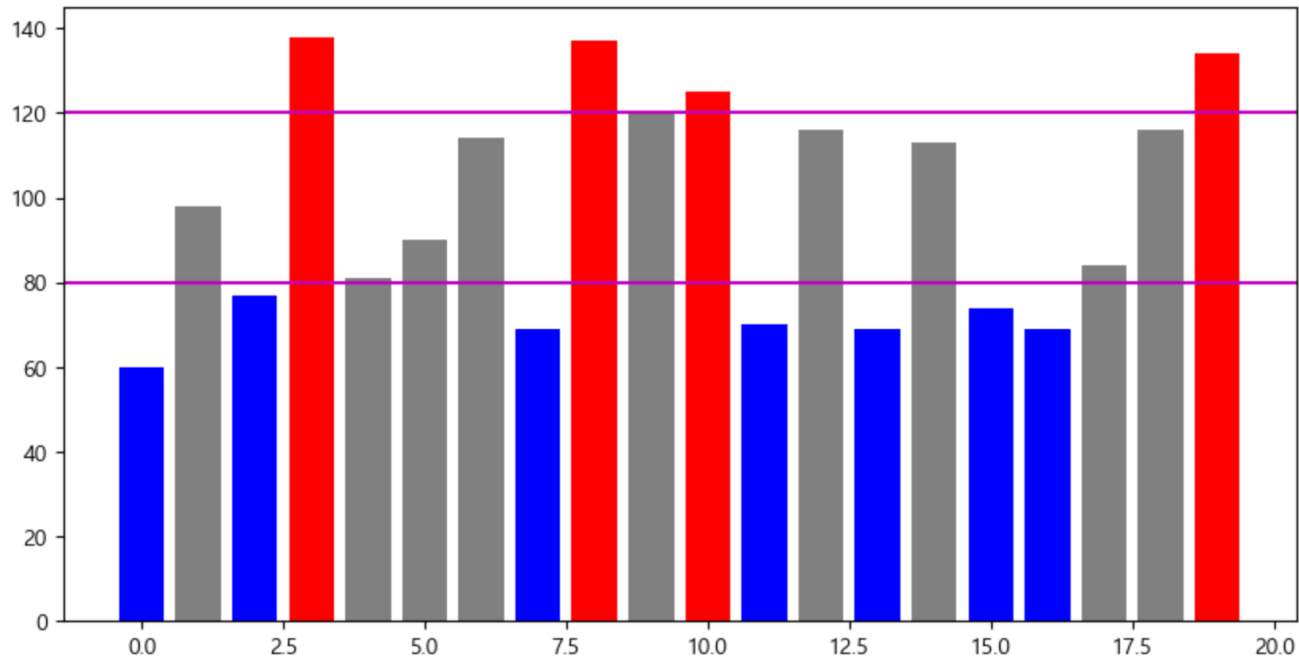
## ax2
color_list = [color_set[val//25] for val in values]
ax2.bar(X, values, color = color_list)
```



실습5

- 주어진 데이터 bp는 20명의 혈압 수치이다. 혈압을 차례로 bar 그래프로 나타내는데, 정상 혈압은 회색 (gray), 저혈압은 파란색 (b), 고혈압 빨간색(r)으로 나타내시오
(정상 혈압: 80~120, 80과 120 포함)

```
bp=np.random.randint(60,140, size=20)
```



Q & A

Thank you