지도(supervised) 학습(회귀, 분류)

생선 분류 문제

도미 데이터 준비하기

https://scikit-learn.org/stable/index.html

In [1]:
```
bream_length = [25.4, 26.3, 26.5, 29.0, 29.0, 29.7, 29.7, 30.0, 30.0, 30.7, 31.0, 31
bream_weight = [242.0, 290.0, 340.0, 363.0, 430.0, 450.0, 500.0, 390.0, 450.0, 500.0
```

In [5]:
```
print(len(bream_length))
print(len(bream_weight))
```

```
35
35
```

In [6]:
```
# 상관을 알기 위해서는 공분산을 알아야 함
```

In [7]:
```
# 공분산의 개념을 알아야 함
```

공분산

확률변수 X에 대한 흩어짐의 정도를 분산으로 표시가능.
또 다른 확률변수 Y에 대한 흩어짐의 정도도 분산으로 표시할 수 있다.
→ 이 각각의 분산에 대한 공통점이 공분산
이에 대한 분석 : 공분산 분석

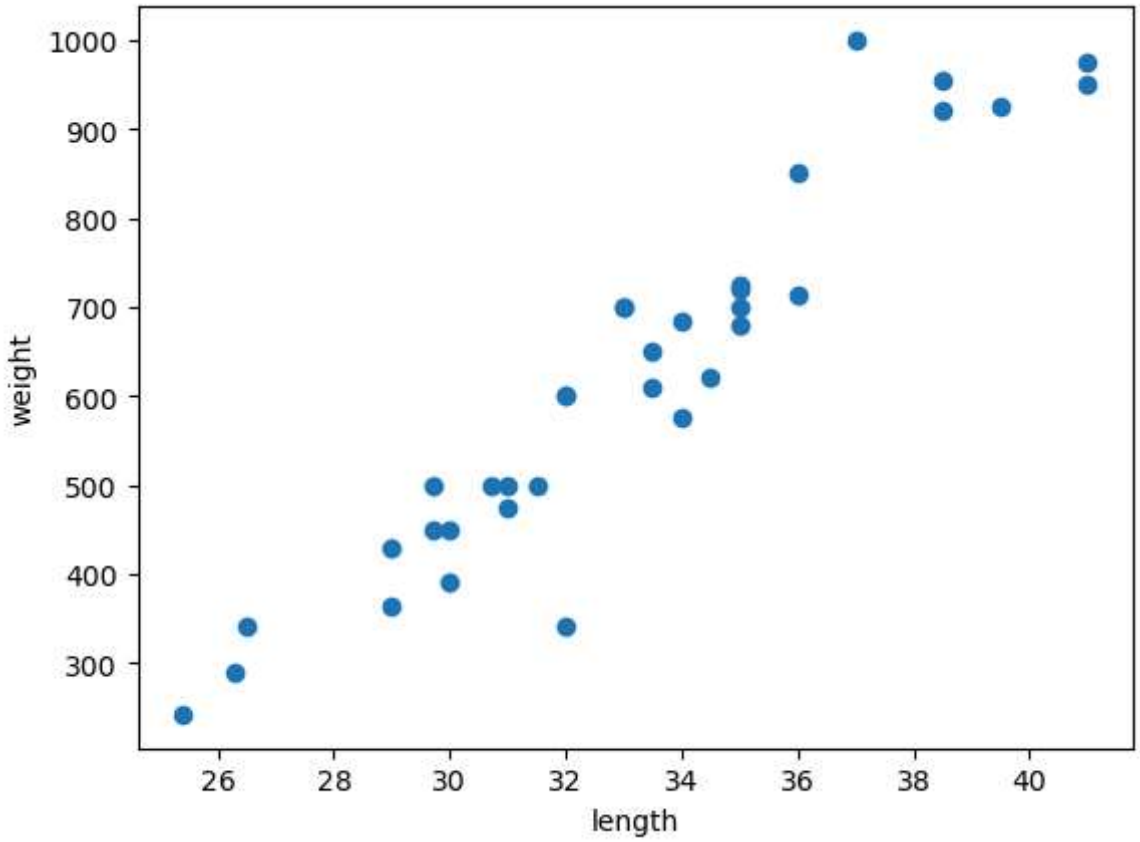
공분산은 Cov (X,Y)로 표현하며,

$$Cov(X, Y) = \frac{\sum_{i=1}^{N}(X_i - \overline{X})(Y_i - \overline{Y})}{N}$$

$$(\text{공분산}) = \frac{[((\text{개별 } X \text{ 측정치}) - (X\text{의 평균})) \times ((\text{개별 } Y \text{ 측정치}) - (Y\text{의 평균}))]\text{의 총합}}{(\text{조합을 이루는 개수})}$$

$$= \frac{[(X\text{의 평균편차}) \times (Y\text{의 평균편차})]\text{의 총합}}{(\text{조합을 이루는 개수})}$$

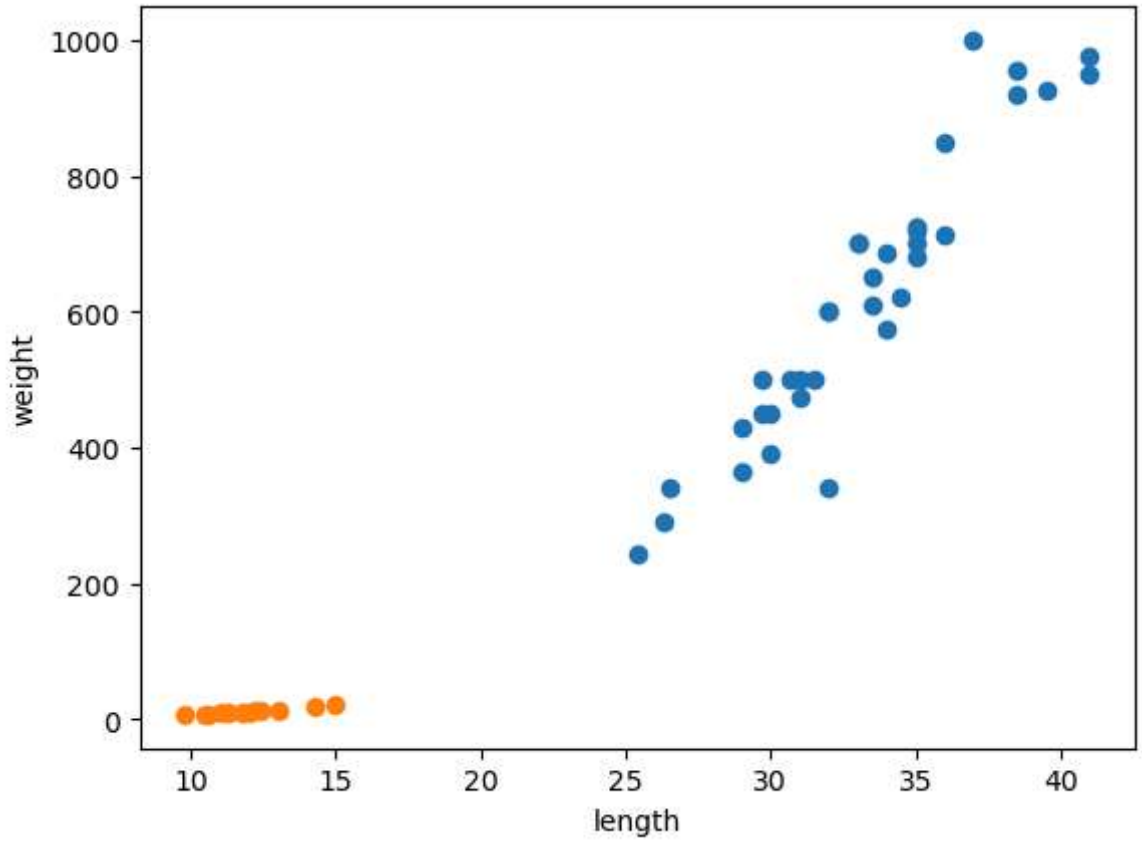
In [2]:
```
import matplotlib.pyplot as plt

plt.scatter(bream_length, bream_weight)
plt.xlabel('length')
plt.ylabel('weight')
plt.show()
```



빙어 데이터 준비하기

In [3]:
```
smelt_length = [9.8, 10.5, 10.6, 11.0, 11.2, 11.3, 11.8, 11.8, 12.0, 12.2, 12.4, 13.
smelt_weight = [6.7, 7.5, 7.0, 9.7, 9.8, 8.7, 10.0, 9.9, 9.8, 12.2, 13.4, 12.2, 19.7
```

In [4]:
```
plt.scatter(bream_length, bream_weight)
plt.scatter(smelt_length, smelt_weight)
plt.xlabel('length')
plt.ylabel('weight')
plt.show()
```



In [8]:
```
import numpy as np
```

In [9]:
```
bream_length_mean=np.mean(bream_length)
bream_weight_mean=np.mean(bream_weight)
```

In [10]:
```
print(bream_length_mean,bream_weight_mean)
```

```
33.10857142857143 617.8285714285714
```

In [12]:
```
print('도미의 길이 : %.2fcm, 도미의 무게 : %.2fg' %(bream_length_mean,bream_weight_m
```

```
도미의 길이 : 33.11cm, 도미의 무게 : 617.83g
```

In [13]:
```
np.sum((bream_length-bream_length_mean)*(bream_weight-bream_weight_mean))/len(bream_
```

Out[13]:
```
752.2871836734696
```

In [14]:
```
np.sum((bream_length-bream_length_mean)*(bream_weight-bream_weight_mean))/len(bream_
```

Out[14]:
```
752.2871836734696
```

In [16]:
```
#help(np.cov)
```

In [17]:
```
np.cov(bream_length,bream_weight)
```

Out[17]:
```
array([[1.53031597e+01, 7.74413277e+02],
       [7.74413277e+02, 4.37670286e+04]])
```

In [18]:
```
7.74413277e+02
```

Out[18]:
```
774.413277
```

In [20]:
```
np.sum((bream_length-bream_length_mean)*(bream_weight-bream_weight_mean))/(len(bream
```

Out[20]: 774.4132773109245

상관계수

공분산이 $Cov(x,y) = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{n}$ 일 때,

상관계수는 $Corr(x,y) = \frac{Cov(x,y)}{\sqrt{\frac{\sum(x_i-\bar{x})^2}{n} \times \frac{\sum(y_i-\bar{y})^2}{n}}}$

이를 정리하면

$$Corr(x,y) = \frac{Cov(x,y)}{S.D(x) \times S.D(y)} \ (S.D(x) : x\text{의 표준편차},\ S.D(y) : y\text{의 표준편차})$$

$$(\text{상관계수}) = \frac{(\text{공분산})}{(X\text{의 표준편차}) \times (Y\text{의 표준편차})}$$

In [24]:
```
np.sqrt(np.std(bream_length)*np.std(bream_weight))
```

Out[24]: 28.195998535286826

In [23]:
```
774.4132773109245/np.sqrt(np.std(bream_length)*np.std(bream_weight))
```

Out[23]: 27.465360956866242

In [27]:
```
774.4132773109245/(np.std(bream_length)*np.std(bream_weight))
```

Out[27]: 0.9740871890915229

In [34]:
```
774.4132773109245/np.sqrt(np.sum((bream_length-bream_length_mean)**2)/(len(bream_le
```

Out[34]: 0.9462561265460507

In [25]:
```
#help(np.corrcoef)
```

In [26]:
```
np.corrcoef(bream_length,bream_weight)
```

Out[26]:
```
array([[1.        , 0.94625613],
       [0.94625613, 1.        ]])
```

In [37]:
```
#!pip install nbconvert[webpdf]
```

In [ ]:
```

```