

In [1]:

```
from sklearn import datasets
import matplotlib.pyplot as plt
import pandas as pd
```

## 가장 연관된 특징들 추출

In [2]:

```
iris = datasets.load_iris()
X = iris.data
```

## 표준화하기

In [3]:

```
from sklearn import preprocessing

std = preprocessing.StandardScaler()
X_std = std.fit_transform(X)
X_std[:5]
```

Out[3]:

```
array([[ -0.90068117,  1.01900435, -1.34022653, -1.3154443 ],
       [-1.14301691, -0.13197948, -1.34022653, -1.3154443 ],
       [-1.38535265,  0.32841405, -1.39706395, -1.3154443 ],
       [-1.50652052,  0.09821729, -1.2833891 , -1.3154443 ],
       [-1.02184904,  1.24920112, -1.34022653, -1.3154443 ]])
```

In [4]:

```
data = pd.DataFrame(X_std, columns=iris.feature_names)
```

In [5]:

```
data
```

Out[5]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	-0.900681	1.019004	-1.340227	-1.315444
1	-1.143017	-0.131979	-1.340227	-1.315444
2	-1.385353	0.328414	-1.397064	-1.315444
3	-1.506521	0.098217	-1.283389	-1.315444
4	-1.021849	1.249201	-1.340227	-1.315444
...	...	...	...	...
145	1.038005	-0.131979	0.819596	1.448832
146	0.553333	-1.282963	0.705921	0.922303
147	0.795669	-0.131979	0.819596	1.053935
148	0.432165	0.788808	0.933271	1.448832
149	0.068662	-0.131979	0.762758	0.790671

150 rows × 4 columns

## PCA 수행

In [6]:

```
from sklearn.decomposition import PCA
```

In [7]:

```
pca = PCA(n_components = 2)
result = pca.fit_transform(data)

result_df = pd.DataFrame(result, columns = ["x", "y"])
```

## 시각화

In [8]:

```
target = pd.DataFrame(iris.target, columns=['type'])
target['type'] = target['type'].apply(lambda x: iris.target_names[x])
target
merged = pd.concat([result_df, target], axis=1)
```

In [9]:

```
import matplotlib.pyplot as plt
import matplotlib as mlp
import seaborn as sns

%matplotlib inline

# font 정의
mlp.rcParams['font.size'] = 20
mlp.rcParams['font.family'] = 'Nanum Gothic'

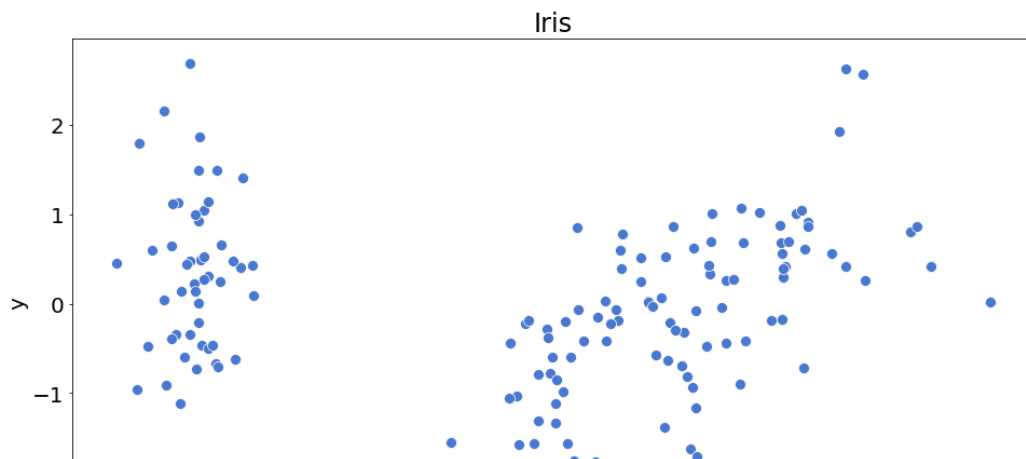
# 시각화
plt.figure(figsize=(16, 9))
sns.set_palette(sns.color_palette("muted"))

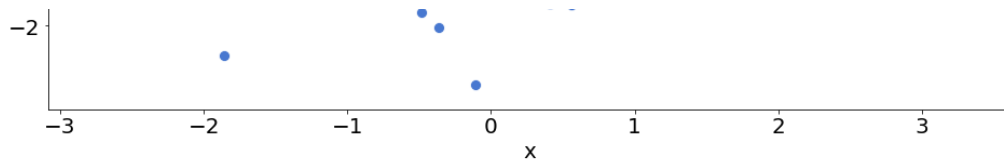
sns.scatterplot(merged['x'],
                merged['y'],

                s=100,
                palette=sns.color_palette('muted', n_colors=3),
                )

plt.title('Iris')
plt.show()
```

```
findfont: Font family ['Nanum Gothic'] not found. Falling back to D
ejaVu Sans.
findfont: Font family ['Nanum Gothic'] not found. Falling back to D
ejaVu Sans.
```





## KMeans 알고리즘 수행

In [12]:

```
from sklearn.cluster import KMeans
km = KMeans(n_clusters = 3, n_jobs=4, random_state = 1)
km.fit(result)
```

Out[12]:

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=3, n_init=10, n_jobs=4, precompute_distances='auto',
       random_state=1, tol=0.0001, verbose=0)
```

In [13]:

```
print(km.cluster_centers_)
```

```
[[-2.22475316  0.28892745]
 [ 0.57262144 -0.80720888]
 [ 1.72103664  0.60288719]]
```

In [14]:

```
new_labels = km.labels_
# Plot the identified clusters and compare with the answers
fig, axes = plt.subplots(1, 2, figsize=(16,8))
axes[0].scatter(merged['x'],
                 merged['y'],
                 c=iris.target,
                 cmap='gist_rainbow')

axes[1].scatter(result[:, 0], result[:, 1], c=new_labels, cmap='jet',
                 edgecolor='k', s=150)

axes[0].set_xlabel('Sepal length', fontsize=18)
axes[0].set_ylabel('Sepal width', fontsize=18)

axes[1].set_xlabel('Sepal length', fontsize=18)
axes[1].set_ylabel('Sepal width', fontsize=18)

axes[0].tick_params(direction='in', length=10, width=5, colors='k', labelsiz=
20)
axes[1].tick_params(direction='in', length=10, width=5, colors='k', labelsiz=
20)

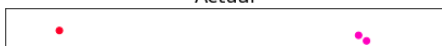
axes[0].set_title('Actual', fontsize=18)
axes[1].set_title('Predicted', fontsize=18)
```

Out[14]:

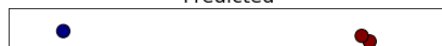
```
Text(0.5, 1.0, 'Predicted')
```

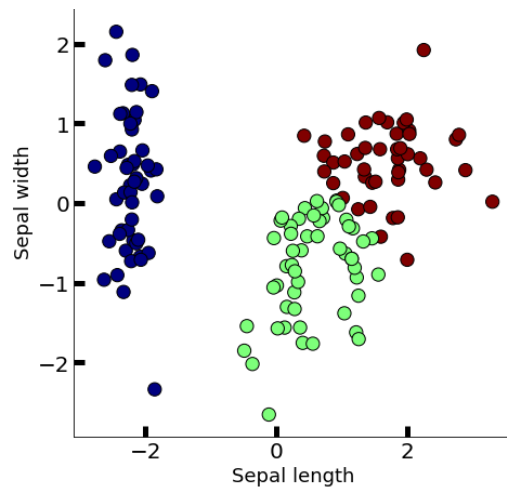
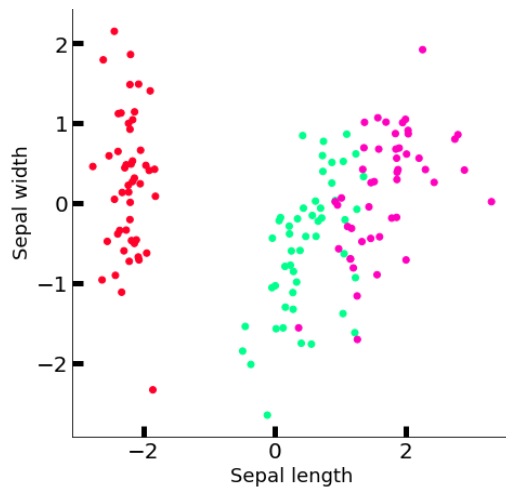
findfont: Font family ['Nanum Gothic'] not found. Falling back to DejaVu Sans.

Actual



Predicted





## hierarchical\_clustering

In [20]:

```
result
```

Out[20]:

```
array([[ -2.26470281,  0.4800266 ],
       [ -2.08096115, -0.67413356],
       [ -2.36422905, -0.34190802],
       [ -2.29938422, -0.59739451],
       [ -2.38984217,  0.64683538],
       [ -2.07563095,  1.48917752],
       [ -2.44402884,  0.0476442 ],
       [ -2.23284716,  0.22314807],
       [ -2.33464048, -1.11532768],
       [ -2.18432817, -0.46901356],
       [ -2.1663101 ,  1.04369065],
       [ -2.32613087,  0.13307834],
       [ -2.2184509 , -0.72867617],
       [ -2.6331007 , -0.96150673],
       [ -2.1987406 ,  1.86005711],
       [ -2.26221453,  2.68628449],
       [ -2.2075877 ,  1.48360936],
       [ -2.19034951,  0.48883832],
       [ -1.898572 ,  1.40501879],
       [ -2.34336905,  1.12784938],
       [ -1.914323 ,  0.40885571],
       [ -2.20701284,  0.92412143],
       [ -2.7743447 ,  0.45834367],
       [ -1.81866953,  0.08555853],
       [ -2.22716331,  0.13725446],
       [ -1.95184633, -0.62561859],
       [ -2.05115137,  0.24216355],
       [ -2.16857717,  0.52714953],
       [ -2.13956345,  0.31321781],
       [ -2.26526149, -0.3377319 ],
       [ -2.14012214, -0.50454069],
       [ -1.83159477,  0.42369507],
       [ -2.61494794,  1.79357586],
       [ -2.44617739,  2.15072788],
       [ -2.10997488, -0.46020184],
       [ -2.2078089 , -0.2061074 ],
       [ -2.04514621,  0.66155811],
       [ -2.52733191,  0.59229277],
       [ -2.42963258, -0.90418004],
       [ -2.16971071,  0.26887896],
       [ -2.28647514,  0.44171539],
       [ -1.85812246, -2.33741516],
```

[-2.5536384 , -0.47910069],  
[-1.96444768, 0.47232667],  
[-2.13705901, 1.14222926],  
[-2.0697443 , -0.71105273],  
[-2.38473317, 1.1204297 ],  
[-2.39437631, -0.38624687],  
[-2.22944655, 0.99795976],  
[-2.20383344, 0.00921636],  
[ 1.10178118, 0.86297242],  
[ 0.73133743, 0.59461473],  
[ 1.24097932, 0.61629765],  
[ 0.40748306, -1.75440399],  
[ 1.0754747 , -0.20842105],  
[ 0.38868734, -0.59328364],  
[ 0.74652974, 0.77301931],  
[-0.48732274, -1.85242909],  
[ 0.92790164, 0.03222608],  
[ 0.01142619, -1.03401828],  
[-0.11019628, -2.65407282],  
[ 0.44069345, -0.06329519],  
[ 0.56210831, -1.76472438],  
[ 0.71956189, -0.18622461],  
[-0.0333547 , -0.43900321],  
[ 0.87540719, 0.50906396],  
[ 0.35025167, -0.19631173],  
[ 0.15881005, -0.79209574],  
[ 1.22509363, -1.6222438 ],  
[ 0.1649179 , -1.30260923],  
[ 0.73768265, 0.39657156],  
[ 0.47628719, -0.41732028],  
[ 1.2341781 , -0.93332573],  
[ 0.6328582 , -0.41638772],  
[ 0.70266118, -0.06341182],  
[ 0.87427365, 0.25079339],  
[ 1.25650912, -0.07725602],  
[ 1.35840512, 0.33131168],  
[ 0.66480037, -0.22592785],  
[-0.04025861, -1.05871855],  
[ 0.13079518, -1.56227183],  
[ 0.02345269, -1.57247559],  
[ 0.24153827, -0.77725638],  
[ 1.06109461, -0.63384324],  
[ 0.22397877, -0.28777351],  
[ 0.42913912, 0.84558224],  
[ 1.04872805, 0.5220518 ],  
[ 1.04453138, -1.38298872],  
[ 0.06958832, -0.21950333],  
[ 0.28347724, -1.32932464],  
[ 0.27907778, -1.12002852],  
[ 0.62456979, 0.02492303],  
[ 0.33653037, -0.98840402],  
[-0.36218338, -2.01923787],  
[ 0.28858624, -0.85573032],  
[ 0.09136066, -0.18119213],  
[ 0.22771687, -0.38492008],  
[ 0.57638829, -0.1548736 ],  
[-0.44766702, -1.54379203],  
[ 0.25673059, -0.5988518 ],  
[ 1.84456887, 0.87042131],  
[ 1.15788161, -0.69886986],  
[ 2.20526679, 0.56201048],  
[ 1.44015066, -0.04698759],  
[ 1.86781222, 0.29504482],  
[ 2.75187334, 0.8004092 ],  
[ 0.36701769, -1.56150289],  
[ 2.30243944, 0.42006558],  
[ 2.00668647, -0.71143865],  
[ 2.25977735, 1.92101038],  
[ 1.36417549, 0.69275645],

```
[ 1.60267867, -0.42170045],
[ 1.8839007 ,  0.41924965],
[ 1.2601151 , -1.16226042],
[ 1.4676452 , -0.44227159],
[ 1.59007732,  0.67624481],
[ 1.47143146,  0.25562182],
[ 2.42632899,  2.55666125],
[ 3.31069558,  0.01778095],
[ 1.26376667, -1.70674538],
[ 2.0377163 ,  0.91046741],
[ 0.97798073, -0.57176432],
[ 2.89765149,  0.41364106],
[ 1.33323218, -0.48181122],
[ 1.7007339 ,  1.01392187],
[ 1.95432671,  1.0077776 ],
[ 1.17510363, -0.31639447],
[ 1.02095055,  0.06434603],
[ 1.78834992, -0.18736121],
[ 1.86364755,  0.56229073],
[ 2.43595373,  0.25928443],
[ 2.30492772,  2.62632347],
[ 1.86270322, -0.17854949],
[ 1.11414774, -0.29292262],
[ 1.2024733 , -0.81131527],
[ 2.79877045,  0.85680333],
[ 1.57625591,  1.06858111],
[ 1.3462921 ,  0.42243061],
[ 0.92482492,  0.0172231 ],
[ 1.85204505,  0.67612817],
[ 2.01481043,  0.61388564],
[ 1.90178409,  0.68957549],
[ 1.15788161, -0.69886986],
[ 2.04055823,  0.8675206 ],
[ 1.9981471 ,  1.04916875],
[ 1.87050329,  0.38696608],
[ 1.56458048, -0.89668681],
[ 1.5211705 ,  0.26906914],
[ 1.37278779,  1.01125442],
[ 0.96065603, -0.02433167]]])
```

In [21]:

```
from scipy.cluster.hierarchy import dendrogram, linkage

linked = linkage(result, 'single')
```

In [22]:

```
iris.target
```

Out[22]:

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

In [40]:

```
in [40]:
```

```
labelList = iris.target

plt.figure(figsize=(30, 7))
dendrogram(linked,
            orientation='top',
            labels=labelList,
            show_leaf_counts=False,
            leaf_font_size=20,
            leaf_rotation= 360
        )
plt.show()
```

