

Matplotlib 라이브러리 IRIS 데이터 셋 실습해보기

학습 내용

- 그래프 - 레이블, 제목 등
- matplotlib - 히스토그램
- matplotlib - 산점도
- matplotlib - 막대 그래프
- subplot - 여러개의 그래프 표시하기

목차

01. 데이터 준비
02. 4개의 피쳐 값 확인 - 히스토그램
03. Scatter Plot (산점도) - 두 변수간 상관관계 확인
04. 막대 그래프(bar plot)

01 데이터 준비

목차로 이동하기

```
In [16]: import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib as mpl
import numpy as np
plt.style.use('ggplot') # R의 ggplot라이브러리와 비슷한 형태로 설정.(1.41이상)

print(pd.__version__)
print(sns.__version__)
print(mpl.__version__)

iris = sns.load_dataset("iris")
iris
```

2.1.4
0.12.2
3.8.0

Out[16]:

	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

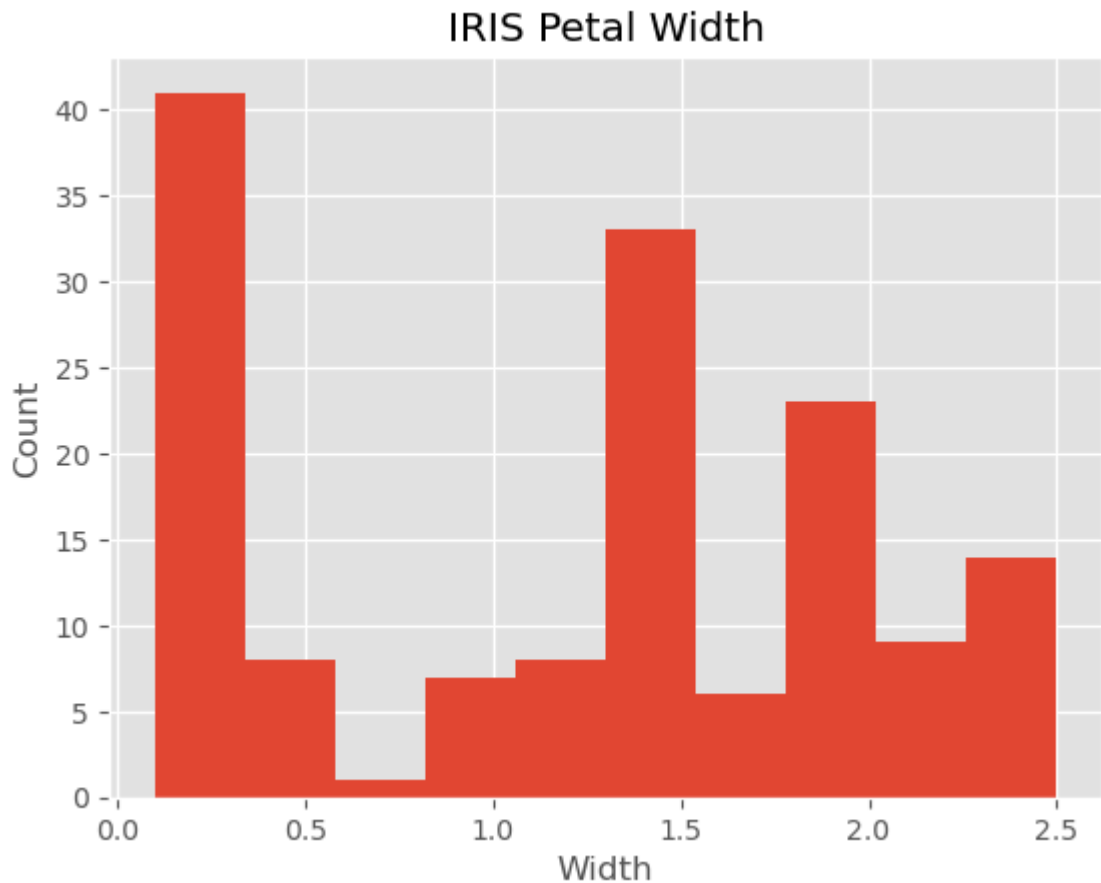
150 rows × 5 columns

02. 4개의 피쳐 값 확인 - 히스토그램

목차로 이동하기

```
In [17]: plt.title("IRIS Petal Width") # 제목
plt.xlabel("Width") # x 레이블
plt.ylabel("Count") # y 레이블
plt.hist(iris['petal_width'])
```

```
Out[17]: (array([41.,  8.,  1.,  7.,  8., 33.,  6., 23.,  9., 14.]),
array([0.1 , 0.34, 0.58, 0.82, 1.06, 1.3 , 1.54, 1.78, 2.02, 2.26, 2.5 ]),
<BarContainer object of 10 artists>)
```

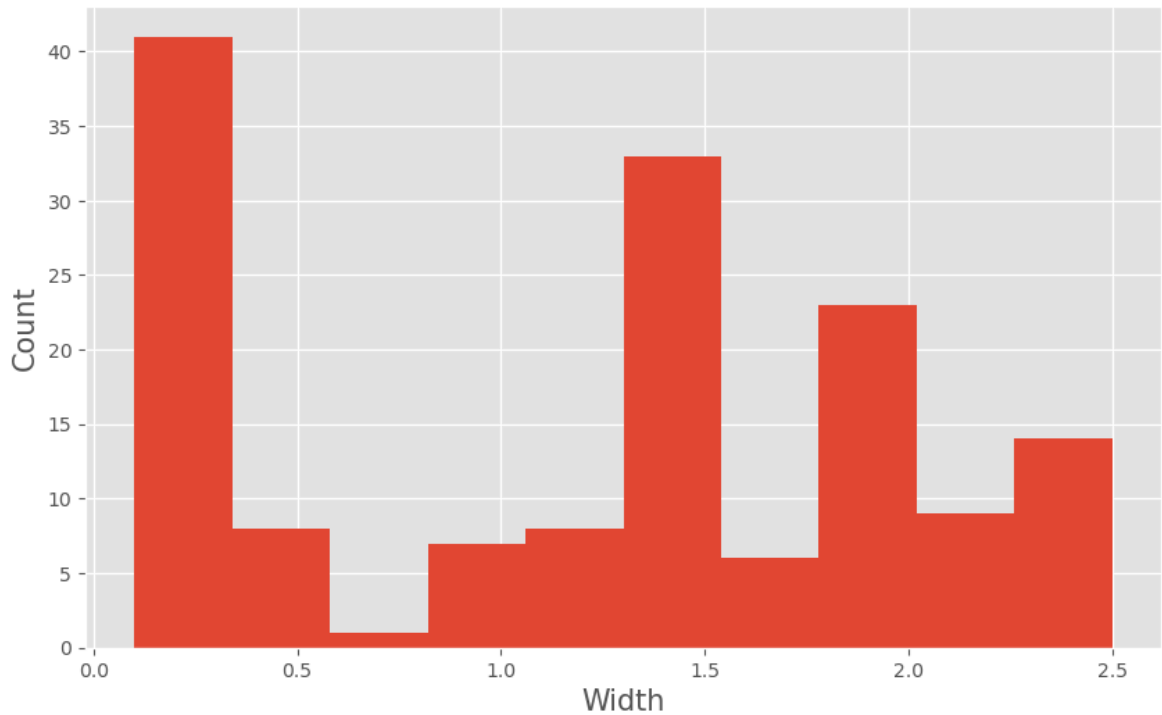


그래프의 크기 설정 및 세부 설정

```
In [18]: plt.figure(figsize=(10,6))
plt.title("IRIS Petal Width", fontsize=25, y=1.01) # y : 그래프로부터 제목까지
plt.xlabel("Width", fontsize=15)                # x 레이블
plt.ylabel("Count", fontsize=15)                 # y 레이블
plt.hist(iris['petal_width'])
```

```
Out[18]: (array([41.,  8.,  1.,  7.,  8., 33.,  6., 23.,  9., 14.]),
array([0.1 , 0.34, 0.58, 0.82, 1.06, 1.3 , 1.54, 1.78, 2.02, 2.26, 2.5 ]),
<BarContainer object of 10 artists>)
```

IRIS Petal Width



```
In [19]: iris.columns
```

```
Out[19]: Index(['sepal_length', 'sepal_width', 'petal_length', 'petal_width',  
              'species'],  
             dtype='object')
```

```
In [20]: fig, ax = plt.subplots(2,2, figsize=(12,8))

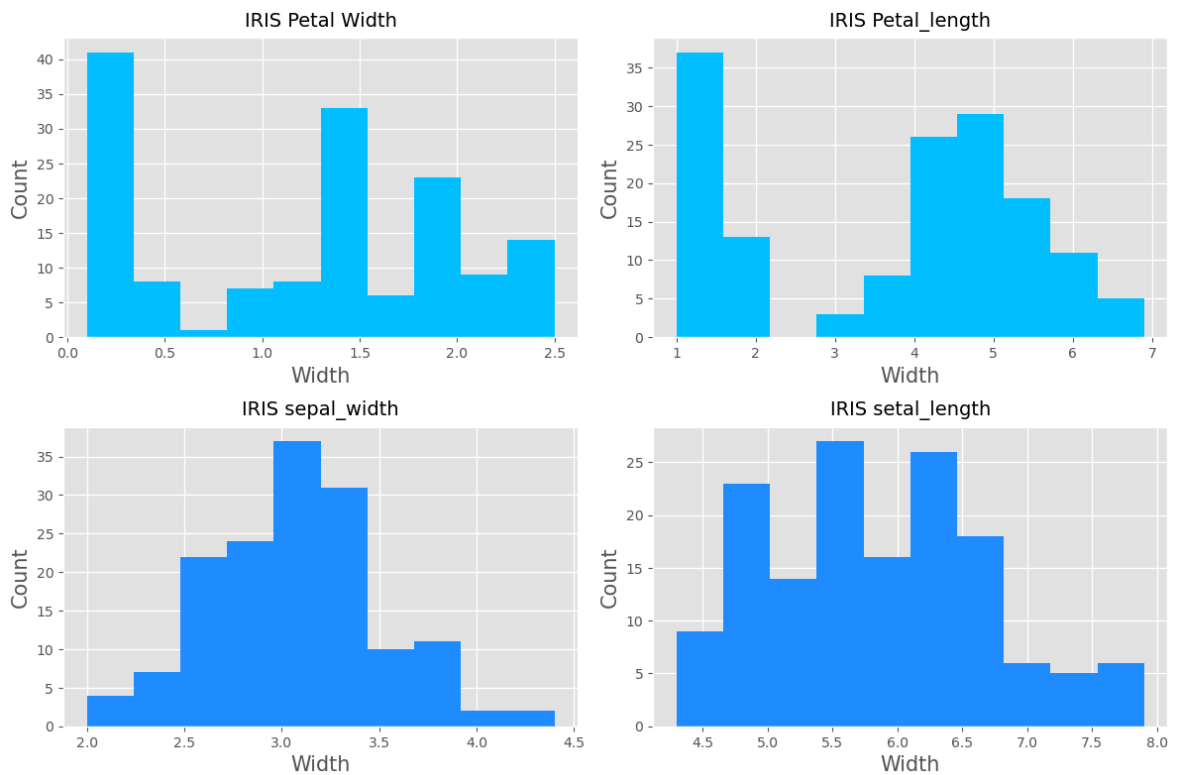
ax[0][0].hist(iris['petal_width'], color='deepskyblue')
ax[0][0].set_ylabel('Count', fontsize=15)
ax[0][0].set_xlabel('Width', fontsize=15)
ax[0][0].set_title("IRIS Petal Width", fontsize=14, y=1.01)

ax[0][1].hist(iris['petal_length'], color='deepskyblue')
ax[0][1].set_ylabel('Count', fontsize=15)
ax[0][1].set_xlabel('Width', fontsize=15)
ax[0][1].set_title("IRIS Petal_length", fontsize=14, y=1.01)

ax[1][0].hist(iris['sepal_width'], color='dodgerblue')
ax[1][0].set_ylabel('Count', fontsize=15)
ax[1][0].set_xlabel('Width', fontsize=15)
ax[1][0].set_title("IRIS sepal_width", fontsize=14, y=1.01)

ax[1][1].hist(iris['sepal_length'], color='dodgerblue')
ax[1][1].set_ylabel('Count', fontsize=15)
ax[1][1].set_xlabel('Width', fontsize=15)
ax[1][1].set_title("IRIS setal_length", fontsize=14, y=1.01)

plt.tight_layout() # 서브 플롯이 겹치지 않도록 자동으로 조절해준다.
```



03. Scatter Plot (산점도) - 두 변수간 상관관계 확인

- width와 length의 관계 조사

목차로 이동하기

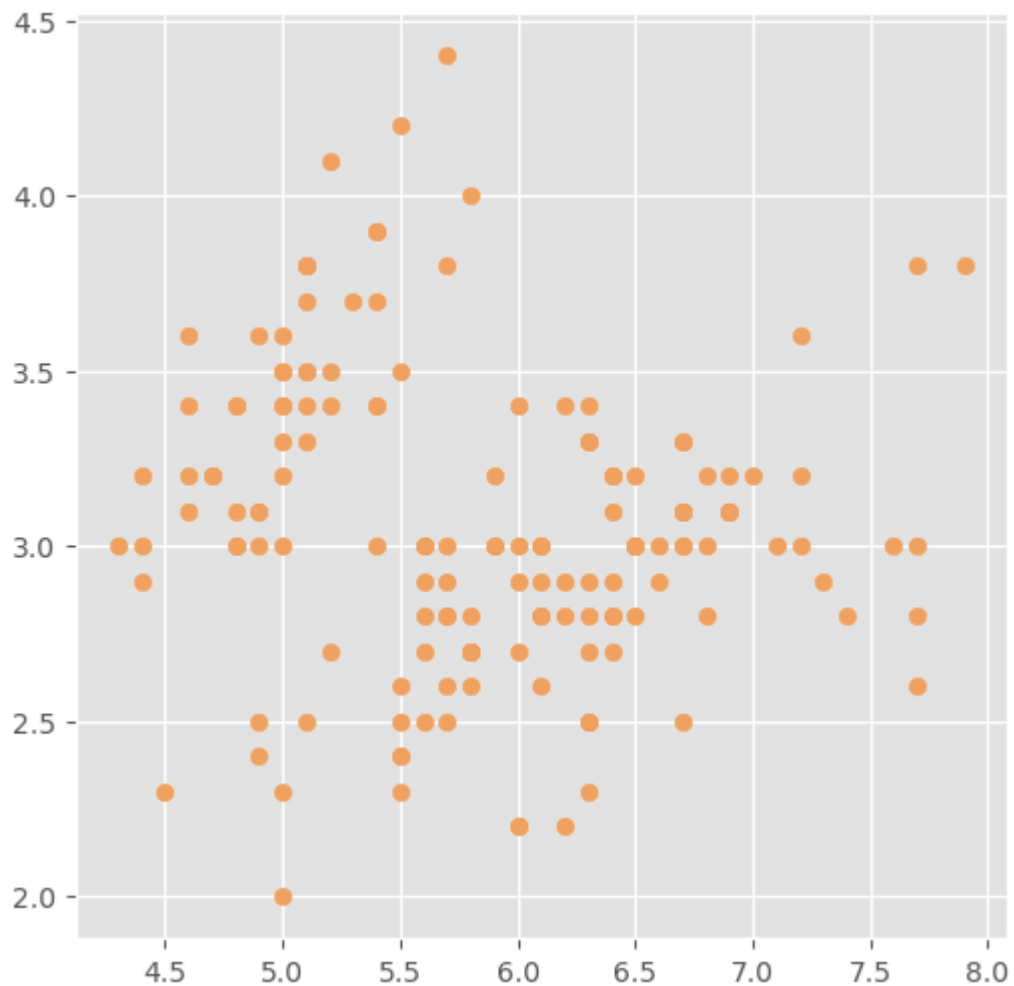
```
In [21]: iris.columns
```

```
Out[21]: Index(['sepal_length', 'sepal_width', 'petal_length', 'petal_width',
               'species'],
              dtype='object')
```

```
In [22]: fig, ax = plt.subplots( figsize=(6,6) )
```

```
ax.scatter( iris['sepal_length'], iris['sepal_width'], color="sandybrown")
```

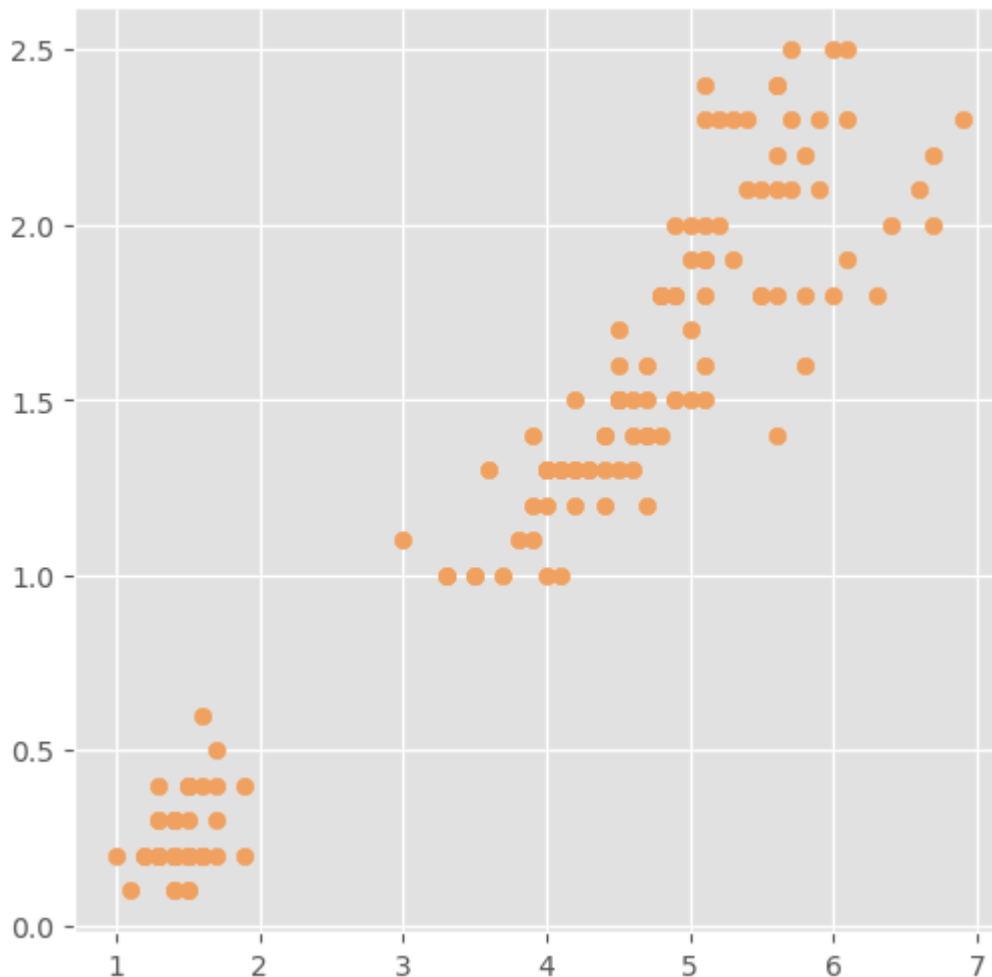
```
Out[22]: <matplotlib.collections.PathCollection at 0x24f66a94b50>
```



실습 - petal_width와 length의 관계를 확인해 보자.

```
In [23]: fig, ax = plt.subplots( figsize=(6,6) )  
  
ax.scatter( iris['petal_length'], iris['petal_width'], color="sandybrown")
```

```
Out[23]: <matplotlib.collections.PathCollection at 0x24f66b52fd0>
```



- 추가 - label과 title를 넣어보기

04. 막대 그래프

목차로 이동하기

```
In [33]: cols = list( iris.columns )[0:4]
cols
```

```
Out[33]: ['sepal_length', 'sepal_width', 'petal_length', 'petal_width']
```

```
In [34]: iris.species.unique()
```

```
Out[34]: array(['setosa', 'versicolor', 'virginica'], dtype=object)
```

```
In [35]: # virginica인 것들의 4개의 피쳐 평균
[ iris[ iris.species == 'virginica'][one].mean() for one in cols ]
```

```
Out[35]: [6.5879999999999998, 2.974, 5.5520000000000005, 2.0260000000000002]
```

```
In [36]: # setosa인 것들의 4개의 피쳐 평균
[ iris[ iris.species == 'setosa'][one].mean() for one in cols ]
```

```
Out[36]: [5.006, 3.428, 1.4620000000000002, 0.24599999999999997]
```

```
In [37]: # versicolor인 것들의 4개의 피쳐 평균
[ iris[ iris.species == 'versicolor' ][one].mean() for one in cols ]
```

```
Out[37]: [5.936, 2.7700000000000005, 4.26, 1.3259999999999998]
```

```
In [40]: # x축 레이블 정의
x = cols
```

```
In [41]: # setosa인 것들의 4개의 피쳐 평균
setosa = [ iris[ iris.species == 'setosa' ][one].mean() for one in cols ]
```

```
In [42]: # setosa인 것들의 4개의 피쳐 평균
setosa = [ iris[ iris.species == 'setosa' ][one].mean() for one in cols ]
# versicolor인 것들의 4개의 피쳐 평균
versi = [ iris[ iris.species == 'versicolor' ][one].mean() for one in cols ]
# virginica인 것들의 4개의 피쳐 평균
virgi = [ iris[ iris.species == 'virginica' ][one].mean() for one in cols ]
```

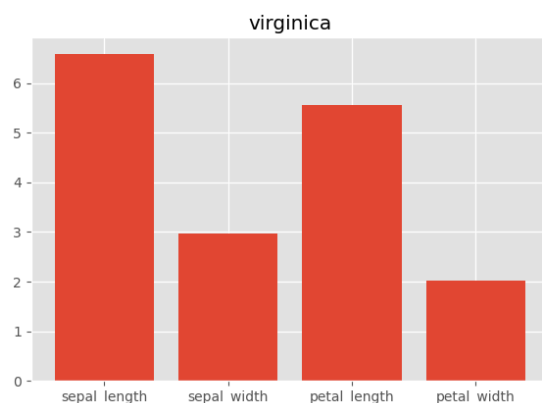
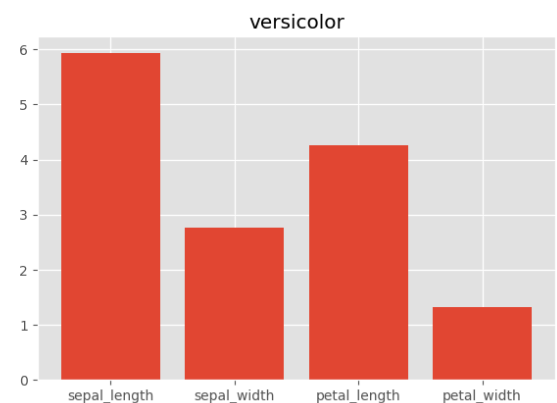
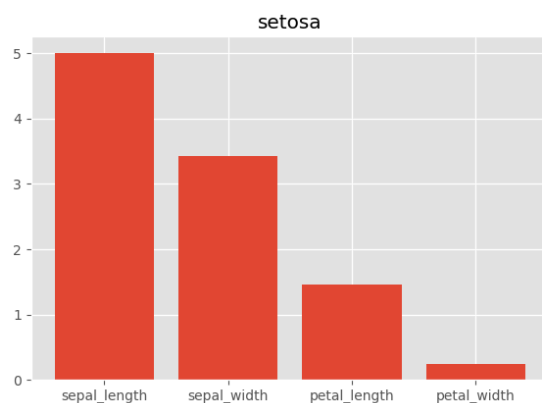
```
In [43]: plt.figure(figsize=(15,10))
```

```
plt.subplot(2,2,1)
plt.bar( x, setosa )
plt.title("setosa")
```

```
plt.subplot(2,2,2)
plt.bar( x, versi )
plt.title("versicolor")
```

```
plt.subplot(2,2,3)
plt.bar( x, virgi )
plt.title("virginica")
```

```
Out[43]: Text(0.5, 1.0, 'virginica')
```



실습해 보기 1

- tips 데이터 셋 또는 자신이 원하는 데이터 셋을 불러와 시각화를 수행해 보자. 이 내용을 제출해 보자.

REF

- https://matplotlib.org/3.1.0/gallery/color/named_colors.html
- last update : 24/06, @by DJ, Lim