

데이터 시각화 (2024)

데이터과학부 정진명

(jmjung@suwon.ac.kr, 글로벌경상관 918호)

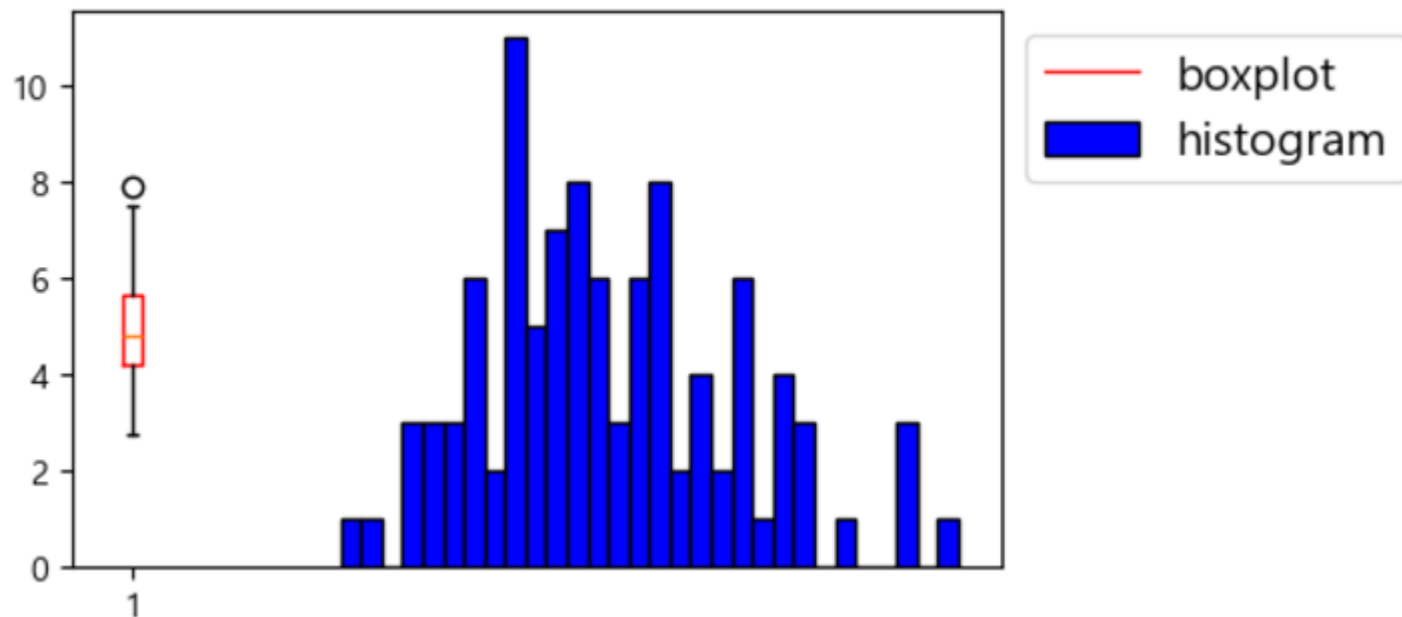
13 주차

Contents

- `annotate`
- `heatmap`
- `add_subplot`

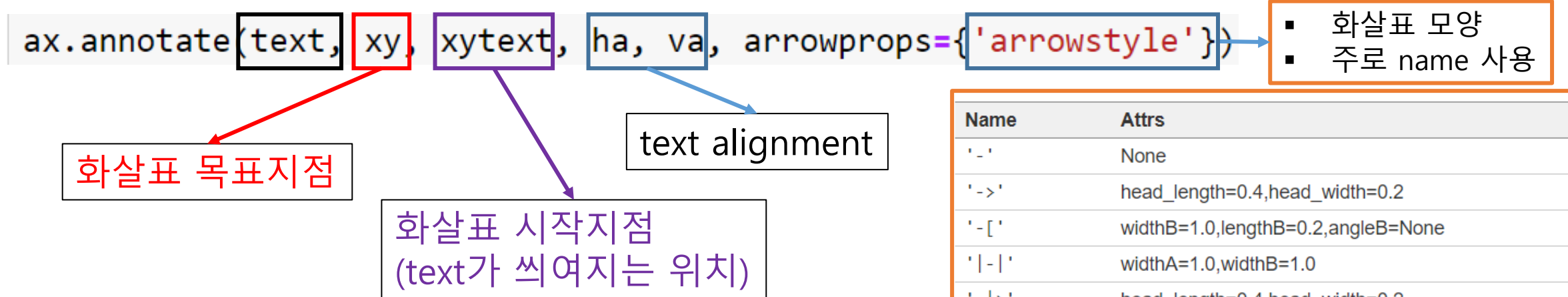
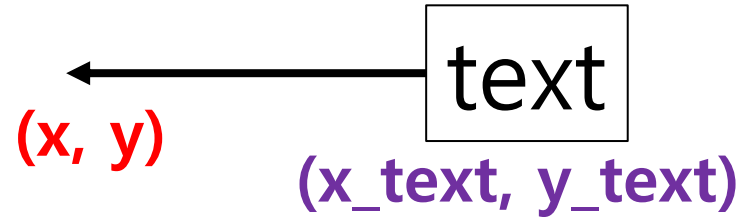
실습1 (12w legend 실습)

```
# 주어진 데이터 d1을 하나의 ax에 1) boxplot과 2) histogram으로 그리고  
# 결과 그림과 같이 legend를 넣으시오  
  
# hist parameter => bins=30, edgecolor='k', color='b'  
# boxplot parameter => boxprops = {'color': 'r'}
```



annotate

annotate (text + arrow)



- 화살표는 text box의 한 가운데를 가르킴 (text box 안에는 화살표 선이 표시되지 않음)

Name	Attrs
'_ '	None
'->'	head_length=0.4,head_width=0.2
'-['	widthB=1.0,lengthB=0.2,angleB=None
' - '	widthA=1.0,widthB=1.0
'- >'	head_length=0.4,head_width=0.2
'<-'	head_length=0.4,head_width=0.2
'<->'	head_length=0.4,head_width=0.2
'< -'	head_length=0.4,head_width=0.2
'< - >'	head_length=0.4,head_width=0.2
'fancy'	head_length=0.4,head_width=0.4,tail_width=0.4
'simple'	head_length=0.5,head_width=0.5,tail_width=0.2
'wedge'	tail_width=0.3,shrink_factor=0.5

annotate (text + arrow)

```
fig=plt.figure(figsize=(8,8), dpi=100)
ax1=fig.subplots()

_ax1.set_xticks(np.arange(-2,2.1,0.5))
_ax1.set_xticks(np.arange(-2,2,0.1), minor=True)
_ax1.set_yticks(np.arange(-2,2.1,0.5))
_ax1.set_yticks(np.arange(-2,2.1,0.1), minor=True)
```

minor tick 사용 방법
(minor tick은 tick label이 없음)

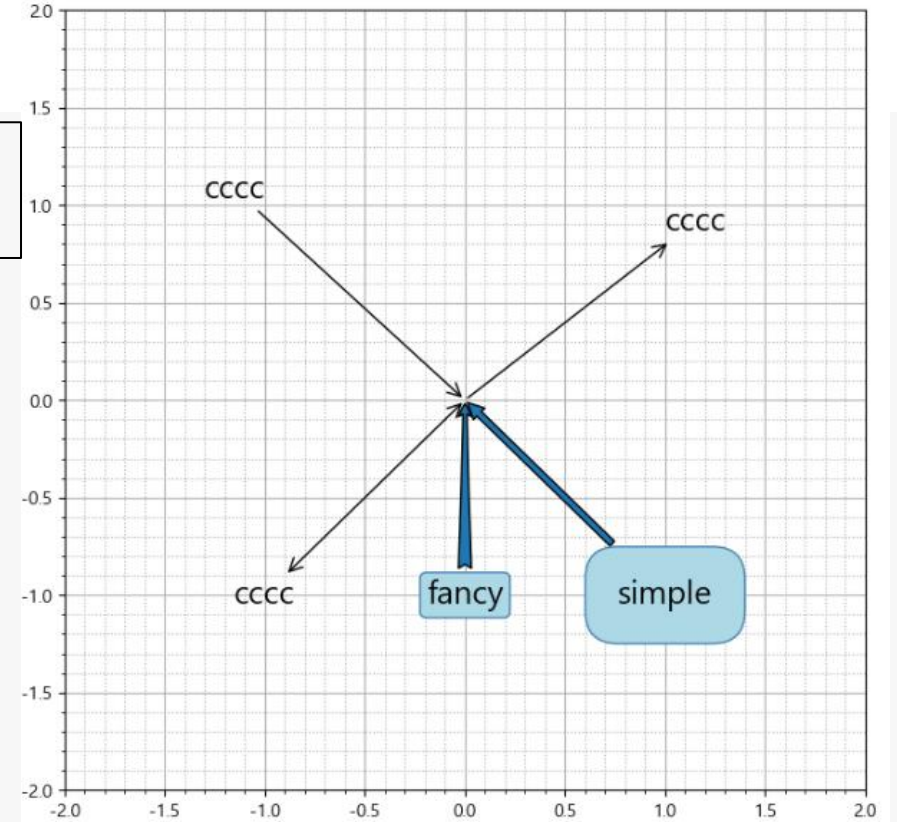
```
ax1.grid(which='major')
ax1.grid(which='minor', ls=':')
```

← grid 사용 방법

```
box1={'facecolor':'lightblue',
      'edgecolor':'steelblue',
      'boxstyle':'round',
      'pad':0.2}
```

```
box2={'facecolor':'lightblue',
      'edgecolor':'steelblue',
      'boxstyle':'round',
      'pad':1}
```

```
ax1.annotate('cccc', xy=(0,0), xytext=(-1,1), ha='right', va='bottom', arrowprops={'arrowstyle': '->'}, fontsize=18)
ax1.annotate('cccc', xy=(0,0), xytext=(1,1), ha='left', va='top', arrowprops={'arrowstyle': '<-'}, fontsize=18)
ax1.annotate('cccc', xy=(0,0), xytext=(-1,-1), ha='center', va='center', arrowprops={'arrowstyle': '<->'}, fontsize=18)
ax1.annotate('fancy', xy=(0,0), xytext=(0,-1), ha='center', va='center', arrowprops={'arrowstyle': 'fancy'}, fontsize=18, bbox=box1)
ax1.annotate('simple', xy=(0,0), xytext=(1,-1), ha='center', va='center', arrowprops={'arrowstyle': 'simple'}, fontsize=18, bbox=box2)
```



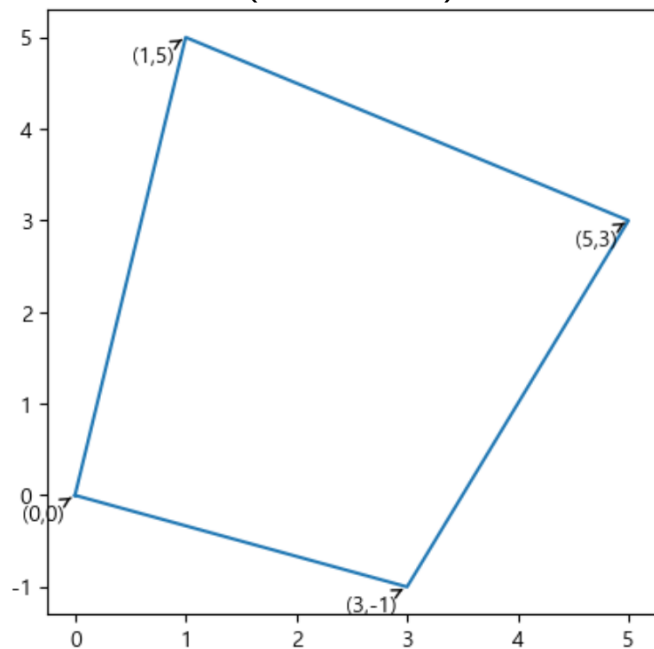
실습2

아래그림과 같이 각 모서리를 가르키는 화살표와 모서리의 좌표를 넣으시오

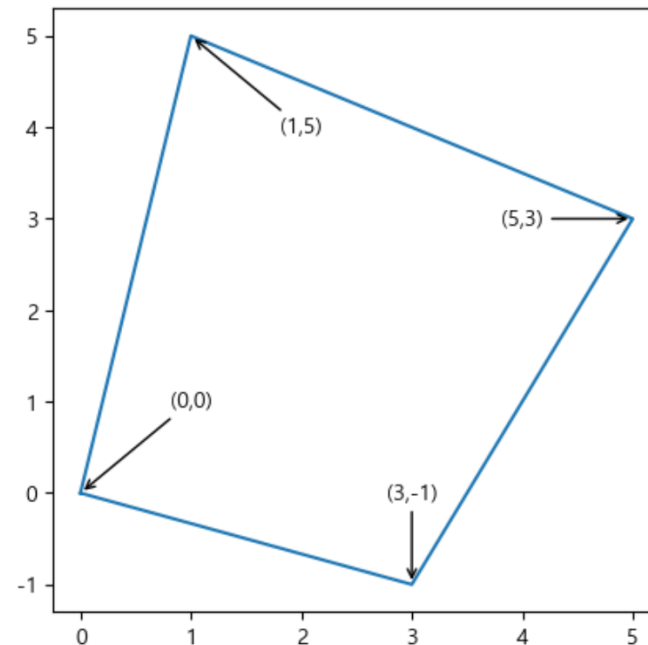
```
fig=plt.figure(figsize=(5,5), dpi=100)
ax1=fig.subplots()

X=[0,1,5,3]
Y=[0,5,3,-1]
_=ax1.plot(X+X[:1],Y+Y[:1])
```

(실습 2-1)



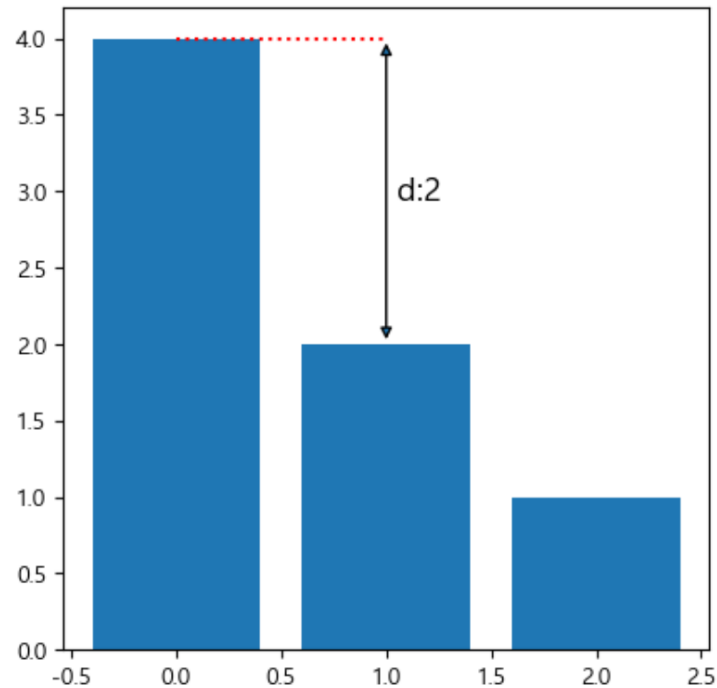
(실습 2-2, 모서리의 좌표가 모두 안쪽, text의 좌표를 따로 지정)



실습3

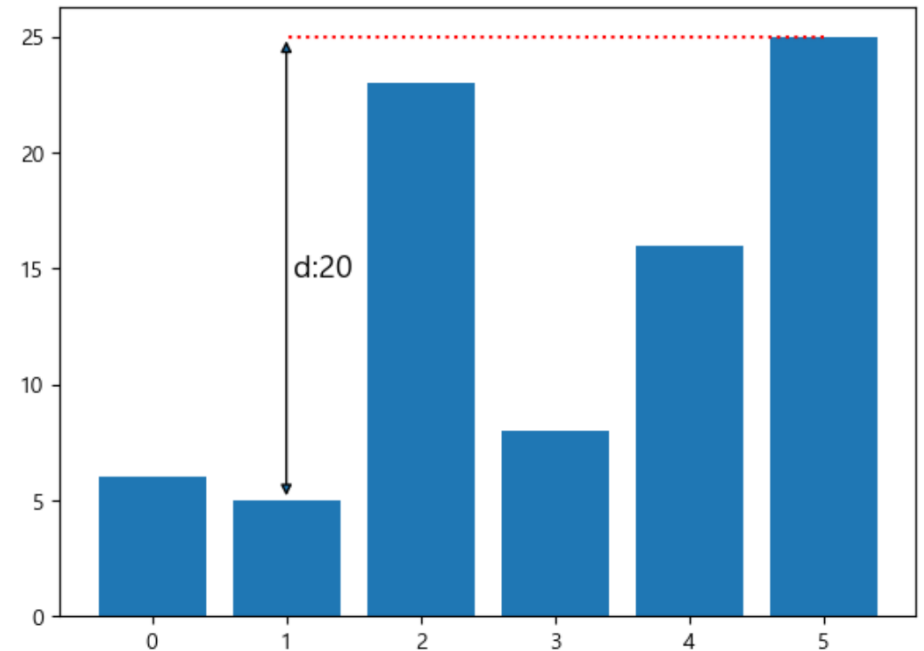
아래 그림과 같이 첫번째 바와 두번째 바의 차이를 두번째 바 위에 표시하시오 (힌트: `annotate`와 `text`를 따로따로 사용)

```
d1=[4,2,1]
```



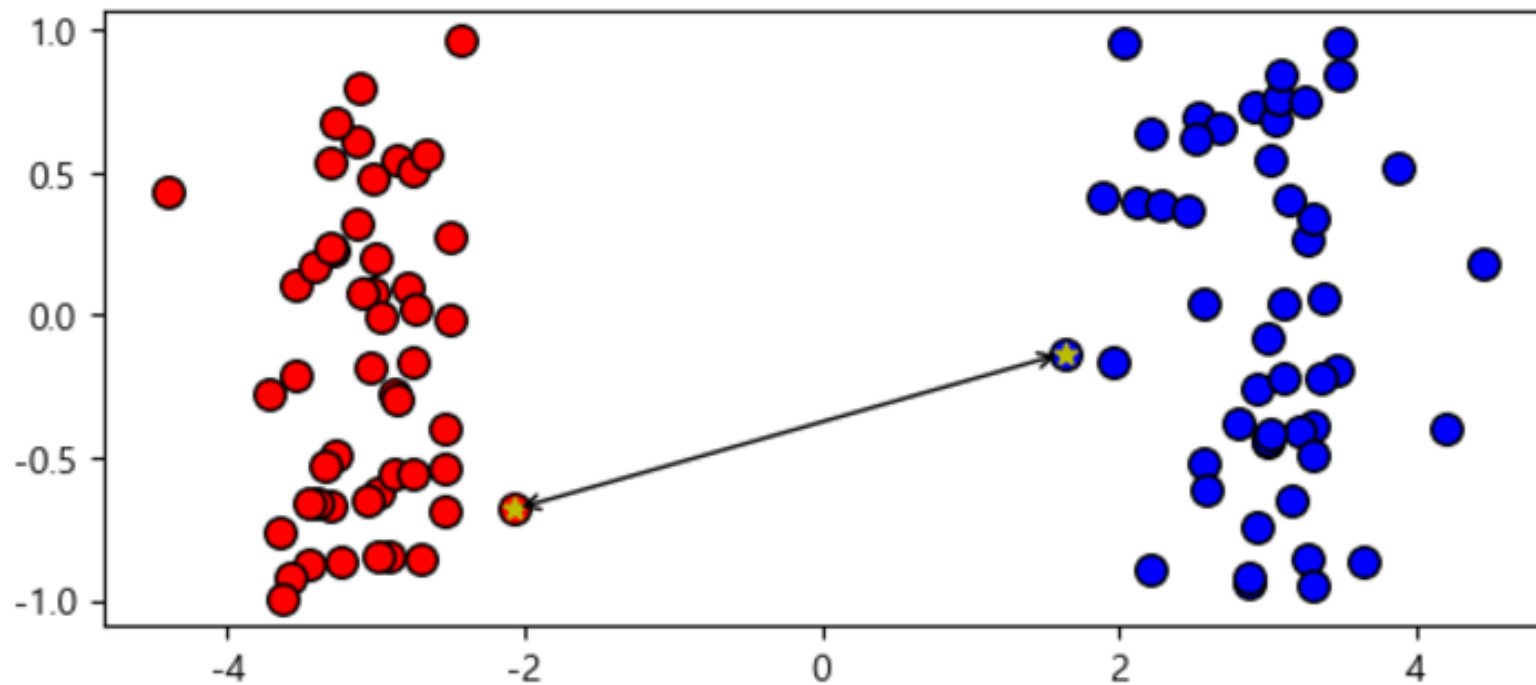
아래 그림과 같이 가장 큰 바와 가장 작은 바의 차이를 가장 작은 바 위에 표시하시오

```
d1=pd.Series(np.random.randint(1,30,6))
```



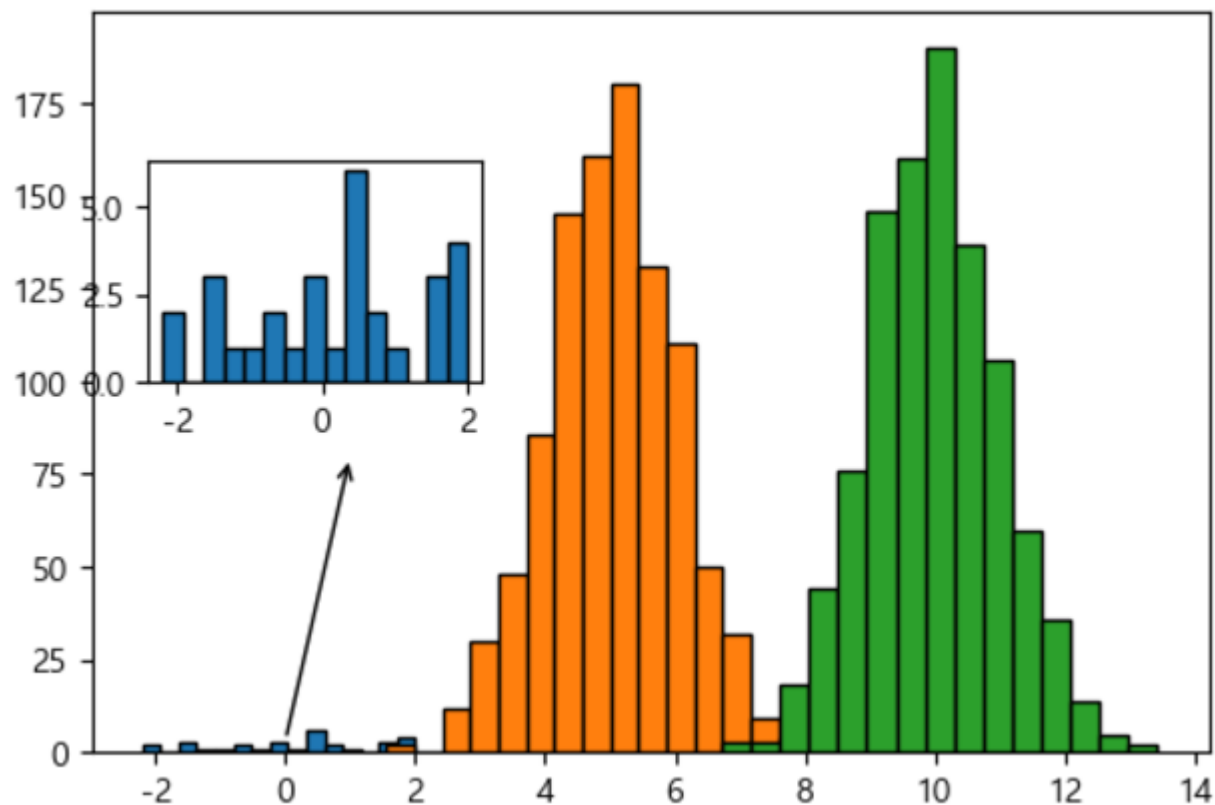
실습4

```
# df1의 x,y를 빨간색으로, df2의 x,y를 파란색으로 scatter 하였다.  
# df1의 점 중 가장 오른쪽에 있는 점을 p1이라고하고, df2의 점 중 가장 왼쪽에 있는 점을 p2라고 할때,  
# 1) p1, p2에 추가로 노란 별표를 scatter 하고  
# 2) p1,p2를 화살표로 연결하시오  
  
# - A: marker='*', color='r'  
# - B: marker='o', color='b'
```



실습5

```
# 주어진 d1, d2, d3를 ax1에 histogram을 그렸더니, d1이 잘 보이지 않는다.  
# d1의 히스토그램을 부그림에 따로 그리시오  
# - 부그림은 주어진 화살표 방향에 넣으시오
```



heatmap

heatmap

- ax.pcolor 사용
- 주요 parameter: 2차원 데이터, colormap

```
fig=plt.figure(figsize=(10,5), dpi=100)
axs=fig.subplots(2,2).flatten()
```

```
ar1=np.arange(0,24).reshape(4,6)
ar2=np.arange(-12,12).reshape(4,6)
```

ar1
ar2

Dataframe도 가능하지만 columns, index 표시 안됨

(2차원 데이터, colormap)

```
axs[0].pcolor(ar1, cmap='Purples')
axs[1].pcolor(ar2, cmap='Purples', edgecolor='k')
```

```
axs[2].pcolor(ar1, cmap='Pastel1', edgecolor='k')
axs[3].pcolor(ar2, cmap='seismic')
```

행, 열

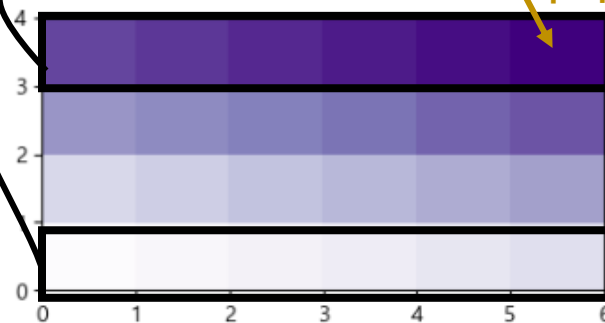
x, y

ar1[0,1] 데이터 → ax의 (1,0)에 위치
ar1[0,2] 데이터 → ax의 (2,0)에 위치
...
ar1[r1, c1] 데이터 → ax의 (c1, r1)에 위치

주어진 데이터 중 상대적인 크기로
해당 color map의 색깔이 결정

ar1

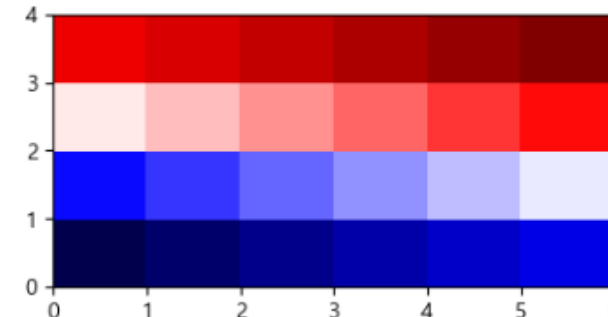
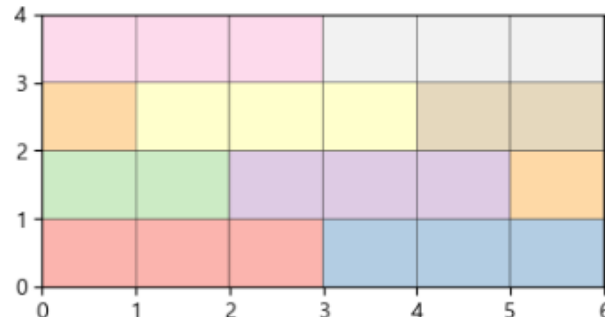
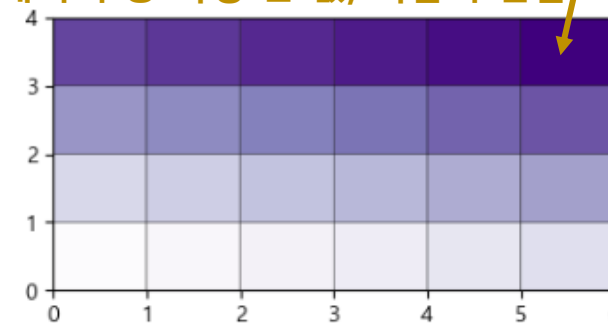
```
array([[ 0,  1,  2,  3,  4,  5],
       [ 6,  7,  8,  9, 10, 11],
       [12, 13, 14, 15, 16, 17],
       [18, 19, 20, 21, 22, 23]])
```



ar2

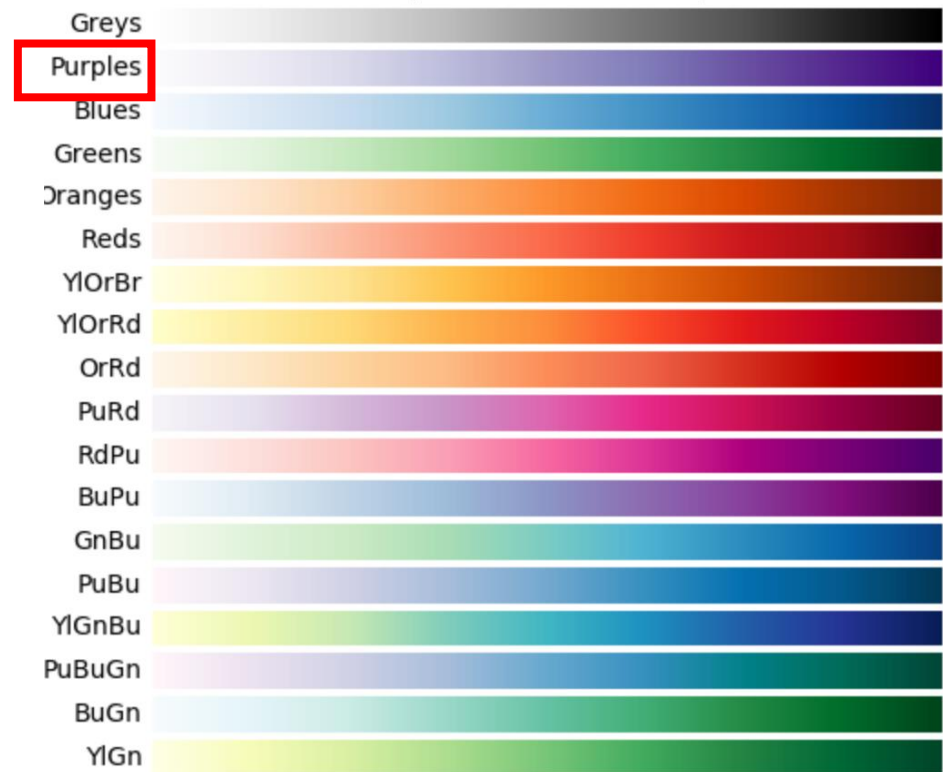
```
array([[ -12, -11, -10,  -9,  -8,  -7],
       [  -6,  -5,  -4,  -3,  -2,  -1],
       [   0,   1,   2,   3,   4,   5],
       [   6,   7,   8,   9,  10,  11]])
```

주어진 데이터 중 가장 큰 값, 색깔이 같음



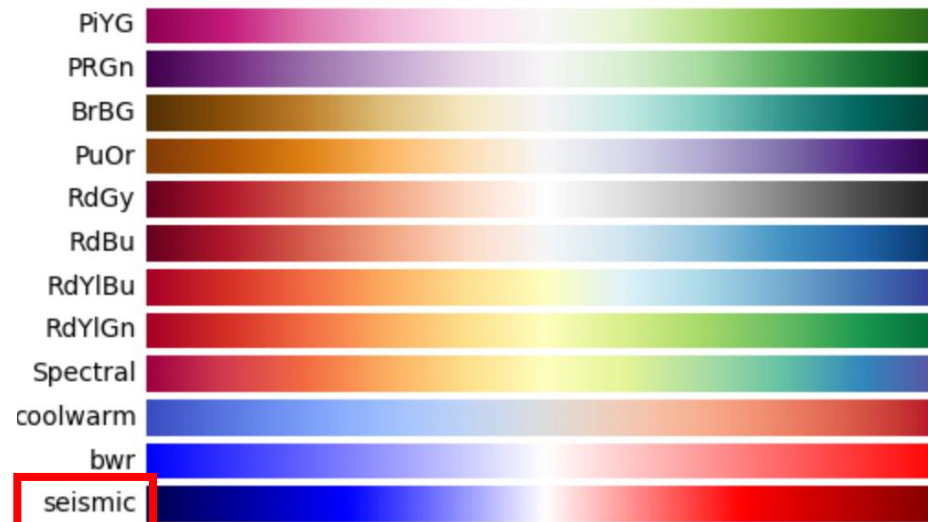
colormap

Sequential colormaps

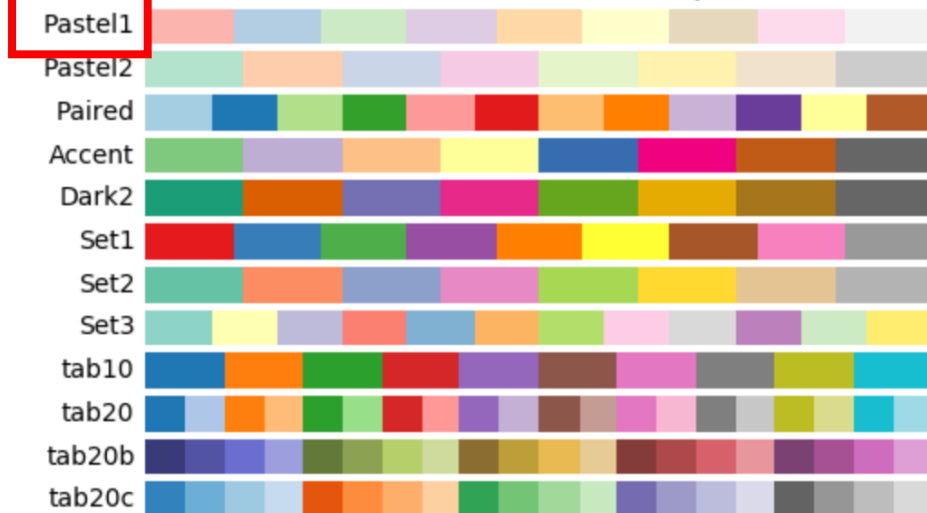


작은값 → 큰 값

Diverging colormaps



Qualitative colormaps



작은값 → 큰 값

heatmap – colorbar 추가

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

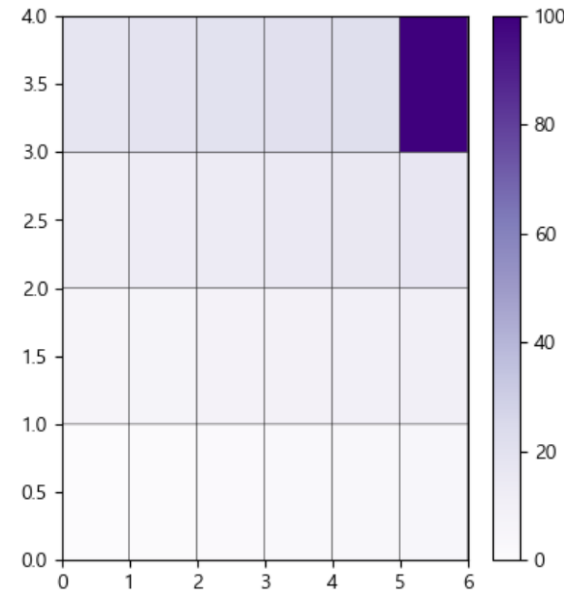
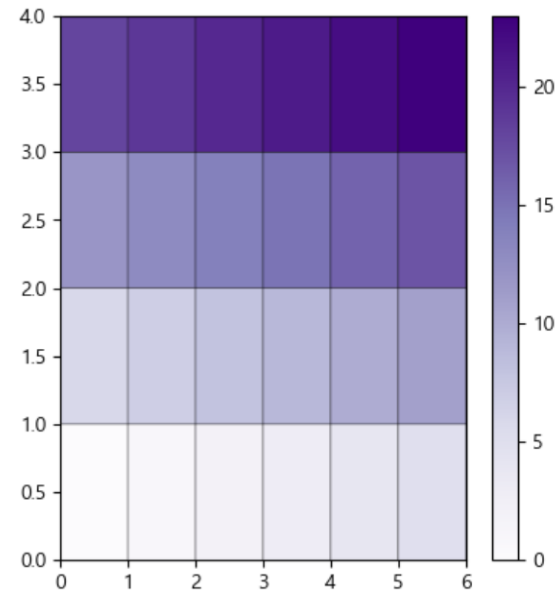
ar1=np.arange(0,24).reshape(4,6)
ar1
hm1=axs[0].pcolor(ar1, cmap='Purples', edgecolor='k')
fig.colorbar(hm1, ax=axs[0])

ar1[-1,-1]=100
ar1
hm2=axs[1].pcolor(ar1, cmap='Purples', edgecolor='k')
fig.colorbar(hm2, ax=axs[1])
```

colorbar 추가하는 부분:
pcolor함수의 return값과 해당 ax를 parameter로 받음

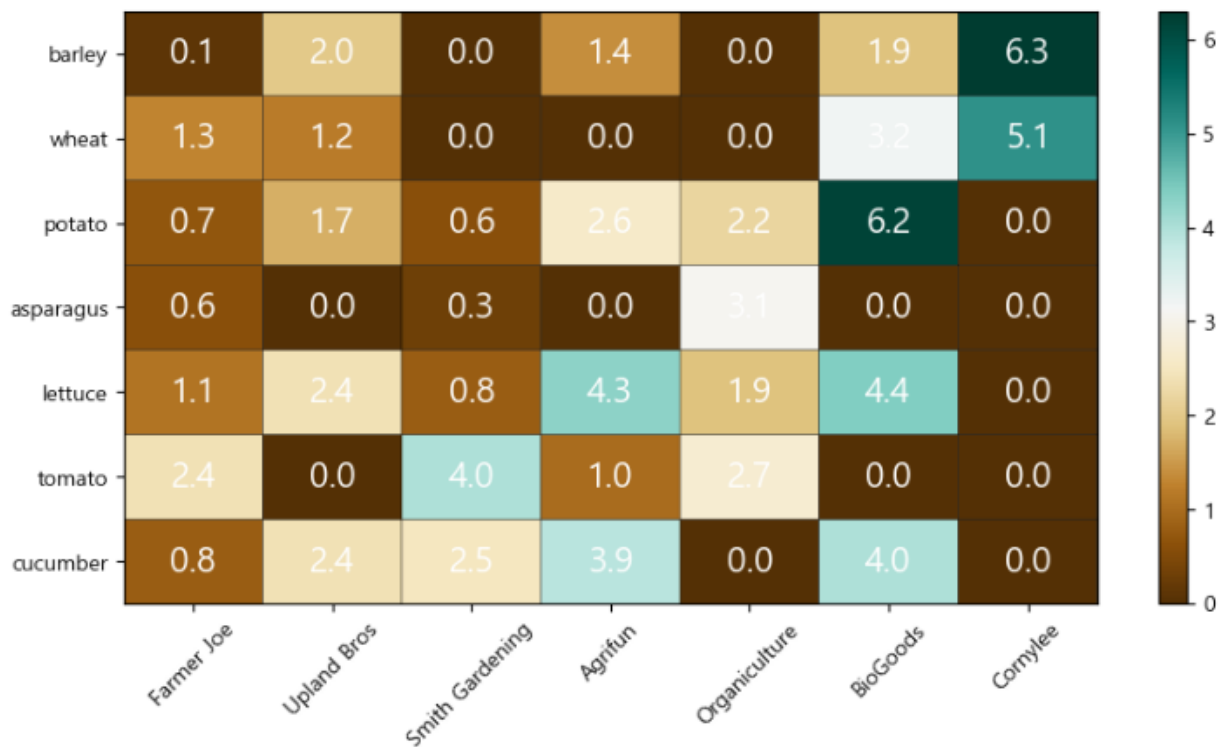
주어진 데이터 중
상대적인 크기로 해당 color map의 색깔이 결정

```
array([[ 0,  1,  2,  3,  4,  5],
       [ 6,  7,  8,  9, 10, 11],
       [12, 13, 14, 15, 16, 17],
       [18, 19, 20, 21, 22, 23]])
array([[ 0,  1,  2,  3,  4,  5],
       [ 6,  7,  8,  9, 10, 11],
       [12, 13, 14, 15, 16, 17],
       [18, 19, 20, 21, 22, 100]])
```



실습6

- 13w_d1.txt 파일을 읽고 아래와 같이 heatmap을 그리시오



13w_d1.txt

	Farmer Jo	Upland Br	Smith Gar	Agrifun	Organicul	BioGoods	Cornylee
cucumber	0.8	2.4	2.5	3.9	0	4	0
tomato	2.4	0	4	1	2.7	0	0
lettuce	1.1	2.4	0.8	4.3	1.9	4.4	0
asparagus	0.6	0	0.3	0	3.1	0	0
potato	0.7	1.7	0.6	2.6	2.2	6.2	0
wheat	1.3	1.2	0	0	0	3.2	5.1
barley	0.1	2	0	1.4	0	1.9	6.3

- seaborn heatmap 활용해보기

folium 지도를 활용한 서울의 스타벅스 매장 시각화

데이터 로드

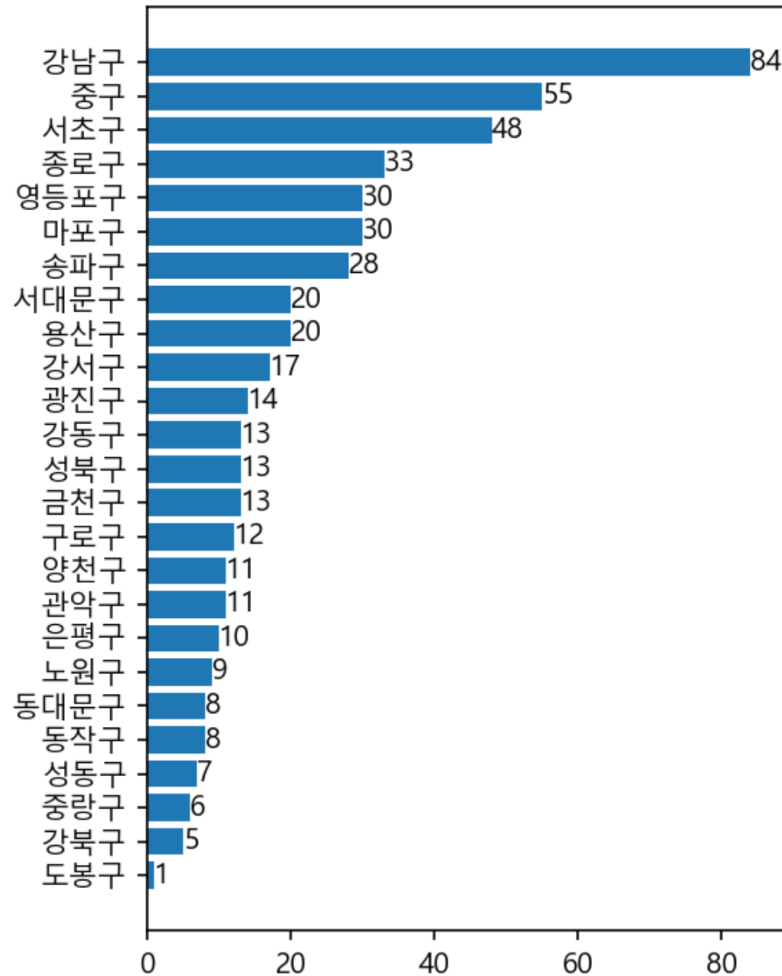
```
## data load
df = pd.read_csv('data/seoul_starbucks.txt', sep='|')
df
```

	상호명	시도명	시군구명	경도	위도
0	스타벅스	서울특별시	용산구	126.994781	37.534303
1	스타벅스종로3가점	서울특별시	종로구	126.990207	37.570585
2	스타벅스	서울특별시	서초구	127.019763	37.513663
3	스타벅스커피여의도IFC1F	서울특별시	영등포구	126.924863	37.525172
4	스타벅스	서울특별시	강남구	127.063878	37.510038
...
501	스타벅스	서울특별시	용산구	126.965220	37.527452
502	스타벅스	서울특별시	관악구	126.952792	37.479298
503	스타벅스	서울특별시	성동구	127.015329	37.548028
504	스타벅스	서울특별시	서대문구	126.956144	37.560080
505	스타벅스	서울특별시	서대문구	126.911859	37.582341

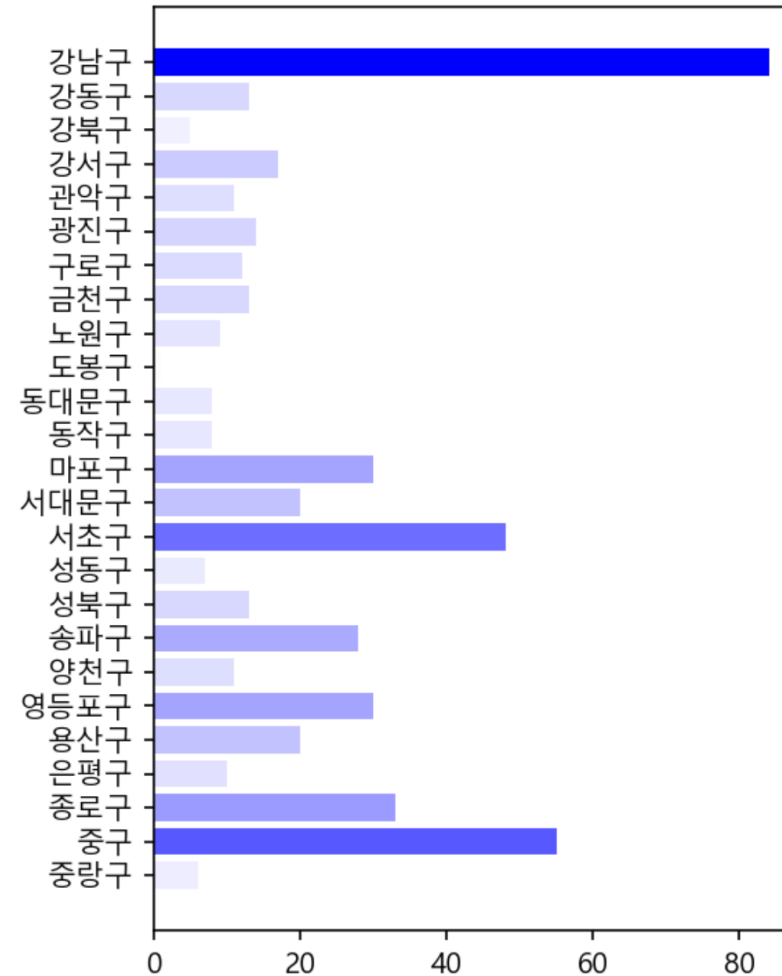
506 rows × 5 columns

실습7 (구별 매장의 개수를 bar 그래프로 나타내기)

- 매장 많은 순으로 정렬
- 매장수를 text로 표현

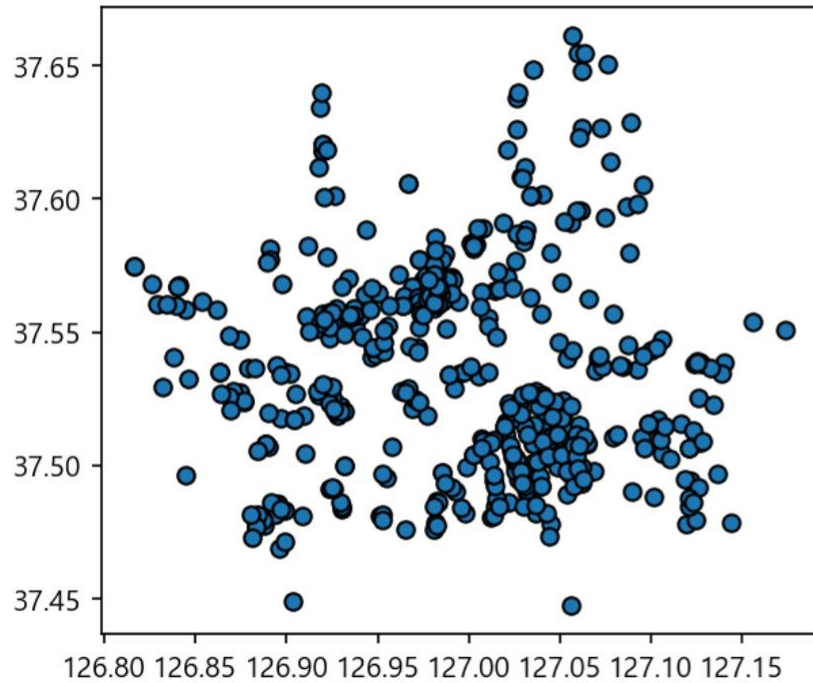


- 구의 이름 순으로 정렬
- 매장수를 bar의 투명도에 반영

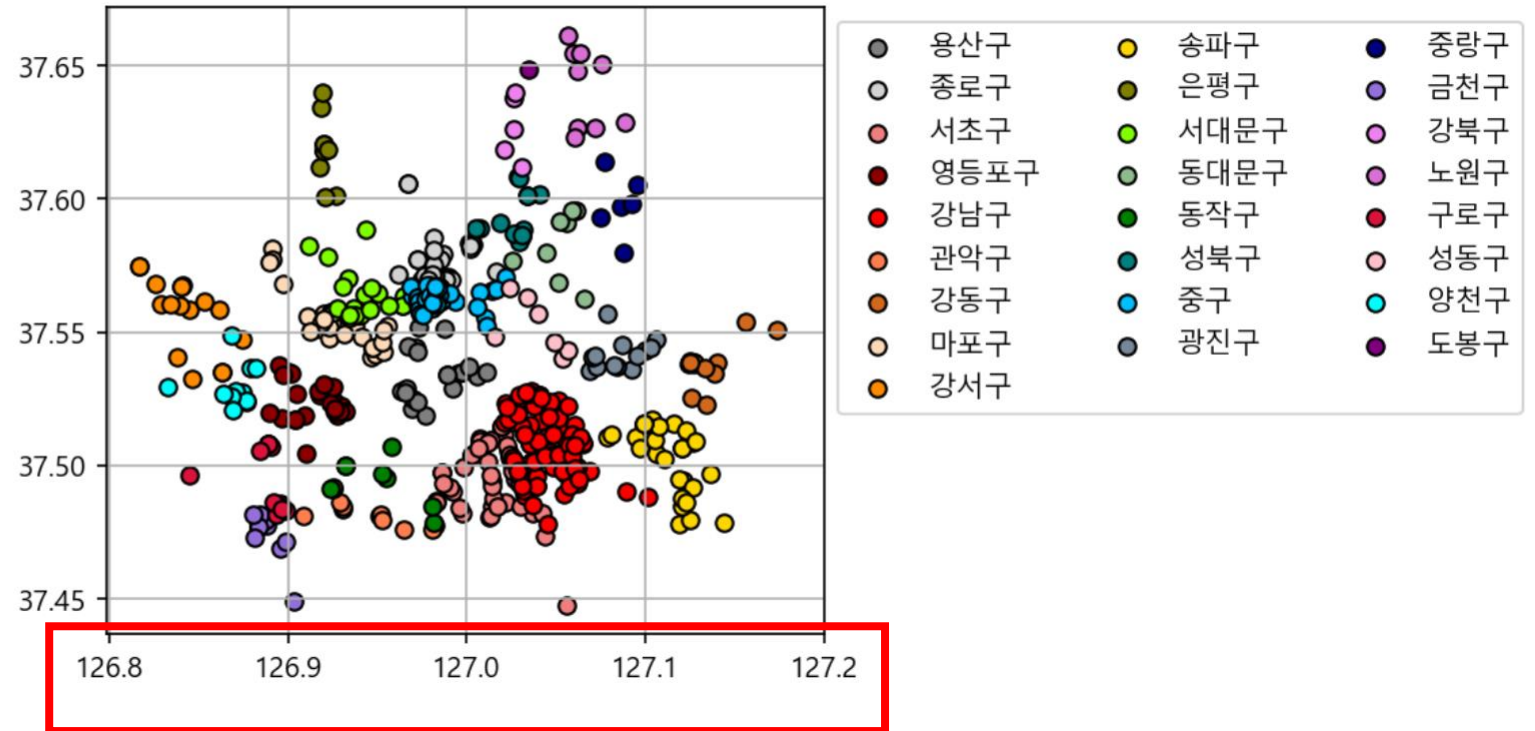


실습8 (각 매장의 위도 경도를 scatter로 나타내기)

- 매장의 위경도를 활용하여 위치scatter



- 매장의 위경도를 활용하여 위치 scatter
- 구별 다른 색깔
- Legend 추가



tick과 grid 주의

folium 활용하여 지도 위에 marker로 표현하기

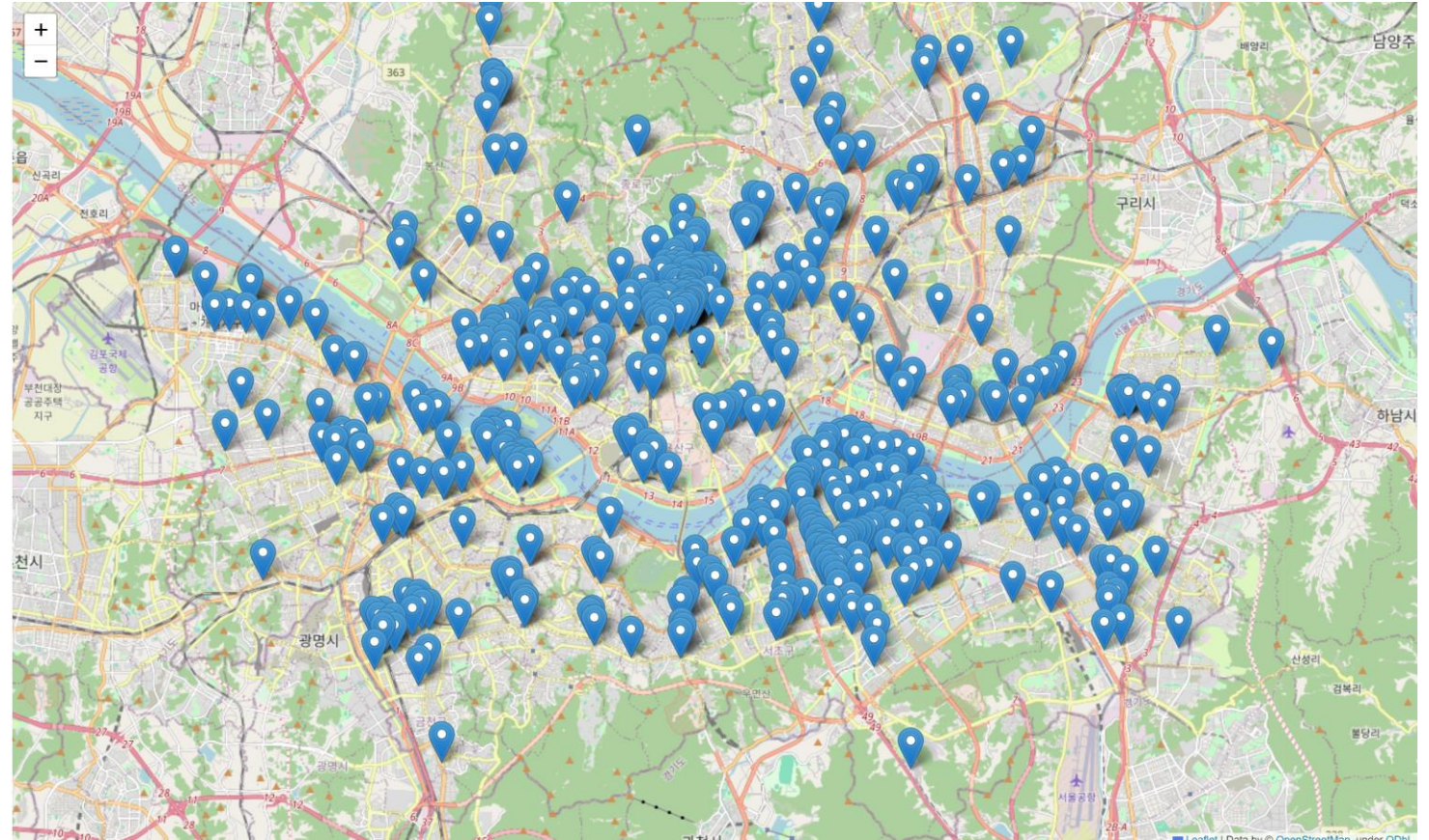
```
## data load
df = pd.read_csv('data/seoul_starbucks.txt', sep='|')
df

m= folium.Map([df["위도"].mean(), df["경도"].mean()], zoom_start=12)

for i in df.index:
    sub_lat = df.loc[i, "위도"]
    sub_long = df.loc[i, "경도"]

    title = "{}".format(df.loc[i, '상호명'])

    _=folium.Marker([sub_lat, sub_long], tooltip=title).add_to(m)
m
m.save('folium1.html')
```



folium 활용하여 지도 위에 marker로 표현하기

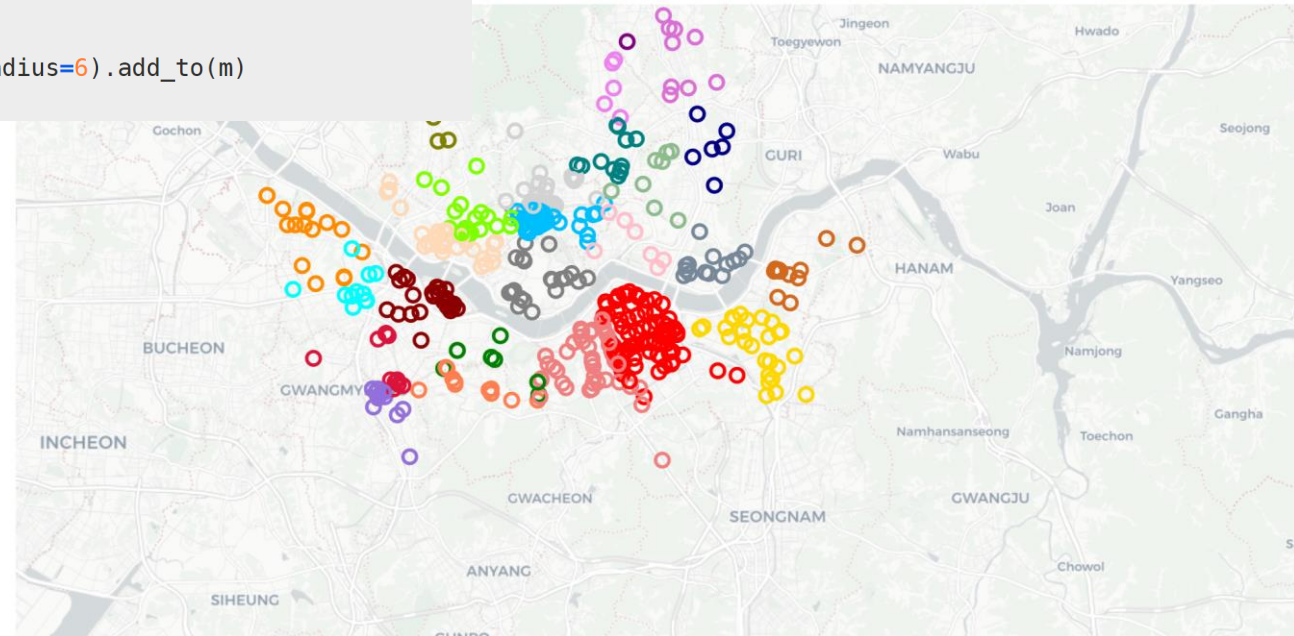
```
## data load
df = pd.read_csv('data/seoul_starbucks.txt', sep='|')
df

color_dic={ '용산구': 'gray', '종로구': 'lightgray', '서초구': 'lightcoral', '영등포구': 'darkred', '강남구': 'red',
            '관악구': 'coral', '강동구': 'chocolate', '마포구': 'peachpuff', '강서구': 'darkorange', '송파구': 'gold',
            '은평구': 'olive', '서대문구': 'chartreuse', '동대문구': 'darkseagreen', '동작구': 'green', '성북구': 'teal',
            '중구': 'deepskyblue', '광진구': 'lightslategray', '중랑구': 'navy', '금천구': 'mediumpurple', '강북구': 'violet',
            '노원구': 'orchid', '구로구': 'crimson', '성동구': 'pink', '양천구': 'cyan', '도봉구': 'purple'}

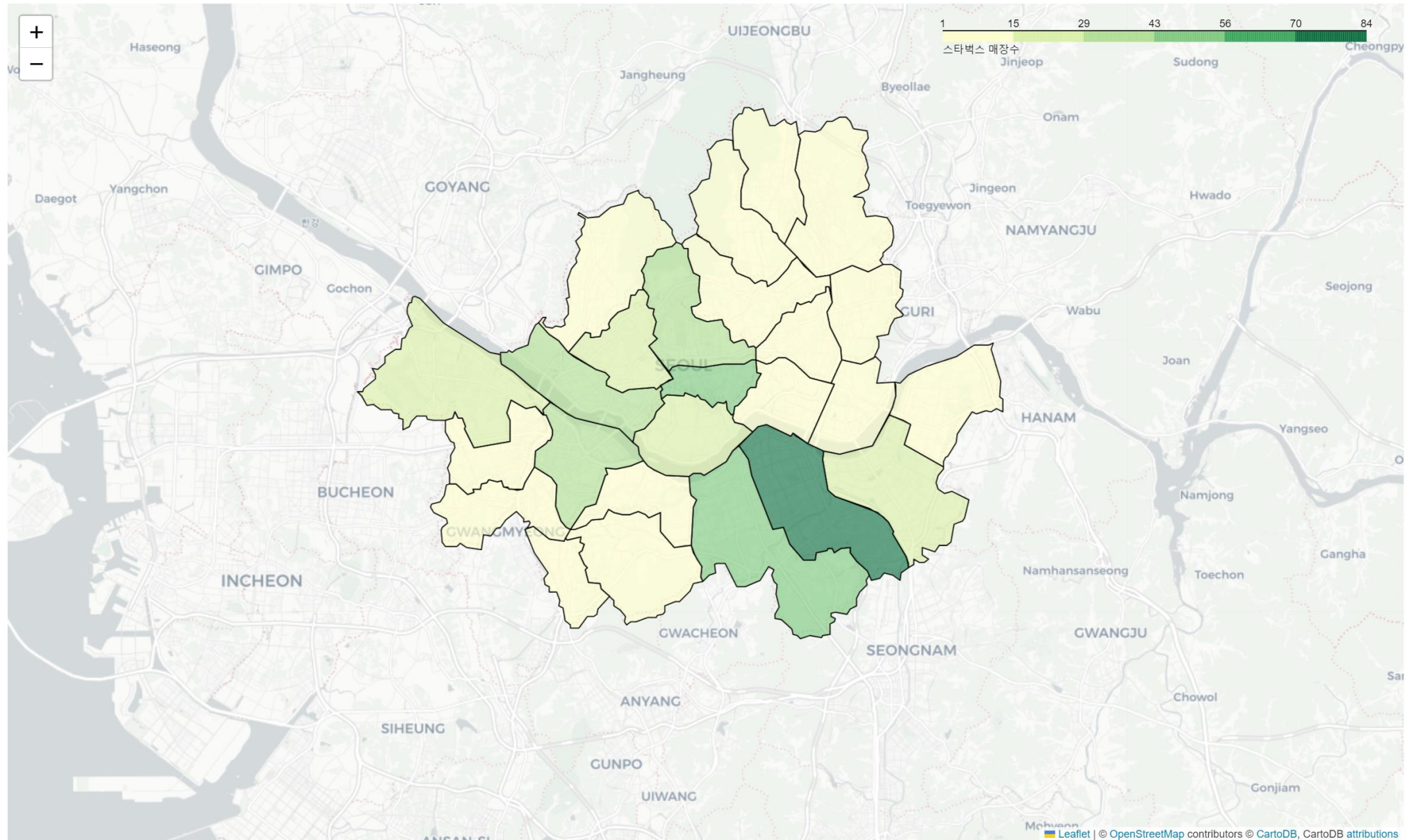
m= folium.Map([df["위도"].mean(), df["경도"].mean()], zoom_start=12, tiles='CartoDB positron')

for i in df.index:
    sub_lat = df.loc[i, "위도"]
    sub_long = df.loc[i, "경도"]
    gu=df.loc[i, '시군구명']
    title = "{}".format(df.loc[i, '상호명'])

    _=folium.CircleMarker(sub_lat, sub_long, color=color_dic[gu], tooltip=title, radius=6).add_to(m)
m
```



Choropleth 사용하여 시군구별 상점수 색깔로 표현하기



Q & A

Thank you