ch04 비선형 변환

학습 내용

- 01 비선형 변환
- 02 실습을 통해 확인해 보기
- 03 데이터를 리지 회귀(L1규제)에 적용

01 비선형 변환

- 제곱항이나 세제곱 항을 추가하면 선형 회귀 모델에 도움이 된다.
- log, exp, sin 같은 수학 함수를 적용하는 방법도 특성 변환에 유용.
- 선형 모델과 신경망은 각 특성의 스케일과 분포에 밀접하게 연관되어 있음.
- log, exp 함수는 데이터의 스케일을 변경하여 선형 모델과 신경망을 올리는데 도움이 된다.

In [12]:

```
import os, warnings
# 경고 메시지 무시하거나 숨길때(ignore), 다시보이게(default)
# warnings.filterwarnings(action='default')
warnings.filterwarnings(action='ignore')
```

In [13]:

```
# 한글
import matplotlib
from matplotlib import font_manager, rc
font_loc = "C:/Windows/Fonts/malgunbd.ttf"
font_name = font_manager.FontProperties(fname=font_loc).get_name()
matplotlib.rc('font', family=font_name)

matplotlib.rcParams['axes.unicode_minus'] = False
```

02 실습을 통해 확인해 보기

In [14]:

```
import numpy as np
import matplotlib.pyplot as plt
```

In [15]:

```
rnd = np.random.RandomState(0)
X_org = rnd.normal(size=(1000,3))
print(X_org.shape)

w = rnd.normal(size=3)
print(w.shape)
```

```
(1000, 3)
(3,)
```

In [16]:

```
X = rnd.poisson(10 * np.exp(X_org))
y = np.dot(X_org, w)
print(X[:10, 0])
print(X[:10, 1])
```

```
[ 56  81  25  20  27  18  12  21 109   7]
[18 57  9 13 13 46  3 20  1 55]
```

In [17]:

```
### 각 값이 가지는 것에 대해 확인해 보기
print("특성 출현 횟수 :₩n",np.bincount(X[:,0]))
```

```
특성 출현 횟수 :
 [28 38 68 48 61 59 45 56 37 40 35 34 36 26 23 26 27 21 23 23 18 21 10  9
 17  9  7 14 12  7  3  8  4  5  5  3  4  2  4  1  1  3  2  5  3  8  2  5
  2  1  2  3  3  2  2  3  3  0  1  2  1  0  0  3  1  0  0  0  1  3  0  1
  0  2  0  1  1  0  0  0  0  1  0  0  2  2  0  1  1  0  0  0  0  1  1  0
  0  0  0  0  0  0  1  0  0  0  0  0  1  1  0  0  1  0  0  0  0  0  0  0
  1  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1]
```

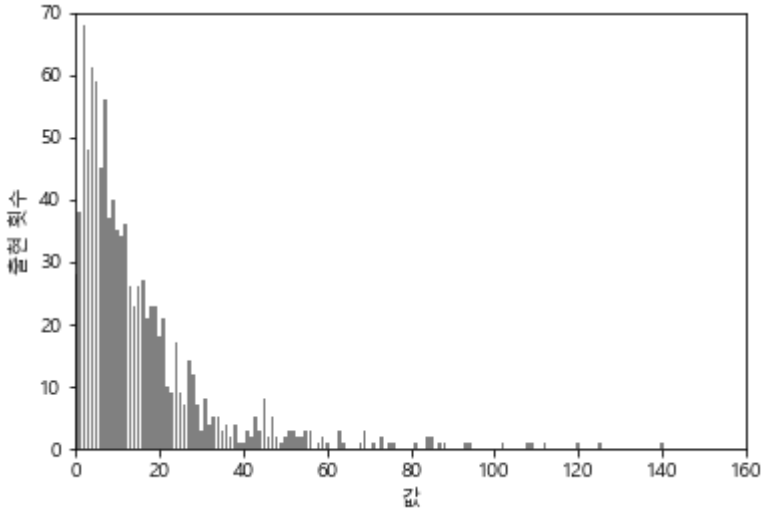
- 2가 68번으로 가장 많이 나타나며. 큰 값의 수는 빠르게 줄어든다.
- 85, 86처럼 아주 큰 값도 약간은 있음

In [18]:

```
plt.xlim(0, 160)
plt.ylim(0, 70)
bins = np.bincount(X[:, 0])
plt.bar(range(len(bins)), bins, color='grey')
plt.ylabel("출현 횟수")
plt.xlabel("값")
```

Out[18]:

```
Text(0.5, 0, '값')
```



In [22]:

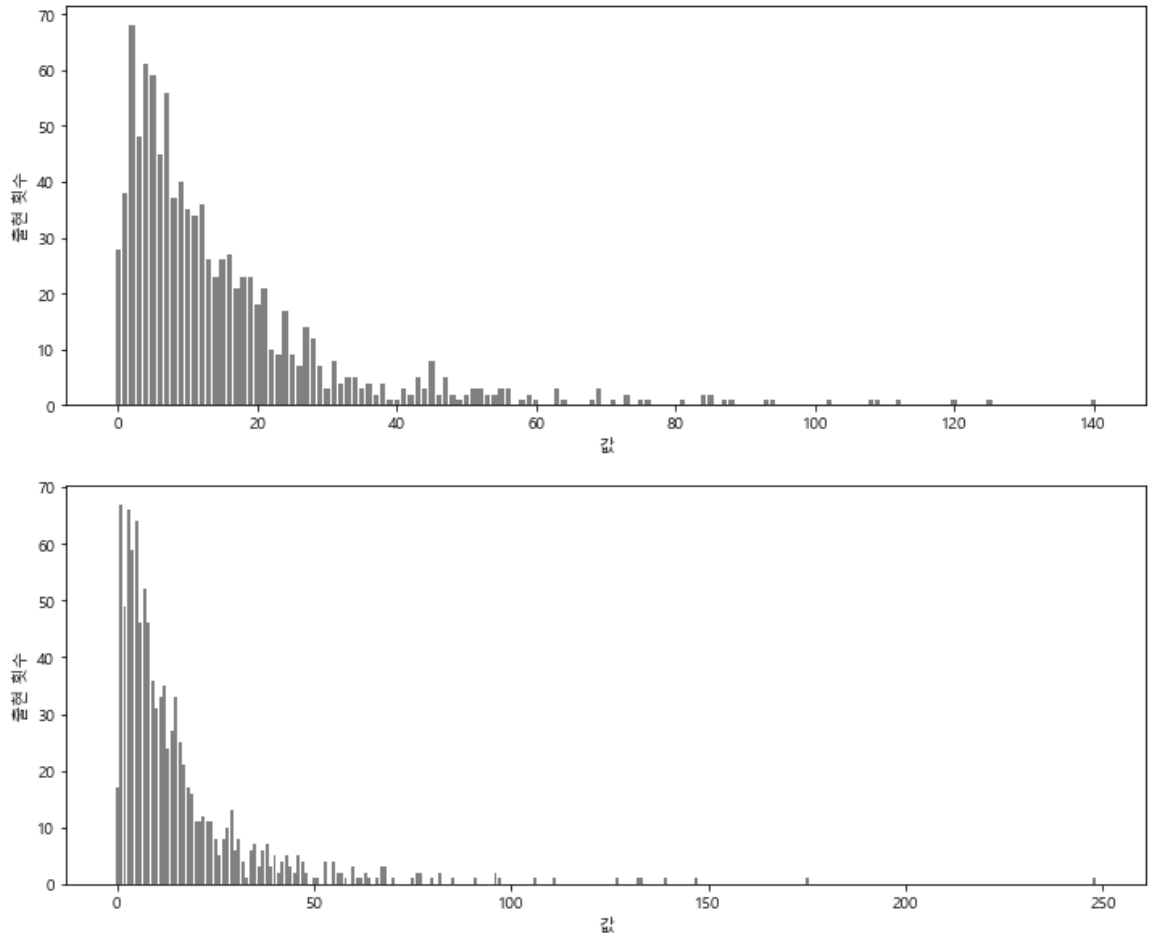
```
plt.figure(figsize=(12, 10))

plt.subplot(2,1,1)
bins = np.bincount(X[:, 0])
plt.bar(range(len(bins)), bins, color='grey')
plt.ylabel("출현 횟수")
plt.xlabel("값")

plt.subplot(2,1,2)
bins = np.bincount(X[:, 1])
plt.bar(range(len(bins)), bins, color='grey')
plt.ylabel("출현 횟수")
plt.xlabel("값")
```

Out[22]:

Text(0.5, 0, '값')



- X[:, 1]과 X[:, 2]의 특성도 비슷하다.

03 데이터를 리지 회귀(L1규제)에 적용

In [24]:

```
from sklearn.linear_model import Ridge
from sklearn.model_selection import train_test_split
```

In [26]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0)
score = Ridge().fit(X_train, y_train).score(X_test, y_test)
print("테스트 점수 : {:.3f}".format(score))
```

테스트 점수 : 0.622

In [27]:

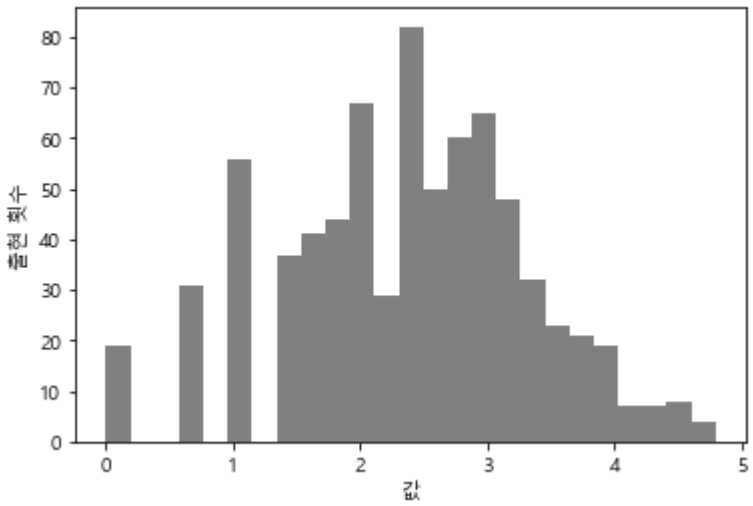
```
X_train_log = np.log(X_train + 1)
X_test_log = np.log(X_test + 1)
```

In [28]:

```
plt.hist( X_train_log[:, 0], bins=25, color='gray')
plt.ylabel("출현 횟수")
plt.xlabel("값")
```

Out[28]:

Text(0.5, 0, '값')



In [29]:

```
score = Ridge().fit(X_train_log, y_train).score(X_test_log, y_test)
print("테스트 점수 : {:.3f}".format(score))
```

테스트 점수 : 0.875

- 트리 기반 모델에서는 이러한 변환은 불필요하지만 선형 모델에서는 필수이다.
- 선형모델, 나이브 베이즈 모델 같은 덜 복잡한 모델에서 구간 분할, 다항식, 상호작용은 데이터가 주어진 상황에서 모델의 성능에 큰 영향을 줄 수 있다.

In [ ]:

```

```