LECTURE 2 (PART 2) - MARCH 15, 2024

DEFINING CUSTOM FUNCTIONS

THE BEST WAY TO DEVELOP AND MANTAIN A LARGE PROGRAM IS TO CONSTRUCT IT FROM SMALLER, MORE MANAGEABLE PIECES. FUNCTIONS HAVE THIS PURPOSE.

WE HAVE CALLED BUILT-IN FUNCTIONS LIKE (int, input, type, len, max, min). WE HAVE ALSO CALLED FUNCTIONS FROM statistics AND random LIBRARIES.

LET US DEFINE A FUNCTION THAT RETURNS THE SQUARE OF A NUMBER.

NAME OF THE FUNCTION   PARAMETER

```
[]: def square(number):
        return number **2
```

RETURN STATEMENT TO OBTAIN THE RESULT THROUGH FUNCTION'S NAME

```
[]: square(7)
```

```
[]: square(2,5)
```

A FUNCTION CAN HAVE MULTIPLE PARAMETERS AND STATEMENTS BEFORE THE RETURN.

GLOBAL AND LOCAL VARIABLES

ALL VARIABLES CREATED OUTSIDE OF A FUNCTION (BEFORE IT IS DEFINED) ARE "GLOBAL" AND CAN BE USED (BUT NOT CHANGED) FROM THE INSIDE OF A FUNCTION.

ALL VARIABLES CREATED INSIDE OF A FUNCTION ARE "LOCAL" AS THEY EXIST ONLY IN THE BODY OF THE FUNCTION AND CANNOT BE SEEN FROM OUTSIDE.

> IF A "GLOBAL" VARIABLE IS ASSIGNED A VALUE INSIDE OF A FUNCTION THIS WILL CREATE A "LOCAL" VARIABLE WITH THE SAME NAME, SO ANY CHANGE TO IT WON'T SURVIVE AFTER THE END OF THE FUNCTION EXECUTION.

```
[ ]: # Global variable
     a = 10
```

```
[ ]: def example1(x):
         # Local variable
         b = 10
         print('Inside a =', a)
         return x * a * b
```

← WE USE THE GLOBAL VARIABLE

```
[ ]: example1(2)
```

```
[ ]: print('Outside b =', b)
```

← ERROR! b IS LOCAL!

A FIRST LOOK TO NumPy AND SciPy LIBRARIES

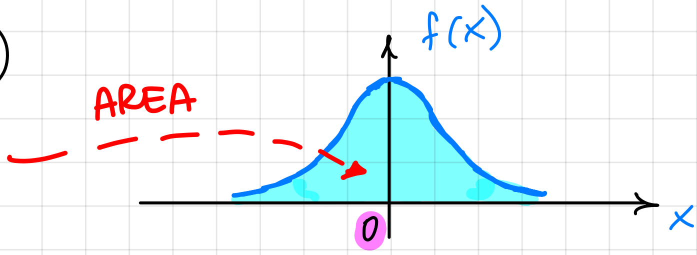
NumPy AND SciPy ARE THE MOST FAMOUS LIBRARIES FOR MATHEMATICS AND SCIENTIFIC CALCULUS.
IN PARTICULAR, NumPy IS USED FOR LINEAR ALGEBRA AS IT IMPLEMENTS ARRAYS (USED FOR VECTORS AND MATRICES).

WE WANT TO PLOT THE GRAPH OF THE STANDARD NORMAL PROBABILITY DENSITY FUNCTION

$$f(x) = \frac{1}{\sqrt{2\pi}} \cdot e^{-\frac{x^2}{2}}$$

AND COMPUTE (NUMERICALLY)

$$\int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx = 1$$



PROBLEM: f(x) IS A TYPICAL EXAMPLE OF FUNCTION THAT HAS A PRIMITIVE BUT IT CANNOT BE EXPRESSED THROUGH ELEMENTARY FUNCTIONS. MATHEMATICALLY

$$\int_{-\infty}^{+\infty} f(x)\,dx = \lim_{a \to -\infty} \int_{a}^{0} f(x)\,dx + \lim_{b \to +\infty} \int_{0}^{b} f(x)\,dx$$

```
[ ]: import numpy as np
     import matplotlib.pyplot as plt
     from scipy.integrate import quad

     # Define the standard normal density
     def f(x):
         return (1 / np.sqrt(2 * np.pi)) *
                                 np.exp(- (x**2) /2)
```

← TO COMPUTE INTEGRALS

-∞ +∞

```
# Compute the integral
I, err = quad(f, -np.infty, np.infty)
```

THE RESULT IS A TUPLE

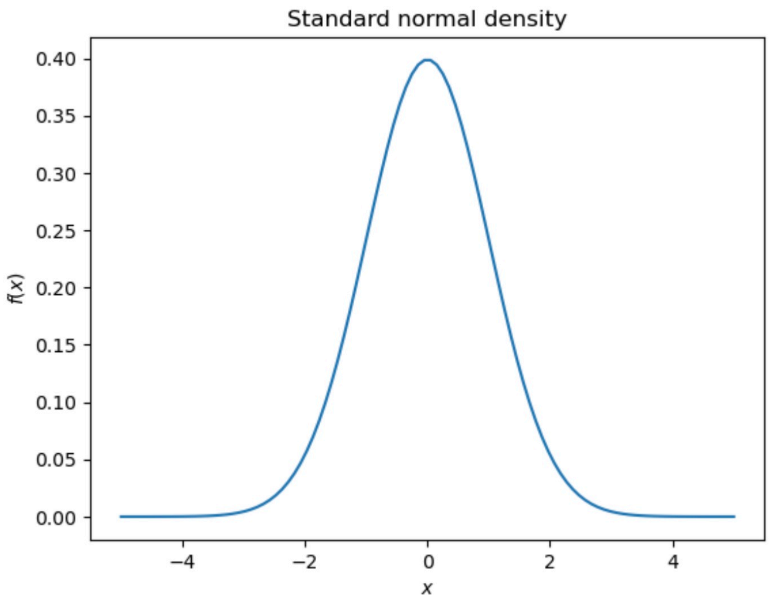
```
print('The integral is:', I)

# Sample some coordinates
x = np.linspace(-5, 5, 100)
```

IT IS AN ARRAY: WE WILL SEE ARRAYS IN NEXT LECTURES

```
y = f(x)
```

VECTORIZED APPLICATION

```
# Plot the graph
plt.title('Standard normal density')
plt.xlabel('$x$')
plt.ylabel('$f(x)$')
plt.plot(x,y)
```



EXERCISE: EXPECTED UTILITY

GIVEN TWO RANDOM MONETARY POSITIONS X, Y AND A UTILITY FUNCTION FOR MONEY $u(x)$, THE EXPECTED UTILITY PARADIGM TELLS US THAT

$$\boxed{\begin{array}{lll} X \text{ IS PREFERRED TO } Y & \Leftrightarrow & \mathbb{E}[u(X)] > \mathbb{E}[u(Y)] \\ X \text{ IS INDIFFERENT TO } Y & \Leftrightarrow & \mathbb{E}[u(X)] = \mathbb{E}[u(Y)] \end{array}}$$

CHOICE CRITERION: MAXIMIZE EXPECTED UTILITY.

GIVEN

$$X_1 \sim N(\overset{\mu}{10}, \overset{\sigma^2}{0.8^2}),\quad X_2 \sim N(9, 0.3^2),\quad X_3 \sim N(9, 0.4^2)$$

$$X_4 \sim N(10, 0.5^2),\quad X_5 \sim N(11, 0.6^2)$$

AND THE EXPONENTIAL UTILITY FUNCTION

$$u(x) = -e^{-ax} \qquad \text{WITH} \quad a = 0.1$$

FIND THE MOST PREFERRED RANDOM POSITION.

```
[ ]: import numpy as np
     from scipy.stats import norm
     from scipy.integrate import quad
```

← NORMAL DISTRIBUTION

```
[ ]: # General normal density function
     def f(x, mu, sigma):
         return norm.pdf(x, mu, sigma)
```

```
[ ]: # Exponential utility function
     def u(x, a):
         return -np.exp(-a * x)
```

IF X HAS PROBABILITY DENSITY FUNCTION f(x)

$$\mathbb{E}[u(X)] = \int_{-\infty}^{+\infty} \underbrace{u(x)\, f(x)}_{\text{EXPECTED UTILITY INTEGRAND}} dx$$

```
[ ]: # Expected utility integrand
     def f_eu(x, mu, sigma, a):
         return u(x, a) * f(x, mu, sigma)
```

```
[ ]: # Expected utility
     def expected_utility(mu, sigma, a):
         I, err = quad(f_eu, -np.infty, np.infty,
                              args = (mu, sigma, a))

         return I
```

$I = \mathbb{E}[u(X)]$

OTHER PARAMETERS FOR f_eu BESIDES X

```
[ ]: # Set the parameter of exponential utility
     a = 0.1
```

```
[ ]: expected_utility(10, 0.8, a)
```

```
[ ]: expected_utility(9, 0.3, a)
```

```
[ ]: expected_utility(9, 0.4, a)
```

```
[ ]: expected_utility(10, 0.5, a)
```

```
[ ]: expected_utility(11, 0.6, a)
```