

# Gestión de inventarios Multiproducto (Python) Clase 25

Investigación Operativa UTN FRBA 2020

Curso: I4051

Elaborado por: Rodrigo Maranzana

Docente: Martín Palazzo

# Enunciado y datos

Se desea conocer la cantidad óptima de pedido y el costo total esperado en la gestión de inventario de dos productos. Se supone que sus cantidades son independientes y no existen restricciones adicionales.

```
1 diasmes = 30
2 interes = 0.1 # anual
3
4 # Datos producto 1:
5 b_1 = 30 #costo por producto
6
7 compra_1 = 100 # unidad
8 calidadrepcion_1 = 200 # pedido
9 demanda_1 = 3000 # por año
10 k_1 = calidadrepcion_1 + compra_1 # costo de orden
11 d_1 = demanda_1 # demanda
12 alquiler_1 = 30 # alquiler diario
13 ba_1 = alquiler_1 * diasmes * 12 # alquiler anual
14 i_1 = interes # mismo interés para ambos productos
15
16 # Datos producto 2:
17 b_2 = 40 #costo por producto
18 compra_2 = 150 # unidad
19 calidadrepcion_2 = 250 # pedido
20 demanda_2 = 4300 # por año
21 k_2 = calidadrepcion_2 + compra_2 # costo de orden
22 d_2 = demanda_2 # demanda
23 alquiler_2 = 40 # diario
24 ba_2 = alquiler_2 * diasmes * 12 # alquiler anual
25 i_2 = interes # mismo interés para ambos productos
```

# Componentes del costo total esperado de cada producto

## 3.1 Cálculo de costo de adquisición

Es el costo total que depende del costo unitario del producto  $b_i$  y la demanda del mismo  $d_i$ .

$$Cadq = b_i * d_i$$

```
1 def calcular_cadq(b, d):  
2     return b * d
```

# Componentes del costo total esperado de cada producto

## 3.2 Cálculo de costo de almacenamiento

Es el costo en que se incurre por almacenar un producto. Depende de la cantidad  $q_i$ , el costo unitario de almacenamiento  $c_{ui}$ .

$$Calm(q_i) = \frac{1}{2} * q_i * c_{ui}$$

El costo unitario  $c_{ui}$  se calcula de la siguiente forma:

$$c_{ui} = b_i * i + b_{ai}$$

Siendo  $b_i$  el costo del producto y  $b_{ai}$ , un costo de almacenamiento adicional por producto, en este ejemplo es el alquiler.

```
1 def calcular_cstock(q, b, i, ba):  
2     return 0.5 * q * (b * i + ba)
```

# Componentes del costo total esperado de cada producto

## 3.3 Cálculo de costo de pedido

Es el costo que surge cada vez que se hace un pedido de producto. Depende de  $k_i$ , el costo administrativo de pedido;  $d_i$ , la demanda de cada producto y  $q_i$ , la cantidad.

$$C_{pedido}(q_i) = k_i * \frac{d_i}{q_i}$$

```
1 def calcular_cpedido(k, d, q):  
2     return k * (d / q)
```

# Costo total esperado

$$CTE(q_i) = Cadq + Calm(q_i) + Cpedido(q_i)$$

```
1 # Costo total esperado:  
2 def calcular_cte(cdemanda, cstock, cpedido):  
3     return cdemanda + cstock + cpedido
```

## Costo total esperado de cada producto:

```
# Producto 1:  
cte_opt1 = calcular_cte(cdemanda1, cstock1, cpedido1)  
# Producto 2:  
cte_opt2 = calcular_cte(cdemanda2, cstock2, cpedido2)
```

## Costo total esperado de ambos productos:

```
# Total:  
cte_opt_t = cte_opt1 + cte_opt2
```

# Cantidad óptima de cada producto

## 4 Cálculo de cantidades óptimas:

La cantidad óptima está donde la derivada del Costo Total Esperado se iguala a cero. La fórmula analítica es la siguiente:

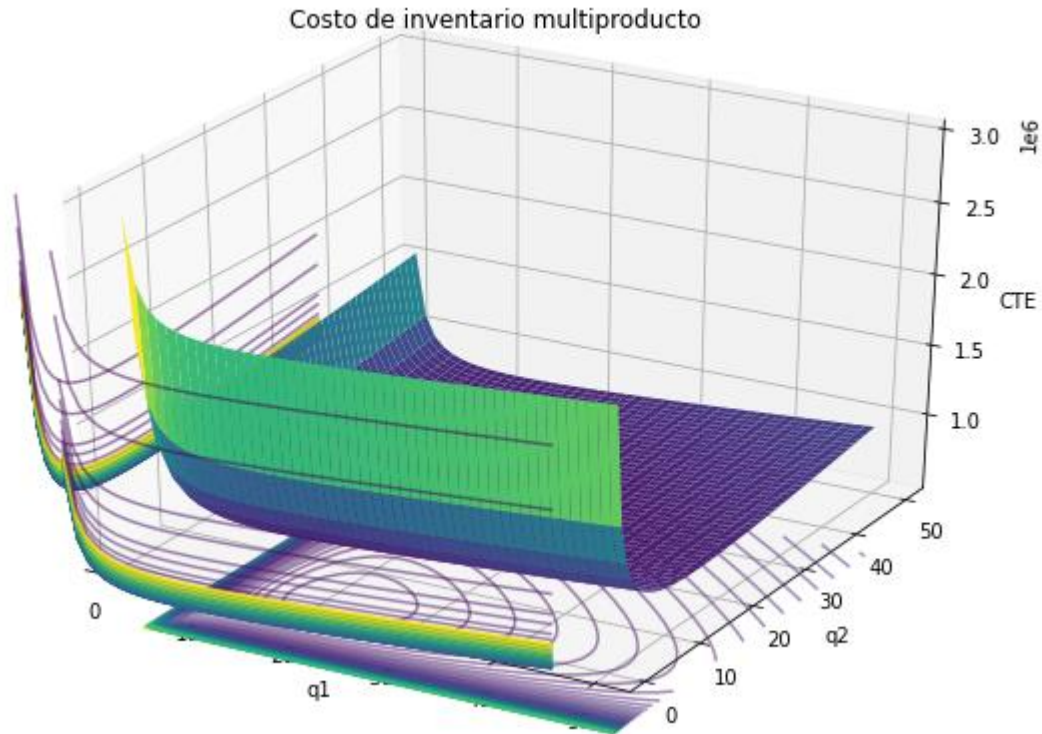
$$q_i^* = \sqrt{\frac{2 * k_i * d_i}{c_u}}$$

Donde  $k_i$  es el costo administrativo de pedido del producto  $i$ ;  $d_i$ , la demanda del producto  $i$  y  $c_u$ , el costo unitario definido más arriba.

Podemos crear la siguiente función de Python que toma los argumentos de costo de orden  $k$ , la demanda  $d$ , el período de análisis  $t$ , y el costo unitario  $c_1$ . El output es el costo óptimo para un producto determinado.

```
1 def calcular_q_opt (k, d, b, i, ba):  
2     return np.sqrt((2 * k * d)/(b * i + ba))
```

# Visualización de Superficie de CTE





# Visualización de Curvas de nivel de CTE

