Víctor Torres Varas 201173076-3

Declaración de Trabajo Individual: Juro o Prometo que la totalidad del trabajo que entregue en esta evaluación corresponde a mi trabajo individual, y es el ~~[illegible]~~ fruto de mi estudio y esfuerzo. (1)

$S'(t) = -\alpha s(t) \cdot i(t)$

$i'(t) = +\alpha s(t) i(t) - \beta i(t)$

$r'(t) = \beta i(t)$

$s(t) = \frac{S(t)}{N}$ $\quad i(t) = \frac{I(t)}{N}$ ; $r(t) = \frac{R(t)}{N}$

$N = 18.000.000$ , $t \in [0, 300]$

$R_0 = \frac{\alpha}{\beta}$ , $R_0 < 1$ Extingue

$R_0 \geq 1$ Propaga

(a) • Inicialmente sólo hay una persona infectada $I(0) = 1$

$I(0) = 1 \rightarrow i(0) = \frac{1}{18.000.000} \approx 0{,}06$

• Sólo hay una persona infectada por lo que el resto es susceptible

$S(0) = 17.999.999$

$s(0) = \frac{17.999.999}{18.000.000} \approx 0.94$

• Las personas que se han recuperado:

$R(0) = 0$ (sólo hay 1 contagiado y el resto sano)

$r(0) = \frac{0}{18.000.000} = 0$

$\beta = 1/14$ $\quad R_0 = [2, 4]$ $\quad \alpha = 1/n_\alpha$ $\quad R_0 = \frac{\alpha}{\beta}$

(b) Determinar rango de $n_\alpha$

$2 \leq R_0 \leq 4$

$2 \leq \frac{\alpha}{\beta} \leq 4$

$2 \leq \frac{1}{\beta \cdot n_\alpha} \leq 4$

$2 \leq \frac{1}{\frac{1}{14} \cdot n_\alpha} \leq 4$

$2 \leq \frac{14}{n_\alpha} \leq 4 \quad / ()^{-1}$

$\frac{1}{2} \leq \frac{n_\alpha}{14} \leq \frac{1}{4} \quad / \cdot 14$

$\boxed{7 \geq n_\alpha \geq 3.5}$

∴ El rango de $n_\alpha$ es $3.5 \geq n_\alpha \geq 7$ días para contagiar a otro.

(c) Sistema Dinámico

$s'(t) = -\alpha s(t) \cdot i(t)$

$i'(t) = \alpha s(t) i(t) - \beta i(t)$

$r'(t) = \beta i(t)$

$i(0) = 0.06$

$s(0) = 0.94$

$r(0) = 0$

Discretización

$s_t = \frac{U_{t+1} - U_t}{1}$ $\quad i_t = \frac{W_{t+1} - W_t}{1}$ $\quad r_t = \frac{X_{t+1} - X_t}{1}$

$U_{t+1} - U_t = -\alpha (U_t \cdot W_t)$

$U_{t+1} = -\alpha (U_t \cdot W_t) + U_t$

$X_{t+1} - X_t = \beta \cdot W_t$

$X_{t+1} = \beta W_t + X_t$

$W_{t+1} - W_t = \alpha (U_t \cdot W_t) - \beta (W_{t+1} - W_t)$

$W_{t+1} - W_t = \alpha (U_t \cdot W_t) - \beta (W_{t+1}) - \beta W_t$

$W_{t+1} + \beta W_{t+1} = \alpha (U_t \cdot W_t) + W_t - \beta W_t$

$W_{t+1}(1 + \beta) = \alpha (U_t W_t) + W_t (1 - \beta)$

$W_{t+1} = \frac{\alpha (U_t W_t) + W_t (1 - \beta)}{(1 - \beta)}$

Nos Queda:

$$U_{t+1} = -\alpha(U_t \cdot W_t) + U_t \qquad U_0 = 6.06$$

$$W_{t+1} = \alpha(U_t W_t) + W_t(1-\beta) \qquad W_0 = 0.94$$

$$X_{t+1} = \beta W_t + X_t \qquad X_0 = 0$$

(2)

Tenemos 3 ecuaciones donde tienen dependencias entre ellas, por lo que se usará DF para cada una, donde en cada iteración tn se llamará al iterador de la siguiente ecuación para el mismo tn. Así se irá anidando para obtener el U, W, X en cada iteración hasta M.

```
def algoritmo(α, β, inf, N, M):
    # se definen U, W, X como las funciones
    U = -α(U·W) + U
    W = ((α(UW) + W(1-β)) / (1-β)
    X = βW + X
    U0, W0, X0  # Valores iniciales
    s, i, r, m = [0, M]  # vectores de dimension M
    # Conocemos las condiciones Iniciales, por lo que llenamos para t=0
    s[0] = U0
    ī[0] = W0
    r[0] = X0
    m[0] = 0.
    for j in range [1, M]
      # resolvemos de forma "anidada".
      s[j] = -α(S_{j-1} ī_{j-1}) + S_{j-1}
      ī[j] = ((α(S_{j-1} ī_{j-1}) + ī_{j-1}(1-β)) / (1-β)
      r[j] = β ī_{j-1} + r_{j-1}
      m[j] = i
    return s, ī, r, m
```