

## 8.2 베르누이분포와 이항분포

베르누이분포와 이항분포는 확률분포 중 가장 단순한 분포지만 분류문제에서 널리 사용된다. 이 절에서는 두 분포의 개념을 알아보고 이 분포들이 스팸메일 필터링에 어떻게 쓰이는지 살펴본다.

### 베르누이 시행

결과가 두 가지 중 하나로만 나오는 실험이나 시행을 **베르누이 시행(Bernoulli trial)**이라고 한다. 예를 들어 동전을 한 번 던져 앞면(H:Head)이 나오거나 뒷면(T:Tail)이 나오게 하는 것도 베르누이 시행이다.

### 베르누이 확률변수

베르누이 시행의 결과를 실수 0 또는 1로 바꾼 것을 **베르누이 확률변수(Bernoulli random variable)**라고 한다. 베르누이 확률변수는 두 값 중 하나만 가질 수 있으므로 이산확률변수(discrete random variable)다. 베르누이 확률변수의 표본값은 보통 정수 1과 0으로 표현하지만 때로는 정수 1과 -1로 표현하는 경우도 있다.

### 베르누이 확률분포

베르누이 확률변수의 분포를 베르누이 확률분포 혹은 베르누이분포라고 한다. 만약 어떤 확률변수  $X$ 가 베르누이분포에 의해 발생된다면 '**확률변수  $X$ 가 베르누이분포를 따른다**'라고 말하고 다음과 같이 수식으로 쓴다.

$$X \sim \text{Bern}(x; \mu) \quad (8.2.1)$$

베르누이분포의 확률질량함수 수식은 다음과 같다.

$$\text{Bern}(x; \mu) = \begin{cases} \mu & \text{if } x = 1, \\ 1 - \mu & \text{if } x = 0 \end{cases} \quad (8.2.2)$$

베르누이분포는 **1이 나올 확률**을 의미하는  $\mu$ 라는 **모수(parameter)**를 가진다. 변수와 모수는 세미콜론(,; semicolon)기호로 분리했다. 0이 나올 확률은  $1 - \mu$ 이다.

위 식을 하나의 수식으로 표현하면 다음처럼 쓸 수 있다.

$$\text{Bern}(x; \mu) = \mu^x (1 - \mu)^{(1-x)} \quad (8.2.3)$$

#### 연습 문제 8.2.1

위 식에서  $x = 1$ 과  $x = 0$ 을 각각 대입하여 원래의 확률질량함수 수식이 나오는 것을 확인한다.

만약 베르누이 확률변수의 표본값이 1과 0이 아니라 1과 -1이라는 값을 가진다면 다음과 같은 수식으로 써야 한다.

$$\text{Bern}(x; \mu) = \mu^{(1+x)/2} (1 - \mu)^{(1-x)/2} \quad (8.2.4)$$

### 사이파이를 사용한 베르누이 확률변수의 시뮬레이션

사이파이의 stats 서브 패키지에 있는 `bernoulli` 클래스는 베르누이 확률변수를 구현했다.  $p$  인수로 분포의 모수  $\mu$ 를 설정한다.

다음 예에서는  $p=0.6$  으로 설정했다. 이러한 확률변수를 만든 것은 앞면이 나올 확률이 0.6인 동전을 만든 것과 같다.

In [1]:

```
mu = 0.6
rv = sp.stats.bernoulli(mu)
type(rv)
```

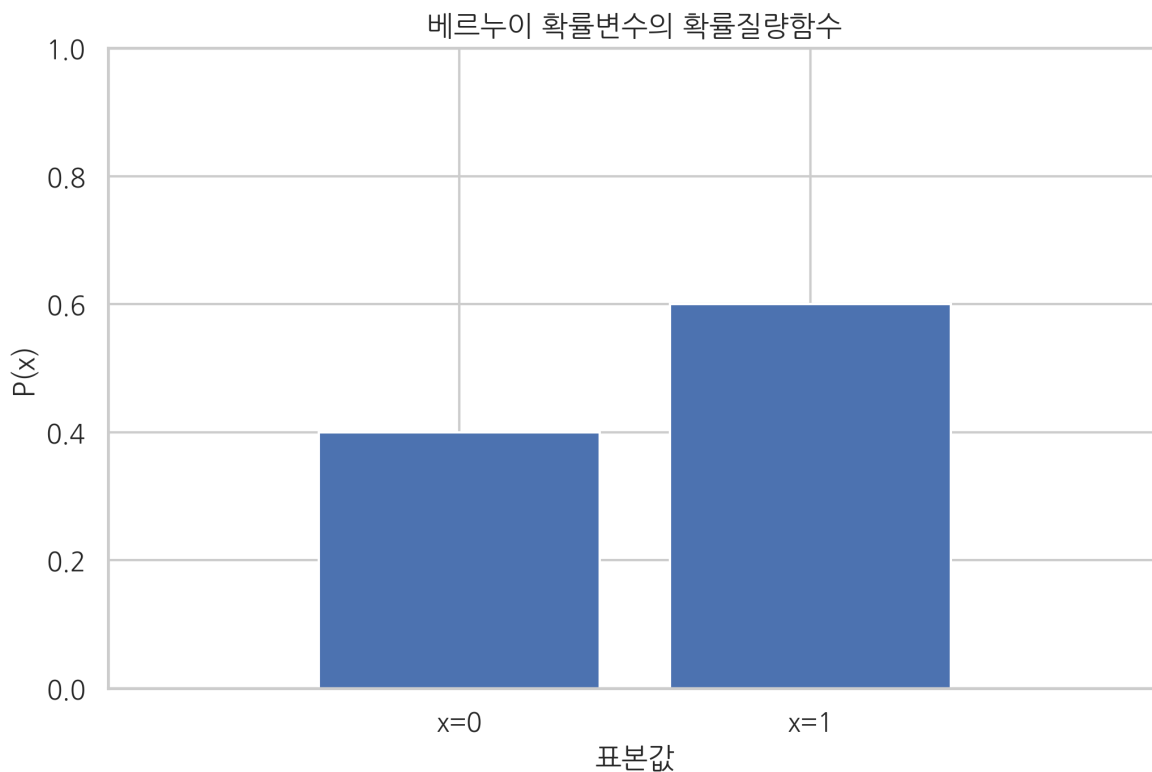
Out[1]:

```
scipy.stats._distn_infrastructure.rv_frozen
```

확률질량함수는 `pmf()` 메서드로 계산한다.

In [2]:

```
xx = [0, 1]
plt.bar(xx, rv.pmf(xx))
plt.xlim(-1, 2)
plt.ylim(0, 1)
plt.xticks([0, 1], ["x=0", "x=1"])
plt.xlabel("표본값")
plt.ylabel("P(x)")
plt.title("베르누이 확률변수의 확률질량함수")
plt.show()
```



표본값을 무작위로 생성하려면 `rvs()` 메서드를 사용한다. 다음 코드는 표본 100개를 생성한다. 이는 위에서 만든 동전을 100번 던져 나온 결과를 나타낸 것과 같다. 그래서 확률변수의 표본을 생성하는 작업을 **시뮬레이션 (simulation)**이라고도 부른다.

In [3]:

```
x = rv.rvs(100, random_state=0)
x
```

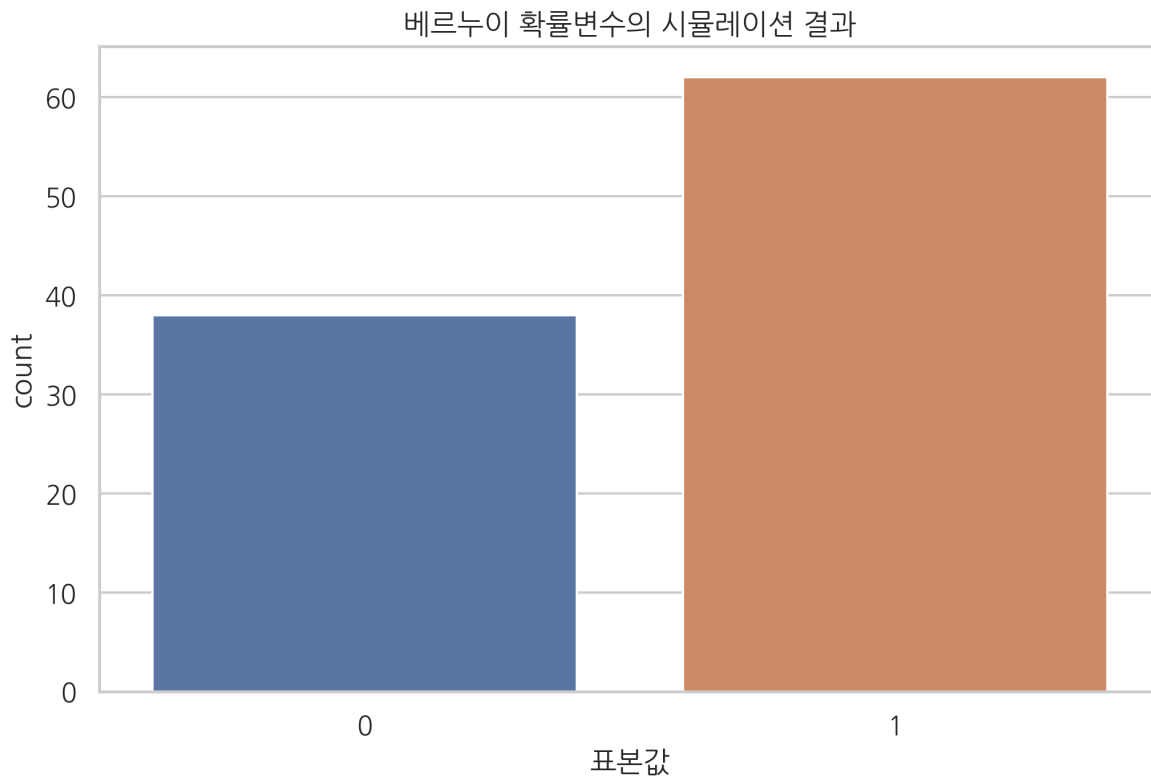
Out[3]:

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

다음은 생성 결과를 시본의 `countplot()` 명령으로 시각화한 것이다.

In [4]:

```
sns.countplot(x)
plt.title("베르누이 확률변수의 시뮬레이션 결과")
plt.xlabel("표본값")
plt.show()
```



이론적인 확률분포와 표본의 확률분포를 동시에 보이려면 다음과 같은 코드를 사용한다.

In [5]:

```
y = np.bincount(x, minlength=2) / float(len(x))

df = pd.DataFrame({"이론": rv.pmf(xx), "시뮬레이션": y})
df.index = [0, 1]
df
```

Out[5]:

	이론	시뮬레이션
0	0.4	0.38
1	0.6	0.62

시본의 `barplot()` 명령으로 시각화하면 다음과 같다.

In [6]:

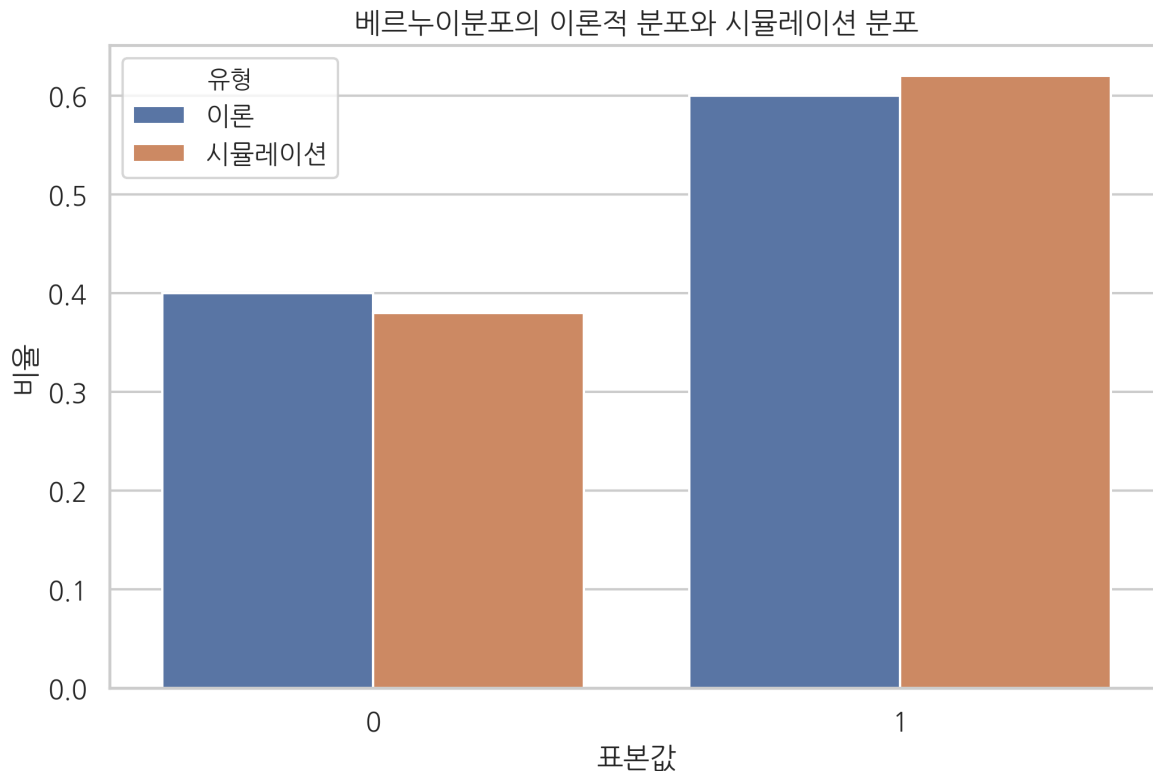
```
df2 = df.stack().reset_index()
df2.columns = ["표본값", "유형", "비율"]
df2
```

Out[6]:

	표본값	유형	비율
0	0	이론	0.40
1	0	시뮬레이션	0.38
2	1	이론	0.60
3	1	시뮬레이션	0.62

In [7]:

```
sns.barplot(x="표본값", y="비율", hue="유형", data=df2)
plt.title("베르누이분포의 이론적 분포와 시뮬레이션 분포")
plt.show()
```



### 연습 문제 8.2.2

베르누이 확률분포의 모수가 다음과 같을 경우에 각각 표본을 생성한 후 기댓값과 분산을 구하고 앞의 예제와 같이 확률 밀도 함수와 비교한 바 플롯을 그린다. 표본 개수가 10개인 경우와 1000개인 경우에 대해 각각 위의 계산을 한다.

(1)  $\mu = 0.5$

(2)  $\mu = 0.9$

### 베르누이분포의 모멘트

베르누이분포의 모멘트는 다음과 같다.

- 기댓값

$$E[X] = \mu \quad (8.2.5)$$

(증명)

$$\begin{aligned} E[X] &= \sum_{x_i \in \Omega} x_i p(x_i) \\ &= 1 \cdot \mu + 0 \cdot (1 - \mu) \\ &= \mu \end{aligned} \quad (8.2.6)$$

- 분산

$$\text{Var}[X] = \mu(1 - \mu) \quad (8.2.7)$$

(증명)

$$\begin{aligned} \text{Var}[X] &= \sum_{x_i \in \Omega} (x_i - \mu)^2 p(x_i) \\ &= (1 - \mu)^2 \cdot \mu + (0 - \mu)^2 \cdot (1 - \mu) \\ &= \mu(1 - \mu) \end{aligned} \quad (8.2.8)$$

앞의 예에서는  $\mu = 0.6$ 이었으므로 이론적인 기댓값과 분산은 다음과 같다.

$$E[X] = 0.6 \quad (8.2.9)$$

$$\text{Var}[X] = 0.6 \cdot (1 - 0.6) = 0.24 \quad (8.2.10)$$

데이터에서 계산한 표본평균 및 표본분산은 다음과 같이 계산한다.

In [8]:

```
np.mean(x)
```

Out [8]:

0.62

In [9]:

```
np.var(x, ddof=1)
```

Out [9]:

0.23797979797979804

기술통계값들은 사이파이가 제공하는 `describe` 명령으로 계산할 수도 있다.

In [10]:

```
s = sp.stats.describe(x)
s[2], s[3]
```

Out [10]:

(0.62, 0.23797979797979804)

## 이항분포

성공확률이  $\mu$  인 베르누이 시행을  $N$ 번 반복하는 경우를 생각해보자. 가장 운이 좋을 때에는  $N$ 번 모두 성공할 것이고 가장 운이 나쁜 경우에는 한 번도 성공하지 못할 것이다.  $N$ 번 중 성공한 횟수를 확률변수  $X$ 라고 한다면  $X$ 의 값은 0 부터  $N$  까지의 정수 중 하나가 될 것이다.

이런 확률변수를 **이항분포(binomial distribution)**를 따르는 확률변수라고 하며 다음과 같이 표시한다.

$$X \sim \text{Bin}(x; N, \mu) \quad (8.2.11)$$

베르누이분포와 이항분포는 모두 베르누이 확률변수에서 나온 표본값이다. 표본 데이터가 하나 뿐이면 베르누이분포가 되고 표본 데이터가 여럿이면 이항분포가 된다.

이항분포 확률변수  $X$ 의 확률질량함수를 구해보자. 우선 베르누이 확률 분포를 따르는 확률변수  $Y$ 를 가정한다.

$$Y \sim \text{Bern}(y; \mu) \quad (8.2.12)$$

이 확률변수의  $N$ 개의 표본을  $y_1, y_2, \dots, y_N$ 라고 하자. 이 값은 모두 0(실패) 아니면 1(성공)이라는 값을 가지기 때문에  $N$ 번 중 성공한 횟수는  $N$ 개의 표본값의 합이다.

$$x = \sum_{i=1}^N y_i \quad (8.2.13)$$

베르누이분포를 따르는 확률변수  $Y$ 의 확률질량함수를 대입하여 정리하면 이항분포 확률변수  $X$ 의 확률질량함수는 다음과 같아진다.

$$\text{Bin}(x; N, \mu) = \binom{N}{x} \mu^x (1 - \mu)^{N-x} \quad (8.2.14)$$

이 식에서  $\binom{N}{x}$  기호는 **조합(combination)**이라는 기호로  $N$ 개 원소 중에  $x$ 개 원소를 순서와 상관없이 선택할 수 있는 경우의 수를 뜻한다. 조합은 다음 공식으로 계산할 수 있다.

$$\binom{N}{x} = \frac{N!}{x!(N-x)!} \quad (8.2.15)$$

! 기호는 **팩토리얼(factorial)**이라고 하며 다음처럼 정의한다.

$$N! = N \cdot (N-1) \cdots 2 \cdot 1 \quad (8.2.16)$$

## 사이파이를 사용한 이항분포의 시뮬레이션

이항분포 확률변수는 사이파이의 stats 서브패키지에 `binom` 클래스로 구현되어 있다. `n` 인수로 전체 시행 횟수  $N$ 를 넣고 `p` 인수로 베르누이 확률 분포의 기댓값  $\mu$ 를 넣는다.

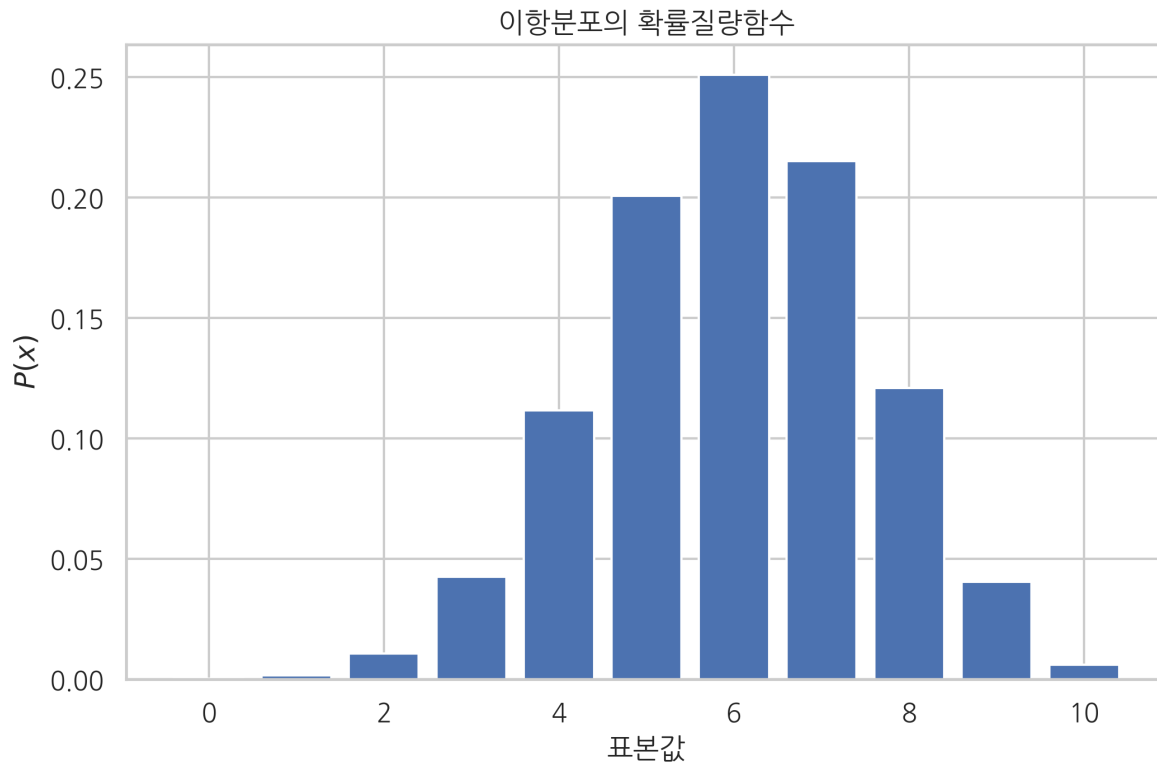
In [11]:

```
N = 10
mu = 0.6
rv = sp.stats.binom(N, mu)
```

이 때 확률질량함수는 다음과 같다. 확률질량함수의 모양이 하나의 봉우리를 가지는 단봉분포지만 0부터  $N$ 사이의 정수값만 가질 수 있고 모수의 값에 따라 봉우리의 위치가 달라진다.

In [12]:

```
xx = np.arange(N + 1)
plt.bar(xx, rv.pmf(xx), align="center")
plt.xlabel("표본값")
plt.ylabel("$P(x)$")
plt.title("이항분포의 확률질량함수")
plt.show()
```



`rvs()` 메서드로 무작위 표본을 뽑아내는 시뮬레이션을 한 결과는 다음과 같다. 시뮬레이션 결과는 확률질량함수의 모양과 비슷하지만 정확히 일치하지는 않는다.



In [13]:

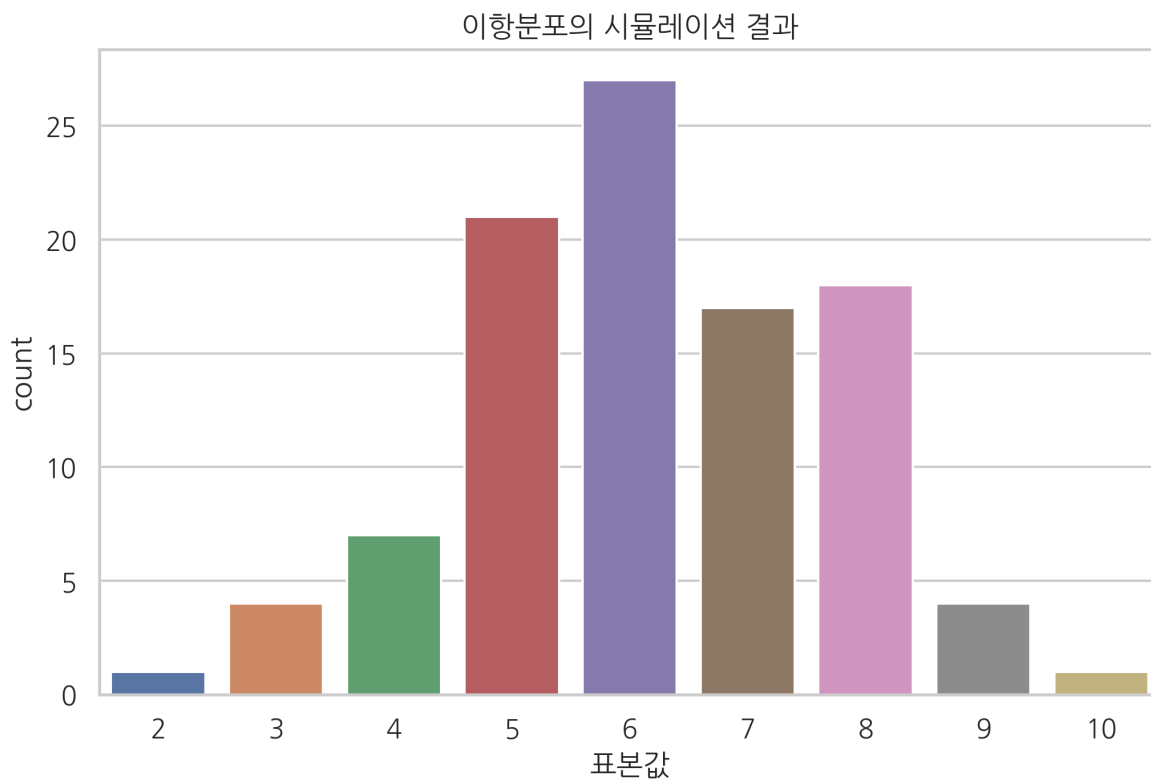
```
np.random.seed(0)
x = rv.rvs(100)
x
```

Out [13]:

```
array([ 6,  5,  6,  6,  6,  5,  6,  4,  3,  6,  5,  6,  6,  4,  8,  8,  9,
        5,  5,  4,  3,  5,  6,  5,  8,  5,  8,  4,  6,  6,  7,  5,  6,  6,
        9,  6,  6,  6,  4,  5,  7,  6,  5,  8,  5,  5,  7,  8,  7,  7,  6,
        6,  2,  8,  7,  8,  5,  7,  6,  7,  8,  8,  5,  8,  7,  7,  5,  8,
        4,  8,  3,  6,  3,  6,  5,  9,  7,  8,  7,  8,  7,  6,  8,  5,  6,
        7,  6,  8,  6,  4,  7,  5,  8,  5,  7,  7,  6,  9,  5, 10])
```

In [14]:

```
sns.countplot(x)
plt.title("이항분포의 시뮬레이션 결과")
plt.xlabel("표본값")
plt.show()
```



이론적인 확률분포와 표본의 확률분포를 같이 나타내려면 다음과 같은 코드를 사용한다.

In [15]:

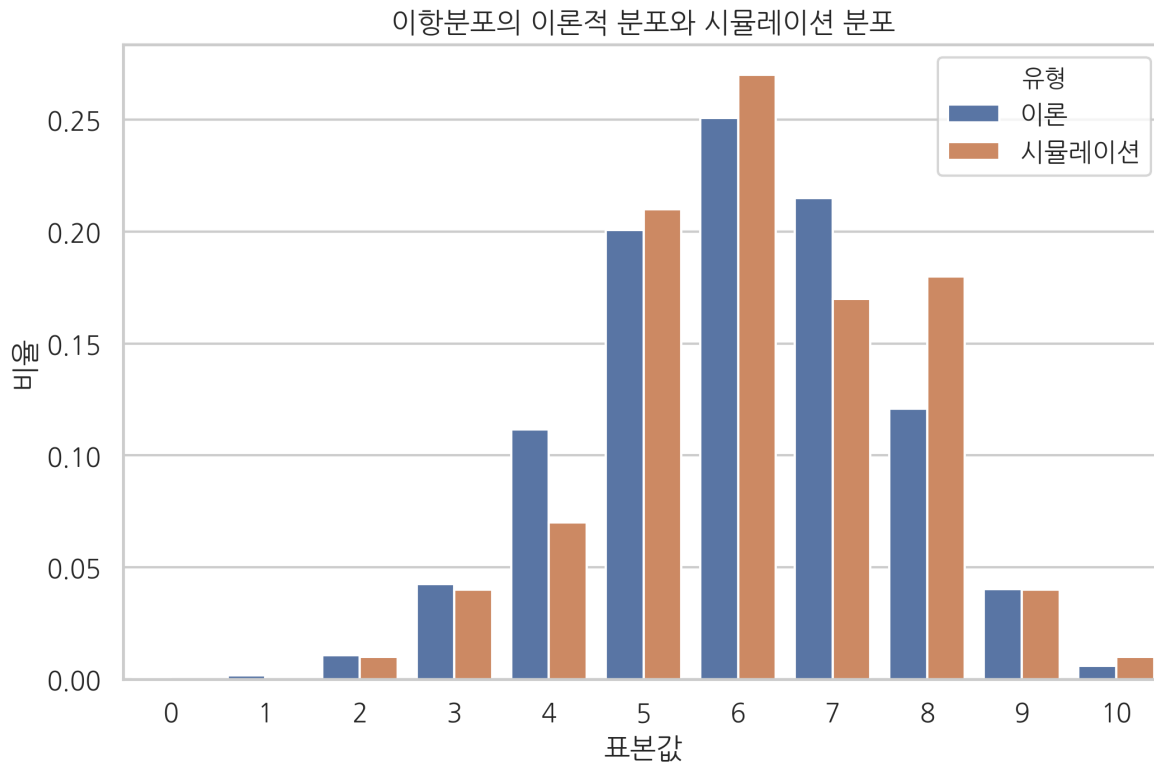
```
y = np.bincount(x, minlength=N+1)/float(len(x))
df = pd.DataFrame({"이론": rv.pmf(xx), "시뮬레이션": y}).stack()
df = df.reset_index()
df.columns = ["표본값", "유형", "비율"]
df.pivot("표본값", "유형", "비율")
df
```

Out [15]:

	표본값	유형	비율
0	0	이론	0.000105
1	0	시뮬레이션	0.000000
2	1	이론	0.001573
3	1	시뮬레이션	0.000000
4	2	이론	0.010617
5	2	시뮬레이션	0.010000
6	3	이론	0.042467
7	3	시뮬레이션	0.040000
8	4	이론	0.111477
9	4	시뮬레이션	0.070000
10	5	이론	0.200658
11	5	시뮬레이션	0.210000
12	6	이론	0.250823
13	6	시뮬레이션	0.270000
14	7	이론	0.214991
15	7	시뮬레이션	0.170000
16	8	이론	0.120932
17	8	시뮬레이션	0.180000
18	9	이론	0.040311
19	9	시뮬레이션	0.040000
20	10	이론	0.006047
21	10	시뮬레이션	0.010000

In [16]:

```
sns.barplot(x="표본값", y="비율", hue="유형", data=df)
plt.title("이항분포의 이론적 분포와 시뮬레이션 분포")
plt.show()
```



## 이항분포의 모멘트

이항분포의 기댓값과 분산은 각각 다음과 같다.

- 기댓값

$$E[X] = N\mu \quad (8.2.17)$$

(증명)

$$E[X] = E\left[\sum_{i=1}^N Y_i\right] = \sum_{i=1}^N E[Y_i] = N\mu \quad (8.2.18)$$

여기에서  $Y_i$ 는 서로 독립인 베르누이분포이다.

- 분산

$$\text{Var}[X] = N\mu(1 - \mu) \quad (8.2.19)$$

(증명)

$$\text{Var}[X] = \text{Var} \left[ \sum_{i=1}^N Y_i \right] = \sum_{i=1}^N \text{Var}[Y_i] = N\mu(1 - \mu) \quad (8.2.20)$$

### 연습 문제 8.2.3

이항 확률분포의 모수가 다음과 같을 경우에 각각 표본을 생성한 후 기댓값과 분산을 구하고 앞의 예제와 같이 확률밀도함수와 비교한 바 플롯을 그린다.

표본 개수가 10개인 경우와 1000개인 경우에 대해 각각 위의 계산을 한다.

(1)  $\mu = 0.5, N = 5$

(2)  $\mu = 0.9, N = 20$

## 베르누이분포와 이항분포의 모수추정

데이터에서 모수의 값을 찾아내는 것을 **모수추정(parameter estimation)**이라고 한다.

베르누이분포와 이항분포는 공통된 모수  $\mu$ 를 가진다. 보통은 표본 데이터가 한 개보다 많기 때문에 이항분포가 된다. 추정된 값  $\hat{\mu}$ 은 다음처럼 계산한다.

$$\hat{\mu} = \frac{\sum_{i=1}^N x_i}{N} = \frac{N_1}{N} \quad (8.2.21)$$

이 식에서  $N$ 은 전체 데이터의 수,  $N_1$ 은 1이 나온 횟수이다.

## 베르누이분포의 활용

베르누이분포는 다음과 같은 경우에 사용된다.

1. 분류예측 문제의 출력 데이터가 두 값으로 구분되는 카테고리값인 경우에 분류 결과 즉, 두 값 중 어느 값이 가능성이 높은지를 표현하는 데 사용된다(베이지안 관점).
2. 입력 데이터가 0 또는 1 혹은 참 또는 거짓, 두 개의 값으로 구분되는 카테고리값인 경우, 두 종류의 값이 나타나는 비율을 표현하는 데 사용된다(빈도주의적 관점).

### 예제

스팸 메일(spam mail)과 정상 메일을 구분해야 하는 스팸 메일 필터(spam mail filter)를 만들어야 한다고 가정하자. 스팸 메일 필터를 만들기 위해 특정한 메일 계정으로 들어오는 메일을 모두 분석하였했다. 만약 메일이 모두 10통이고 이 중 6통이 스팸 메일이라면 이 메일 계정으로 들어온 메일이 스팸 메일일 가능성은 60%이다. 이러한 상황은  $\mu = 0.6$ 인 베르누이분포로 나타낼 수 있다.

$$p(y) = \text{Bern}(y; \mu = 0.6) \quad (8.2.22)$$

확률변수  $Y$ 는 메일이 스팸 메일인지 아닌지를 나타내며  $Y = 1$ 이면 스팸 메일이다.

### 예제

이번에는 스팸 메일 필터의 입력 데이터에서 베르누이분포를 활용하는 방법을 알아보자. 스팸 메일은 특정한 단어(키워드)를 가지고 있을 확률이 높다. 스팸 메일을 구분하기 위한 키워드가 여러 개라면 다음과 같이 BOW(Bag of Words) 인코딩된 벡터로 나타낼 수 있다. 이 예에서는 4개의 키워드 4개를 사용하였했다. 만약 어떤 메일이 첫 번째와 세 번째 키워드를 포함하고 있으며 두 번째와 네 번째 키워드는 포함하지 않으면 다음과 같은 특징 벡터로 표시할 수 있다.

$$x = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad (8.2.23)$$

여러개의 메일이 있으면 다음처럼 특징 행렬로 표시한다. 특징 행렬에서는 행 벡터가 메일을, 열 벡터가 키워드를 나타낸다.

$$X_{\text{spam}} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix} \quad (8.2.24)$$

이때, 스팸 메일의 특성은 4개의 베르누이 확률변수의 튜플  $(X_1, X_2, X_3, X_4)$ 로 나타낼 수 있다.

- $X_1$ : 메일이 첫 번째 키워드를 포함하면 1, 아니면 0이 되는 확률변수

$$p(X_1 = 1|Y = 1) = \text{Bern}(x_1; \mu_{\text{spam},1}) \quad (8.2.25)$$

- $X_2$ : 메일이 두 번째 키워드를 포함하면 1, 아니면 0이 되는 확률변수

$$p(X_2 = 1|Y = 1) = \text{Bern}(x_2; \mu_{\text{spam},2}) \quad (8.2.26)$$

- $X_3$ : 메일이 세 번째 키워드를 포함하면 1, 아니면 0이 되는 확률변수

$$p(X_3 = 1|Y = 1) = \text{Bern}(x_3; \mu_{\text{spam},3}) \quad (8.2.27)$$

- $X_4$ : 메일이 네 번째 키워드를 포함하면 1, 아니면 0이 되는 확률변수

$$p(X_4 = 1|Y = 1) = \text{Bern}(x_4; \mu_{\text{spam},4}) \quad (8.2.28)$$

특징 행렬의 각 열로부터 각 베르누이 확률분포의 모수의 추정값을 구하면 다음과 같다.

$$\hat{\mu}_{\text{spam},1} = \frac{5}{6}, \quad \hat{\mu}_{\text{spam},2} = \frac{4}{6}, \quad \hat{\mu}_{\text{spam},3} = \frac{3}{6}, \quad \hat{\mu}_{\text{spam},4} = \frac{3}{6} \quad (8.2.29)$$

#### 연습 문제 8.2.4

(1) 위 예제와 같이 키워드 4개를 사용하는 스팸 메일 필터를 만드는 데 스팸 메일이 아닌 정상 메일 5통을 다음과 같은 특징 행렬로 표시한다고 가정하자.

$$X_{\text{ham}} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8.2.30)$$

이 정상 메일에 각각의 키워드가 있을 확률을 나타내는 베르누이 확률분포

$$p(X_1 = 1|Y = 0) = \text{Bern}(x_1; \mu_{\text{ham},1}) \quad (8.2.31)$$

$$p(X_2 = 1|Y = 0) = \text{Bern}(x_2; \mu_{\text{ham},2}) \quad (8.2.32)$$

$$p(X_3 = 1|Y = 0) = \text{Bern}(x_3; \mu_{\text{ham},3}) \quad (8.2.33)$$

$$p(X_4 = 1|Y = 0) = \text{Bern}(x_4; \mu_{\text{ham},4}) \quad (8.2.34)$$

의 모수를 구하라. 우리가 구한 모수에 문제점은 없는가?

(2) 키워드 4개를 사용하는 스팸 메일 필터를 만드는 데 스팸 메일과 정상 메일의 특성을 모두 모형화하려면 베르누이 확률변수가 몇 개 필요한가?