Introducción a la programación

Usando Python

G. Sebastián Pedersen

Instituto de Industria
Universidad Nacional de General Sarmiento

Matemática para Economistas III, 1er. cuat. 2019

`https://sebasped.github.io/pythonungs/`

Resumen de conceptos importantes

- Variables: las utilizamos para guardar valores y reutilizarlos posteriormente (en una cuenta, imprimirlos, etc.)
- Ciclos `while` y `for`: se utilizan para hacer muchas veces lo mismo cuando a priori no sabemos la cantidad total de veces que lo tenemos que hacer.
- Condicional `if`: sirve para separa casos con comportamientos distintos.
- Funciones.
- Duplicación de información.
- Particionar y encapsular el problema.

¿Cómo se usaba una función?

Primero la defino:

```
def nombre_func(x,a):
    ...
    cuentas o lo que haga la función con la entrada
    ...

    return lo_que_devuelvo
```

¿Cuál es la entrada y cuál es la salida?

Y luego la utilizo:

```
salida = nombre_func(3,8)
print(salida)
```

En la utilización, ¿cuánto valen la entrada y la salida?

Resolviendo ecuaciones

Supongamos que quiero resolver la siguiente ecuación:

$$-3x = 2^x$$

O lo que es lo mismo

$$-3x - 2^x = 0$$

¿Cómo hago? Analíticamente no se puede, y entonces cobra sentido resolverlas numéricamente con la compu.
Existe varios métodos numéricos para dar una solución aproximada. Por ejemplo Newton-Raphson.

Resolviendo ecuaciones

Este programita resuelve la ecuación $-3x - 2^x = 0$ utilizando Newton-Raphson:

```
from scipy import optimize

a=-3
b=2

def f(x):
    func = a*x - b**x
    return func

# la entrada para optimize es la función y un valor inicial.
# resuelve por N-R la ecuación f=0
raiz = optimize.newton(f, 1.5)
print(raiz)

comprobar = a*raiz-b**raiz
print(comprobar)
```

La salida de `print(raiz)` da: -0.2754059364858210 7
La salida de `print(comprobar)` da: -2.220446049250313e-16

Calculando Yield to Maturity (YTM)

Si ahora quiero calcular la YTM me voy a encontrar con una ecuación de este estilo

$$p = \frac{c}{(1+y/n)^1} + \frac{c}{(1+y/n)^2} + \cdots + \frac{100+c}{(1+y/n)^{nT}}$$

Donde:

- p es el precio del bono
- c es el cupón (lo que paga en cada período en %)
- T son los años hasta el vencimiento.
- n es la frecuencia de pago por año.
- y es la YTM (lo que quiero calcular).

Entonces estamos en una situación similar a la anterior: la incógnita y no se puede despejar.
O lo que es lo mismo, calcular la raíz de:

$$\frac{c}{(1+y/n)^1} + \frac{c}{(1+y/n)^2} + \cdots + \frac{100+c}{(1+y/n)^{nT}} - p = 0$$

Por lo tanto tiene sentido encontrar una solución aproximada por Newton-Raphson utilizando la compu.

Calculando Yield to Maturity (YTM)

Resumiendo, tenemos una $f(y) = 0$:

$$\underbrace{\frac{c}{(1+y/n)^1} + \frac{c}{(1+y/n)^2} + \cdots + \frac{100+c}{(1+y/n)^{nT}} - p}_{f(y)} = 0$$

Donde:

- p, c, T y n son valores ya dados.
- Queremos encontrar el valor de y que satisface la ecuación.

¡Twist and Code!

Ejercicio:

1. Hacer un programa que resuelva por Newton-Raphson la YTM. Suponer que el precio p, el cupón c, los años hasta el vencimiento T y la frecuencia de pago por año n son parámetros que ya están dados (se pueden fijar antes de definir a la función. Ver la 2da. diapo de "Resolviendo ecuaciones").
2. Utilizar el programa para calcular algunas YTMs.