

# AAE718<sub>HW2</sub>

Jesmyn ZHANG

May 2024

## Problem 01

Assuming `np.random.rand(shape)` generates an array of random values between 0 and 1, we can map these values to be between 'a' and 'b' using a linear transformation. The specific mathematical formula is as follows:

$$y = a + (b - a) \times x$$

where 'x' is the random value between 0 and 1, and 'y' is the desired value between 'a' and 'b'. This formula linearly scales and shifts the values originally in the range  $[0, 1]$  to the new range  $[a, b]$ .

## Problem 07

## Problem 08

### Arrays vs. Lists

- **Sorting Arrays:** Takes about 0.60 to 0.62 seconds for 10,000,000 integers.
- **Sorting Lists:** Takes about 1.77 to 2.00 seconds for the same data, making arrays nearly three times faster.

### Random Number Range Effect

- Arrays take slightly longer to sort when numbers are between 0 and 1 (0.82 to 0.88 seconds).
- Lists' sorting times remain stable (1.70 to 1.92 seconds).

### Conclusion

- Arrays are significantly faster than lists for sorting large datasets.
- Vectorized operations in NumPy enhance performance.

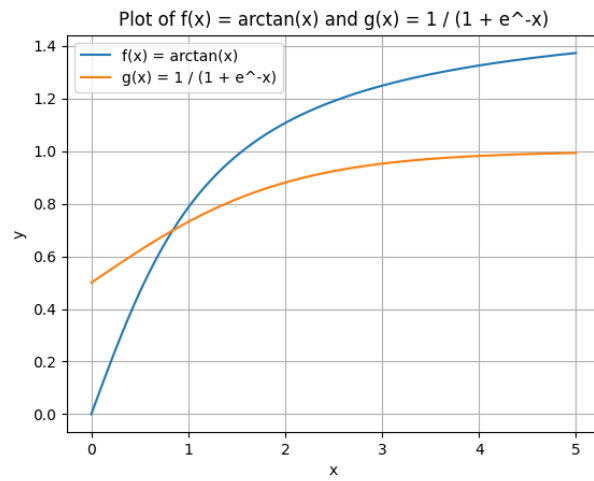


Figure 1: Plot arctan sigmoid

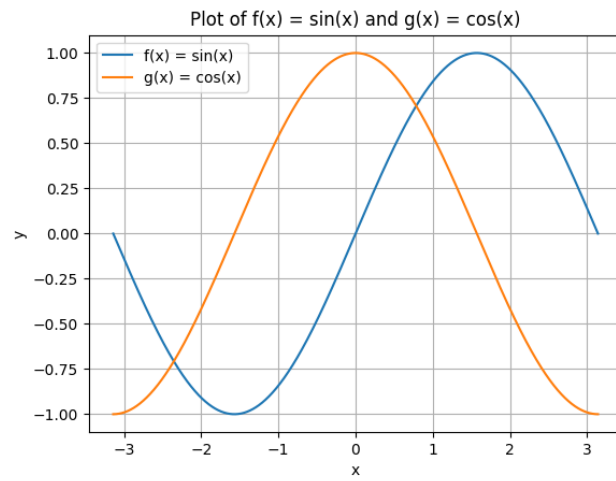


Figure 2: Plot sin cos

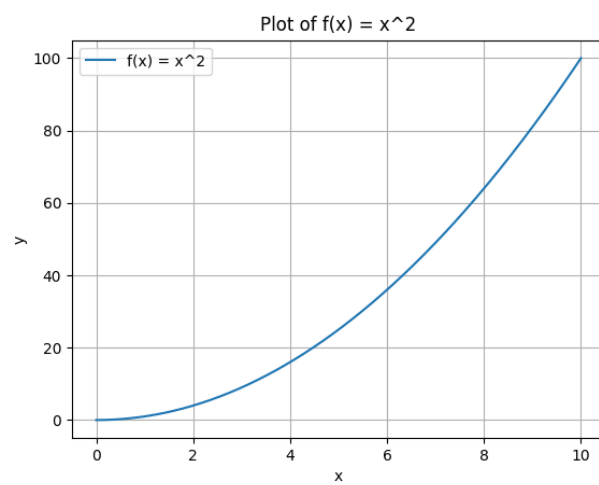


Figure 3: Plot  $x$  squared