Assignment 1a: Jupyter Notebooks

Hunter Welch 2/2/2023

1. Markdown and LaTeX

Normal Probability Distribution Function

1a. The univariate normal probability distribution function (pdf), $p(x|\mu, \sigma)$, is commonly referred to as the Gaussian distribution. Gaussian distribution is the typical "bell-shaped" curve that is very important for statistics and seen a lot in everyday life. In the distribution, the mean, μ, is located in the center of the normal curve and is the maximum point on the graph. The standard deviation, σ, determines how wide the curve and how large the range of possibilities are. A larger σ denotes a larger range of possibilities while a smaller σ explains that most of the data points fall near the μ. The equation of a normal pdf is:

$$p(x|\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\right) \tag{1}$$

1b. The joint probability of independent random variables (with the same mean and variance) $p(x_{1:N}|\mu, \sigma)$ is the probability that an event occurs given all of these variables. The joint probability density function of two independent variables is always the product of their univariate pdfs. The function for the joint pdf is:

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

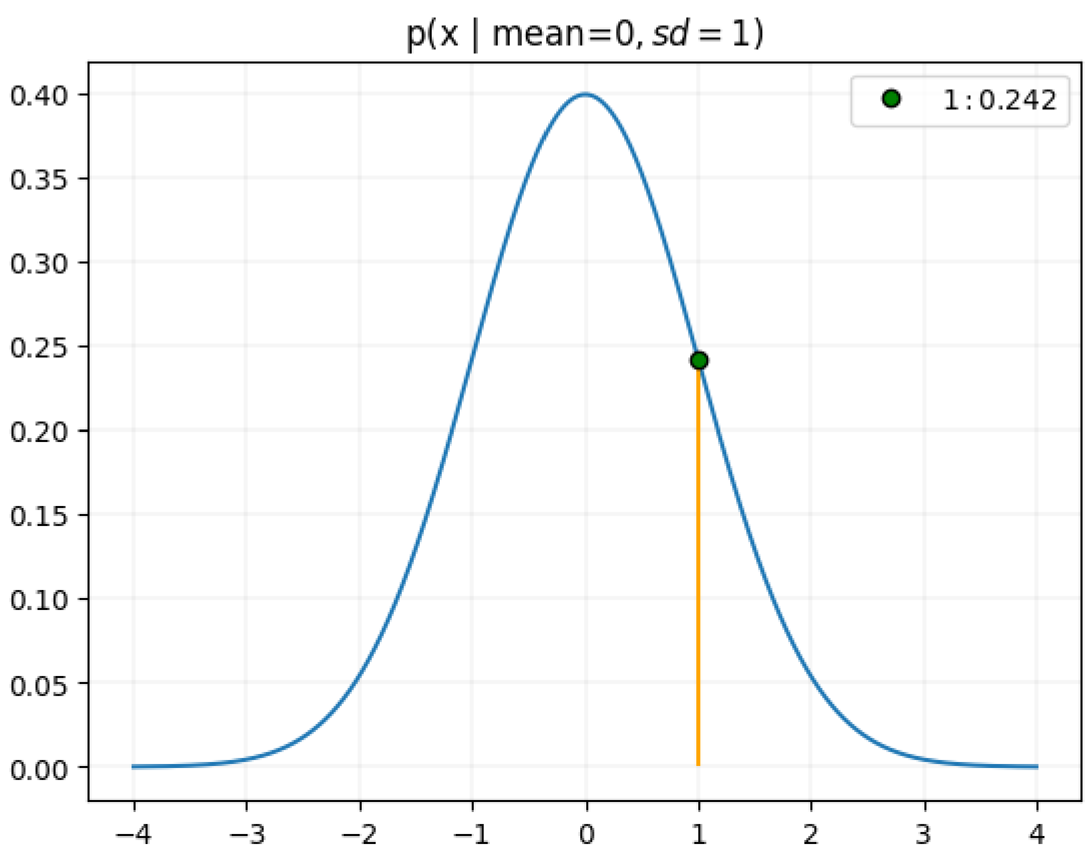
Since the μ and σ is the same for each function, we can use multiplication rules to keep all constants outside of our product equation and the joint pdf equation becomes:

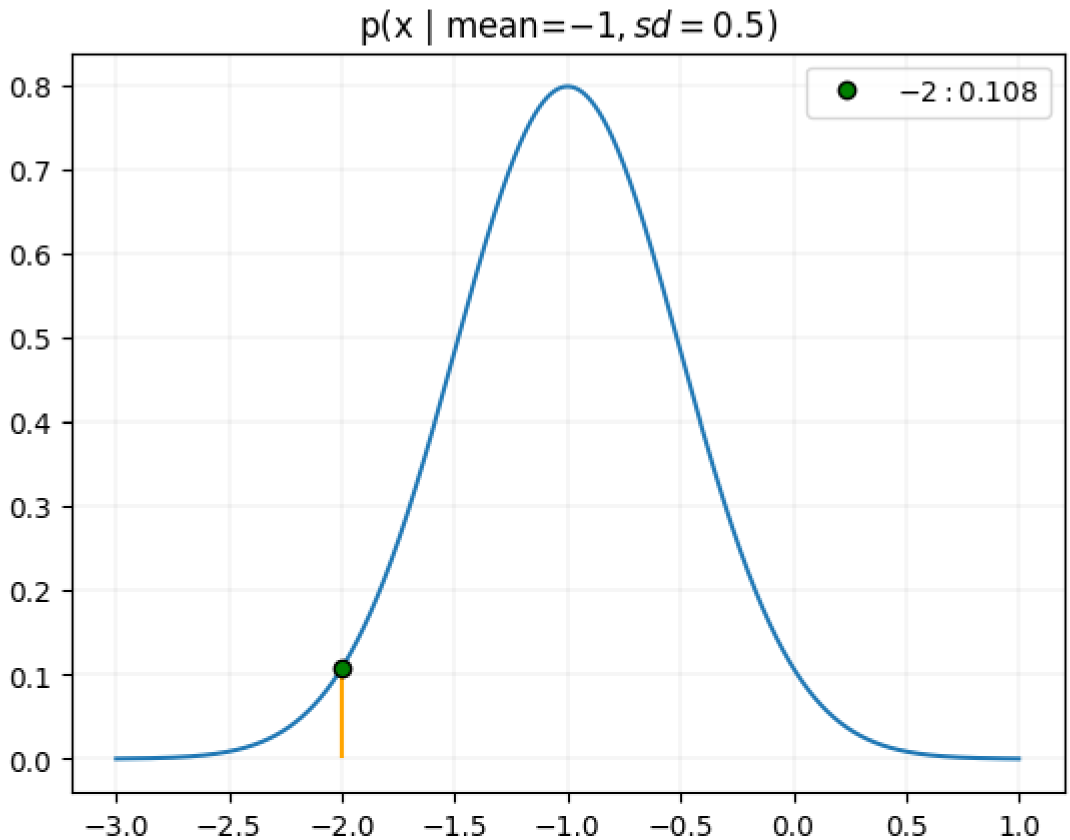
$$f(x_{1:N}) = \frac{1}{\left(\sigma\sqrt{2\pi}\right)^N} \exp\left(-\frac{1}{2}\sum_{i=1}^{N}\frac{(x-\mu)^2}{\sigma^2}\right) \tag{3}$$

2. Simple Functions and Plotting

In [8]:
```
import math
import matplotlib.pyplot as plt
import numpy as np

def compute_gaussian(x, mean, sd):
    return math.exp(-1/2 * ((x - mean) / sd)**2) / (sd * math.sqrt(2 * math.pi))
```

```python
def plot_gaussian(x1, mean, sd):
    x = np.arange(mean - (4.0 * sd), mean + (4.0 * sd), 0.001)
    y = []
    for i in x:
        y.append(compute_gaussian(i, mean, sd))
    y = np.array(y)
    plt.plot(x, y)

    # get the plot for vertical line to x1
    y1 = compute_gaussian(x1, mean, sd)
    plt.vlines(x1, ymin=0, ymax=y1, colors="orange")
    plt.plot(x1, y1, 'go', mec="black", label = f'${x1}: ${round(y1, 3)}')

    plt.grid(color='gray', linestyle='-', linewidth=0.1)
    plt.title(f'p(x | mean=${mean}, sd=${sd})')
    plt.legend()
    plt.show()


plot_gaussian(1, 0, 1)
plot_gaussian(-2, -1, 0.5)
```




p(x | mean=−1, sd = 0.5)
−2 : 0.108
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
−3.0
−2.5
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0