연속형 확률변수
연속형 확률분포

준비

In [1] :

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

%precision 3
%matplotlib inline
```

In [2] :

```
from scipy import integrate
import warnings

# 적분에 관한 warning을 출력하지 않도록 한다
warnings.filterwarnings('ignore',
                        category=integrate.IntegrationWarning)
```

1차형 연속형 확률변수

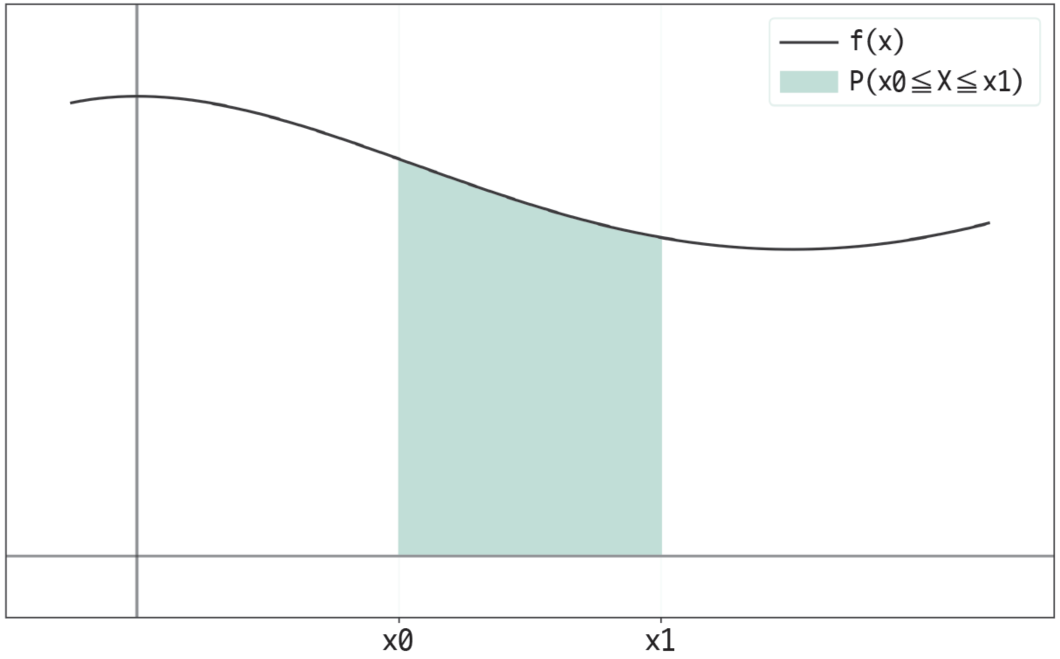
- 연속형 확률변수
  - 확률변수가 취할 수 있는 값이 연속적인 확률변수
  - 특정 값을 취하는 확률은 정의되지 않음
  - 확률변수가 어느 구간에 들어가는 확률을 정의
  - [예] 룰렛
    - 취할 수 있는 값이 0부터 1사이의 실수
    - 큰 수일수록 나오기 쉬워지는 불공정한 구조
    - 0.5라는 값을 취할 확률은 0
    - 정확하게 0.5000000…을 취할 가능성은 없으므로 확률은 0

1차형 연속형 확률변수

확률밀도함수

- 확률변수가 취할 수 있는 값은 구간 [a, b]
- 확률은 확률밀도함수(PDF) 또는 밀도함수 $f(x)$에 의해 정의
- 어떤 특정 값을 취하는 확률로는 정의되지 않음
- $f(x) \neq P(X = x)$

$$P(x_0 \leq X \leq x_1) = \int_{x_0}^{x_1} f(x)dx$$



[그림 7-1] 밀도함수로 정의되는 확률

1차형 연속형 확률변수

- 확률밀도함수

$$f(x) = \begin{cases} 2x & (0 \leq x \leq 1) \\ 0 & (otherwise) \end{cases}$$

In [3] :

```
x_range = np.array([0, 1])
```

In [4] :

```
def f(x):
    if x_range[0] <= x <= x_range[1]:
        return 2 * x
    else:
        return 0
```

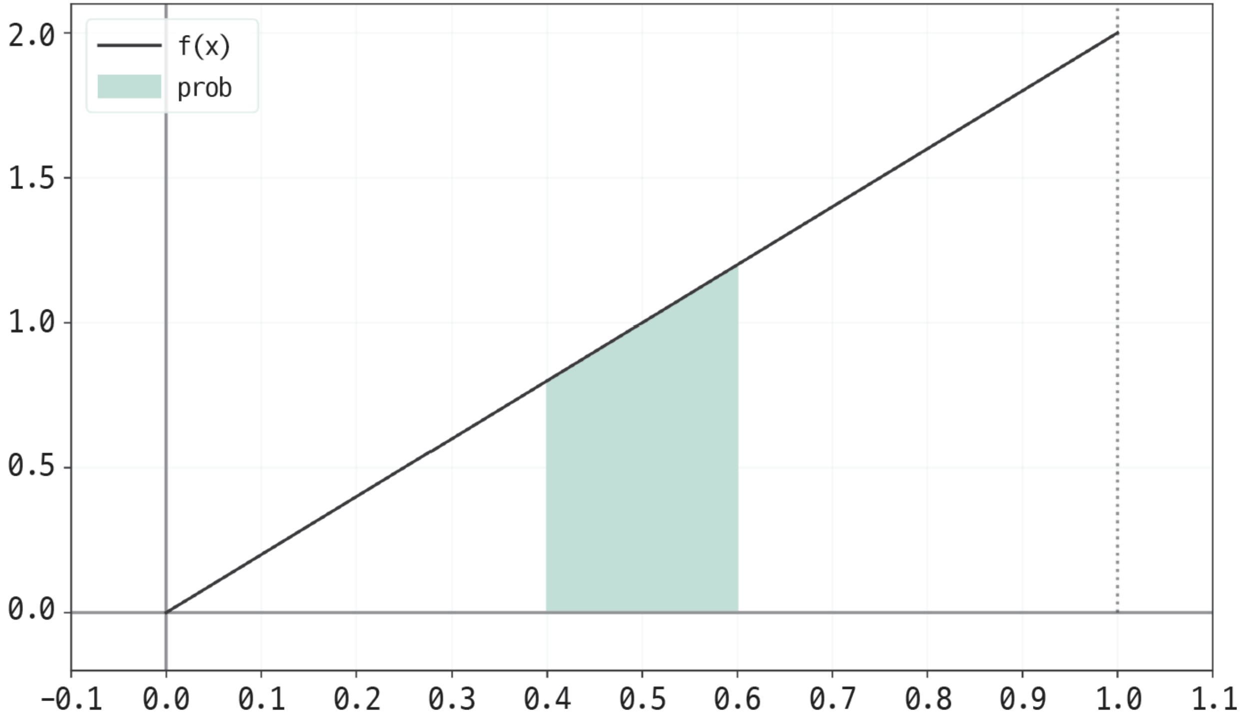
In [5] :

```
X = [x_range, f]
```

In [6] :

```
xs = np.linspace(x_range[0], x_range[1], 100)

fig = plt.figure(figsize=(10, 6))
ax = fig.add_subplot(111)

ax.plot(xs, [f(x) for x in xs], label='f(x)', color='gray')
ax.hlines(0, -0.2, 1.2, alpha=0.3)
ax.vlines(0, -0.2, 2.2, alpha=0.3)
ax.vlines(xs.max(), 0, 2.2, linestyles=':', color='gray')

# 0.4부터 0.6까지 x좌표를 준비
xs = np.linspace(0.4, 0.6, 100)
# xs의 범위로 f(x)와 x축으로 둘러싸인 영역에 색을 적용
ax.fill_between(xs, [f(x) for x in xs], label='prob')

ax.set_xticks(np.arange(-0.2, 1.3, 0.1))
ax.set_xlim(-0.1, 1.1)
ax.set_ylim(-0.2, 2.1)
ax.legend()

plt.show()
```

1차형 연속형 확률변수



[그림 7-2] 불공정한 룰렛에 대한 밀도함수와 확률

$$P(0.4 \le X \le 0.6) = \int_{0.4}^{0.6} 2x\, dx$$

$$\int_{0.4}^{0.6} 2x dx = 0.6^2 - 0.4^2 = 0.36 - 0.16 = 0.2$$

In [7] :

```
# 첫 번째 인수는 피적분함수, 두 번째 인수와 세 번째 인수는 적분 범위
integrate.quad(f, 0.4, 0.6)
```

Out [7] :

```
(0.200, 0.000)
```

1차형 연속형 확률변수

확률의 성질

연속형 확률변수에서는 확률의 성질로 다음 두 식을 만족해야 합니다.

$$f(x) \geq 0$$

$$\int_{-\infty}^{\infty} f(x)dx = 1$$

In [9] :

```
integrate.quad(f, -np.inf, np.inf)[0]
```

Out [9] :

```
1.000
```

In [8] :

```
from scipy.optimize import minimize_scalar

res = minimize_scalar(f)
# 함수의 최솟값은 fun이라는 인스턴스 변수에
res.fun
```

Out [8] :

```
0
```

```
In [9]: integrate.quad(f, -np.inf, np.inf)
Out[9]: (1.000, 0.000)
```

1차형 연속형 확률변수

```
In [9]: integrate.quad(f, -np.inf, np.inf)
Out[9]: (1.000, 0.000)
```

scipy.integrate.quad

scipy.integrate.quad(*func, a, b, args=(), full_output=0, epsabs=1.49e-08, epsrel=1.49e-08, limit=50, points=None, weight=None, wvar=None, wopts=None, maxp1=50, limlst=50*) [source]

Compute a definite integral.

Integrate func from *a* to *b* (possibly infinite interval) using a technique from the Fortran library QUADPACK.

Examples

Calculate $\int_0^4 x^2\,dx$ and compare with an analytic result

```
>>> from scipy import integrate
>>> x2 = lambda x: x**2
>>> integrate.quad(x2, 0, 4)
(21.333333333333332, 2.3684757858670003e-13)
>>> print(4**3 / 3.)  # analytical result
21.3333333333
```

Calculate $\int_0^\infty e^{-x}\,dx$

```
>>> invexp = lambda x: np.exp(-x)
>>> integrate.quad(invexp, 0, np.inf)
(1.0, 5.842605999138044e-11)
```

1차형 연속형 확률변수

- 누적분포함수

$$F(x) = P(X \leq x) = \int_{-\infty}^{x} f(x)dx$$

In [10] :

```
def F(x):
    return integrate.quad(f, -np.inf, x)[0]
```

$$P(0.4 \leq X \leq 0.6) = F(0.6) - F(0.4)$$

In [11] :

```
F(0.6) - F(0.4)
```

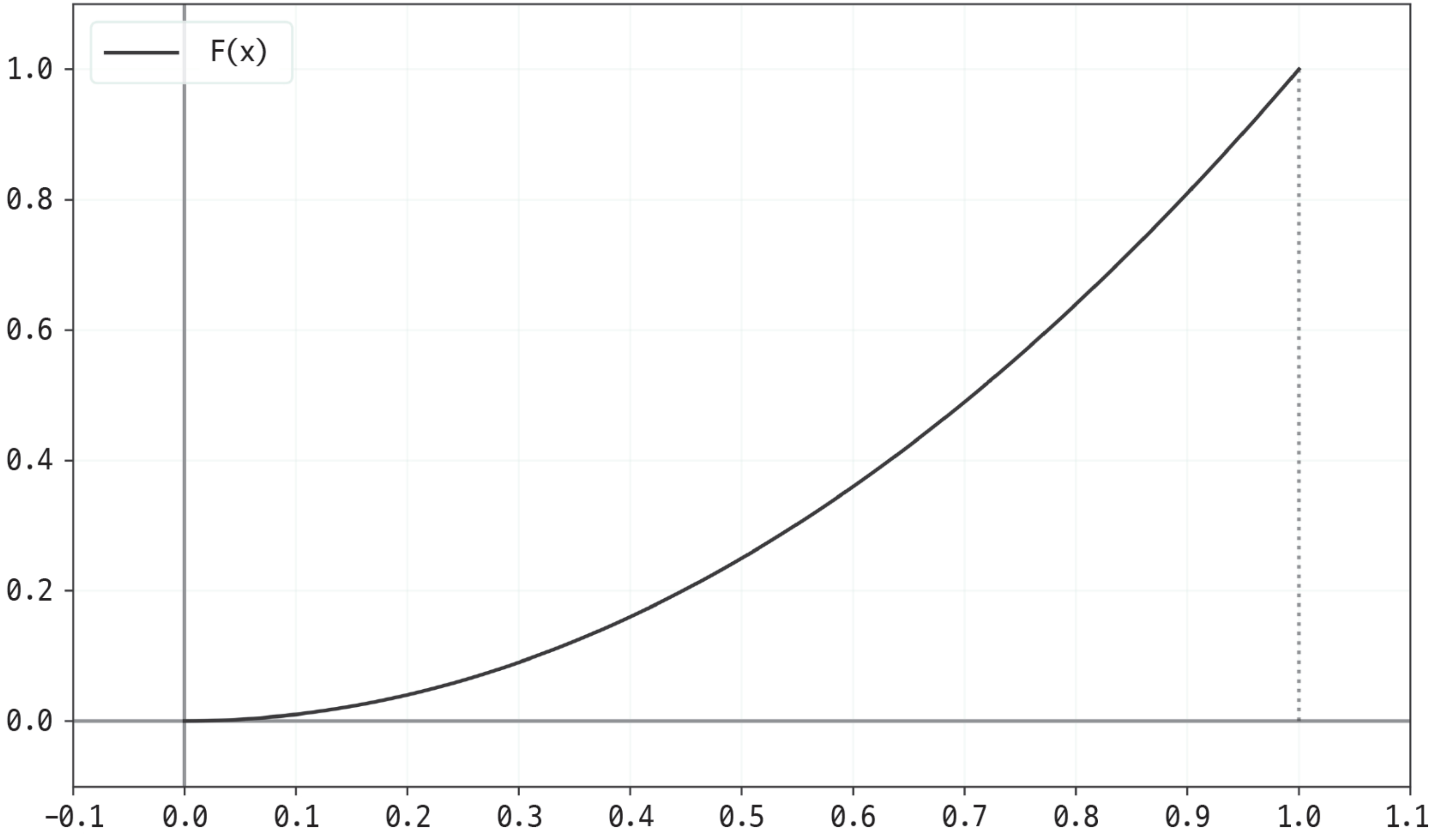
Out [11] :

```
0.200
```

In [12] :

```
xs = np.linspace(x_range[0], x_range[1], 100)

fig = plt.figure(figsize=(10, 6))
ax = fig.add_subplot(111)

ax.plot(xs, [F(x) for x in xs], label='F(x)', color='gray')
ax.hlines(0, -0.1, 1.1, alpha=0.3)
ax.vlines(0, -0.1, 1.1, alpha=0.3)
ax.vlines(xs.max(), 0, 1, linestyles=':', color='gray')

ax.set_xticks(np.arange(-0.1, 1.2, 0.1))
ax.set_xlim(-0.1, 1.1)
ax.set_ylim(-0.1, 1.1)
ax.legend()

plt.show()
```

1차형 연속형 확률변수



[그림 7-3] 분포함수

1차형 연속형 확률변수

확률변수의 변환

- X가 연속형 확률변수
- $Y = 2X + 3$인 Y도 연속형 확률변수
- Y의 확률밀도함수 $g(y)$

$$g(y) = \begin{cases} \dfrac{(y-3)}{2} & (3 \le y \le 5) \\ 0 & (otherwise) \end{cases}$$

- 확률분포함수 $G(y)$

$$G(y) = P(Y \le y) = \int_{-\infty}^{y} g(y)dy$$

In [13] :

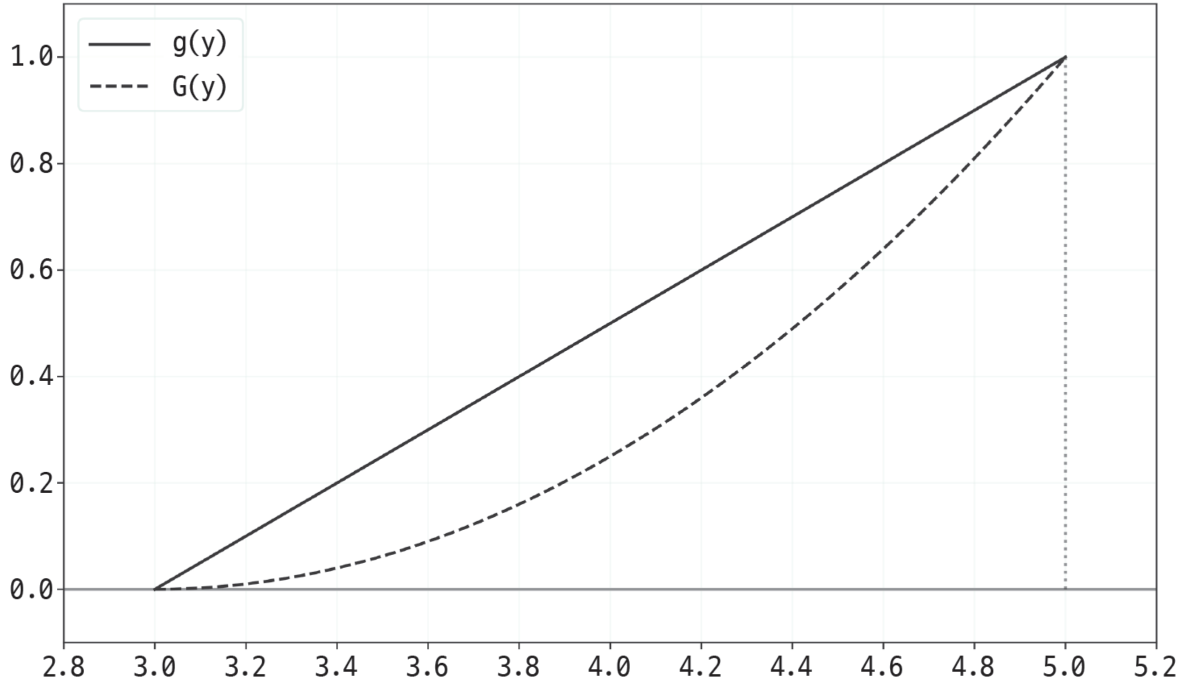
```python
y_range = [3, 5]

def g(y):
    if y_range[0] <= y <= y_range[1]:
        return (y - 3) / 2
    else:
        return 0

def G(y):
    return integrate.quad(g, -np.inf, y)[0]
```

1차형 연속형 확률변수

▪확률밀도함수와 확률분포함수

In [14] :

```
ys = np.linspace(y_range[0], y_range[1], 100)

fig = plt.figure(figsize=(10, 6))
ax = fig.add_subplot(111)

ax.plot(ys, [g(y) for y in ys],
        label='g(y)', color='gray')
ax.plot(ys, [G(y) for y in ys],
        label='G(y)', ls='--', color='gray')
ax.hlines(0, 2.8, 5.2, alpha=0.3)
ax.vlines(ys.max(), 0, 1, linestyles=':', color='gray')

ax.set_xticks(np.arange(2.8, 5.2, 0.2))
ax.set_xlim(2.8, 5.2)
ax.set_ylim(-0.1, 1.1)
ax.legend()

plt.show()
```



[그림 7-4] 밀도함수와 분포함수

1차형 연속형 확률변수의 지표

- 기댓값

$$\mu = E(X) = \int_{-\infty}^{\infty} x f(x) dx$$

- 불공정한 룰렛의 기댓값

In [15] :

```
def integrand(x):
    return x * f(x)

integrate.quad(integrand, -np.inf, np.inf)[0]
```

Out [15] :

```
0.667
```

- 변환환 확률변수의 기댓값

$$E(Y) = E(2X+3) = \int_{-\infty}^{\infty} (2x+3) f(x) dx$$

1차형 연속형 확률변수의 지표

연속형 확률변수의 기댓값

$$E(g(X)) = \int_{-\infty}^{\infty} g(x)f(x)dx$$

In [16] :

```
def E(X, g=lambda x: x):
    x_range, f = X
    def integrand(x):
        return g(x) * f(x)
    return integrate.quad(integrand, -np.inf, np.inf)[0]
```

- X 의 기댓값

In [17] :

```
E(X)
```

Out [17] :

```
0.667
```

- $Y = 2X + 3$의 기댓값

In [18] :

```
E(X, g=lambda x: 2*x+3)
```

Out [18] :

```
4.333
```

- 연속형 확률변수도 기댓값의 선형성 성립
  - E(2X+3) = 2E(X)+3

In [19] :

```
2 * E(X) + 3
```

Out [19] :

```
4.333
```

1차형 연속형 확률변수의 지표

- 분산

$$\sigma^2 = V(X) = \int_{-\infty}^{\infty} (x-\mu)^2 f(x)dx$$

- 불공정한 룰렛의 분산

In [20] :

```
mean = E(X)
def integrand(x):
    return (x - mean) ** 2 * f(x)

integrate.quad(integrand, -np.inf, np.inf)[0]
```

Out [20] :

```
0.056
```

- 변환환 확률변수의 분산

$$V(Y) = V(2X+3) = \int_{-\infty}^{\infty} ((2x+3) - \mu)^2 f(x)dx$$

1차형 연속형 확률변수의 지표

연속형 확률변수의 분산

$$V(g(X)) = \int_{-\infty}^{\infty} (g(x) - E(g(X)))^2 f(x) dx$$

In [21] :

```
def V(X, g=lambda x: x):
    x_range, f = X
    mean = E(X, g)
    def integrand(x):
        return (g(x) - mean) ** 2 * f(x)
    return integrate.quad(integrand, -np.inf, np.inf)[0]
```

- X 의 분산

In [22] :

```
V(X)
```

Out [22] :

```
0.056
```

- $Y = 2X + 3$의 분산

In [23] :

```
V(X, lambda x: 2*x + 3)
```

Out [23] :

```
0.222
```

1차형 연속형 확률변수의 지표

- $V(2X + 3) = 2^2 V(X)$

In [24] :

```
2**2 * V(X)
```

Out [24] :

```
0.222
```

분산의 공식

a, b를 실수, X를 확률변수라고 하면

$$V(aX + b) = a^2 V(X)$$

가 성립합니다.

2차형 연속형 확률변수

- 결합확률밀도함수
  - 2차원 연속형 확률변수 (X,Y)는 $\{(x,y)\,|\,a \le x \le b;\, c \le y \le d\}$와 $f(x,y)$ 에 의해 정의

    $f(x,y)$ 는 결합확률밀도함수

    $$P(x_0 \le X \le x_1,\, y_0 \le Y \le y_1) = \int_{x_0}^{x_1}\int_{y_0}^{y_1} f(x,y)\,dx\,dy$$

    - 예: 불공정한 룰렛 A, B
      - A와 B를 더한 것이 X, A를 Y로 한 2차원 확률변수 (X,Y)

$$\{0 \le X \le 2,\ 0 \le Y \le 1\}$$

$$f(x,y) = \begin{cases} 4y(x-y) & (0 \le y \le 1 \quad \text{및} \quad 0 \le x-y \le 1) \\ 0 & (otherwise) \end{cases}$$

2차형 연속형 확률변수

- 확률의 성질

$$f(x, y) \geq 0$$

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x, y) = 1$$

$$f(x,y) = \begin{cases} 4y(x-y) & (0 \leq y \leq 1 \quad \text{및} \quad 0 \leq x-y \leq 1) \\ 0 & (otherwise) \end{cases}$$

In [25] :

```
x_range = [0, 2]
y_range = [0, 1]
```

In [26] :

```
def f_xy(x, y):
    if 0 <= y <= 1 and 0 <= x - y <= 1:
        return 4 * y * (x - y)
    else:
        return 0
```

In [27] :

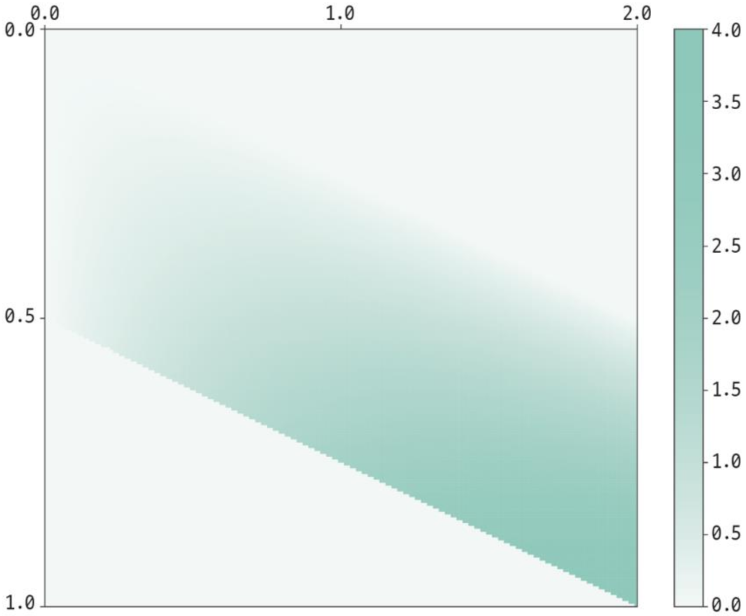
```
XY = [x_range, y_range, f_xy]
```

2차형 연속형 확률변수

- 결합확률밀도함수

In [28] :

```
xs = np.linspace(x_range[0], x_range[1], 200)
ys = np.linspace(y_range[0], y_range[1], 200)
pd = np.array([[f_xy(x, y) for y in ys] for x in xs])

fig = plt.figure(figsize=(10, 8))
ax = fig.add_subplot(111)

c = ax.pcolor(pd)
ax.set_xticks(np.linspace(0, 200, 3), minor=False)
ax.set_yticks(np.linspace(0, 200, 3), minor=False)
ax.set_xticklabels(np.linspace(0, 2, 3))
ax.set_yticklabels(np.linspace(0, 1, 3))
ax.invert_yaxis()
ax.xaxis.tick_top()
fig.colorbar(c, ax=ax)
plt.show()
```



[그림 7-5] 히트맵

In [29] :

```
# 첫 번째 인수는 피적분함수, 두 번째 인수는 x의 적분구간과 y의 적분구간
integrate.nquad(f_xy,
                [[-np.inf, np.inf],
                 [-np.inf, np.inf]])[0]
```

Out [29] :

```
1.000
```

2차형 연속형 확률변수

- 주변확률밀도함수
  - 확률변수 X만의 움직임

$$f_X(x) = \int_{-\infty}^{\infty} f(x, y)dy$$

In [32] :

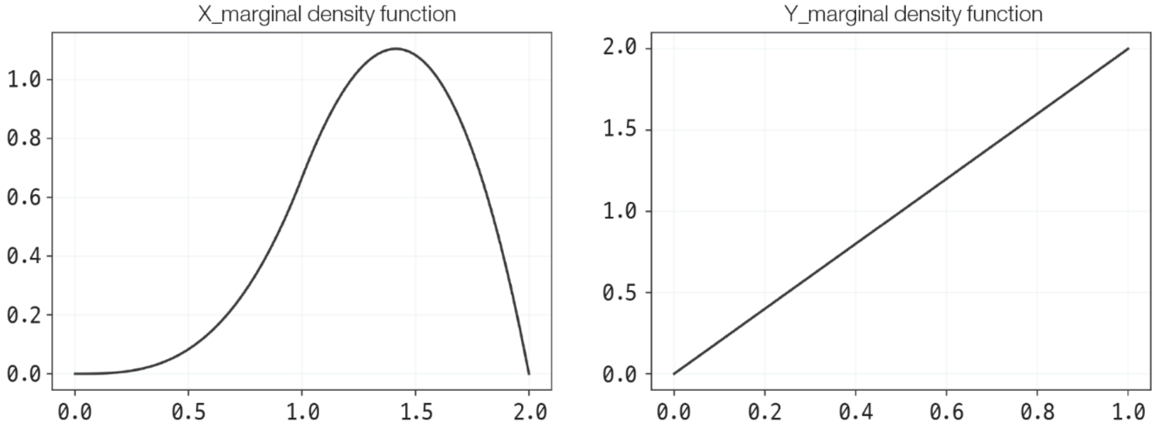
```
xs = np.linspace(*x_range, 100)
ys = np.linspace(*y_range, 100)

fig = plt.figure(figsize=(12, 4))
ax1 = fig.add_subplot(121)
ax2 = fig.add_subplot(122)
ax1.plot(xs, [f_X(x) for x in xs], color='gray')
ax2.plot(ys, [f_Y(y) for y in ys], color='gray')
ax1.set_title('X_marginal density function')
ax2.set_title('Y_marginal density function')

plt.show()
```

In [30] :

```
from functools import partial

def f_X(x):
    return integrate.quad(partial(f_xy, x), -np.inf, np.inf)[0]
def f_Y(y):
    return integrate.quad(partial(f_xy, y=y), -np.inf, np.inf)[0]
```

In [31] :

```
X = [x_range, f_X]
Y = [y_range, f_Y]
```



[그림 7-6] 주변밀도함수

2차형 연속형 확률변수의 지표

- 기댓값

$$\mu_X = E(X) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x f(x,y)dx\,dy$$

$$E(g(X,\ Y)) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x,y)f(x,y)dx\,dy$$

In [33] :

```
def integrand(x, y):
    return x * f_xy(x, y)

integrate.nquad(integrand,
                [[-np.inf, np.inf],
                 [-np.inf, np.inf]])[0]
```

Out [33] :

```
1.333
```

In [34] :

```
def E(XY, g):
    x_range, y_range, f_xy = XY
    def integrand(x, y):
        return g(x, y) * f_xy(x, y)

    return integrate.nquad(integrand,
                           [[-np.inf, np.inf],
                            [-np.inf, np.inf]])[0]
```

2차형 연속형 확률변수의 지표

- X와 Y의 기댓값

In [35] :

```
mean_X = E(XY, lambda x, y: x)
mean_X
```

Out [35] :

```
1.333
```

In [36] :

```
mean_Y = E(XY, lambda x, y: y)
mean_Y
```

Out [36] :

```
0.667
```

기댓값의 선형성 $E(2X+3Y)=2E(X)+3E(Y)$

In [37] :

```
a, b = 2, 3
```

In [38] :

```
E(XY, lambda x, y: a*x + b*y)
```

Out [38] :

```
4.667
```

In [39] :

```
a * mean_X + b * mean_Y
```

Out [39] :

```
4.667
```

2차형 연속형 확률변수의 지표

- X와 Y의 기댓값

In [35] :

```
mean_X = E(XY, lambda x, y: x)
mean_X
```

Out [35] :

```
1.333
```

In [36] :

```
mean_Y = E(XY, lambda x, y: y)
mean_Y
```

Out [36] :

```
0.667
```

기댓값의 선형성 $E(2X+3Y)=2E(X)+3E(Y)$

In [37] :

```
a, b = 2, 3
```

In [38] :

```
E(XY, lambda x, y: a*x + b*y)
```

Out [38] :

```
4.667
```

In [39] :

```
a * mean_X + b * mean_Y
```

Out [39] :

```
4.667
```

2차형 연속형 확률변수의 지표

- 분산

$$\sigma_X^2 = V(X) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(x-\mu_X)^2 f(x,y)\,dx\,dy$$

In [40] :

```
def integrand(x, y):
    return (x - mean_X)**2 * f_xy(x, y)

integrate.nquad(integrand,
                [[-np.inf, np.inf],
                 [-np.inf, np.inf]])[0]
```

Out [40] :

```
0.111
```

$g(X,Y)$의 분산

$$V(g(X,Y)) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(g(x,y)-E(g(X,Y)))^2 f(x,y)\,dx\,dy$$

In [41] :

```
def V(XY, g):
    x_range, y_range, f_xy = XY
    mean = E(XY, g)
    def integrand(x, y):
        return (g(x, y) - mean)**2 * f_xy(x, y)

    return integrate.nquad(integrand,
                           [[-np.inf, np.inf],
                            [-np.inf, np.inf]])[0]
```

2차형 연속형 확률변수의 지표

- X와 Y의 분산

In [42] :

```
var_X = V(XY, lambda x, y: x)
var_X
```

Out [42] :

```
0.111
```

In [43] :

```
var_Y = V(XY, lambda x, y: y)
var_Y
```

Out [43] :

```
0.056
```

- 공분산
  - 두 확률변수 (X, Y) 사이의 상관

$$\sigma_{XY} = Cov(X, Y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(x-\mu_X)(y-\mu_Y)f(x,y)dx\,dy$$

In [44] :

```
def Cov(XY):
    x_range, y_range, f_xy = XY
    mean_X = E(XY, lambda x, y: x)
    mean_Y = E(XY, lambda x, y: y)
    def integrand(x, y):
        return (x-mean_X) * (y-mean_Y) * f_xy(x, y)

    return integrate.nquad(integrand,
                           [[-np.inf, np.inf],
                            [-np.inf, np.inf]])[0]
```

In [45] :

```
cov_xy = Cov(XY)
cov_xy
```

Out [45] :

```
0.056
```

2차형 연속형 확률변수의 지표

- 분산과 공분산

a, b를 실수, X, Y를 확률변수로 했을 때

$$V(aX + bY) = a^2 V(X) + b^2 V(Y) + 2ab\,Cov(X, Y)$$

가 성립합니다.

In [46] :

```
V(XY, lambda x, y: a*x + b*y)
```

Out [46] :

```
1.611
```

In [47] :

```
a**2 * var_X + b**2 * var_Y + 2*a*b * cov_xy
```

Out [47] :

```
1.611
```

- 상관계수

$$\rho_{XY} = \rho(X, Y) = \frac{\sigma_{XY}}{\sigma_X \sigma_Y}$$

In [48] :

```
cov_xy / np.sqrt(var_X * var_Y)
```

Out [48] :

```
0.707
```