

1. Markdown and latex

1a.

If the random variable x obeys the normal distribution as follows $x \sim N(\mu, \sigma)$, including μ is the mean and σ is the standard deviation, the normal probability density function is like that:

$$P(x|\mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{1}$$

1b.

Probability theory states that when random variables are independent of one another, their joint probability equals the product of their probabilities. So, the joint probability density function (pdf) for random variables with the same mean and variance is given by: for $x_i \sim N(\mu, \sigma), i = 1 \dots N$

$$p(x_{1:N}|\mu, \sigma) = \prod_{i=1}^N p(x_i|\mu, \sigma) \tag{2}$$

Because random variables are indepentdent each other, variables don't affect the probability each pther. Then the joint probability of them is the product of the individual probabilities of each variable as follows.

$$p(x_{1:N}|\mu, \sigma) = (\frac{1}{\sqrt{2\pi\sigma^2}})^N e^{-\frac{1}{2} \sum_{i=1}^N (\frac{x_i-\mu}{\sigma})^2} \tag{3}$$

2. Simple functions and plotting

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

# Normal distribution Function
def g(x, mu, sigma):
    return (1 / (np.sqrt(2 * np.pi * (sigma ** 2)))) * np.exp(-(x - mu)
    ** 2 / (2 * sigma ** 2))

#### Main Function
# Design Construct the normal distribution function coordinate system using sigma ,mu and x1
sigma = 1.0
mu = 0
x = np.linspace(-4.0, 4.0, num=10000)
fig = plt.figure(figsize=(5,3))
plt.plot(x, g(x, mu, sigma))
plt.xlabel('$x$', fontsize=13)
plt.ylabel('$p(x)$', fontsize=13)
plt.title('$p(x | \mu=\{0\}, \sigma=\{1\})$')
plt.ylim([-0.009, 0.43])
plt.xlim([-4.0, 4.0])
x1 = 0.0
y1 = g(x1, mu, sigma)

# a circle mark at the likelihood for a point x1 with a connecting line to the x-axis.
plt.plot([x1, x1], [0.0, y1], markevery=[1], marker="o")

# an annotation that displays the numerical value of the likelihood to 3 digits.
plt.annotate('$p(x = %.1f | \mu, \sigma) = %.3f$'%(x1,g(x1,mu=mu, sigma=sigma)),
            xy =(x1, g(x1, mu=mu, sigma=sigma)))
plt.show()
```

