

Actividad #11 (Apocalipsis Zombie)

Instructor: Carlos Lizárraga Celaya

Student: Antonio Cota Rodríguez

Introducción

Como ltima actividad modelaremos diversos ataques de zombies, para el cual utilizaremos un diferentes cdigos en python, los casos son los siguientes :

- Modelo basico de Zombis.
- Modelo con Infeccion Latente.
- Modelo con Cuarentena.
- Modelo con Tratamiento.
- Modelo con Erradicacion Impulsiva.

Según zombipedia, un zombi es una figura de leyenda, parte de la cultura popular. Es una representación de un cadaver que vuelve a revivir producto de haber sido infectado por un virus. Hay una gran variedad de zombis.

El siguiente ejemplo demuestra como resolver un sistema de ecuaciones diferenciales ordinarias usando SciPy, Notar que el orden de la ecuación también puede ser resuelta usando SciPy haciendo una transformacin en el sistema de ecuación de primer orden. Es un ejemplo simple, un sistema de EDO's pueden ser utilizados como un modelo de "invasión zombie", usando las ecuaciones especificados en Munz et al. 2009.

El sistema es dado por :

$$dS/dt = P - BSZ - dS \quad dZ/dt = BSZ + GR - ASZ \quad dR/dt = dS + ASZ - GR$$

Con las siguientes notaciones :

- S : Número de víctimas susceptibles.
- Z : Número de zombies
- R : Número de personas asesinadas
- P : Tasa de nacimientos
- d : Oportunidad de muerte natural
- B : La enfermedad zombie sea transmitida (una persona viva se convierta en zombie)
- G : Una persona muerta sea resucitada en un zombie
- A : El zombie sea toltamente destruido

Esto involucra resolver un sistema de ecuaciones diferenciales de primer orden dadas por : $dy/dt = f(t, y)$. Cuando $y = [S, Z, R]$.

El cdigo que puede ser utilizado es el siguiente :

```
# zombie apocalypse modeling
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint
plt.ion()
plt.rcParams['figure.figsize'] = 10, 8

P = 0 # birth rate
d = 0.0001 # natural death percent (per day)
B = 0.0095 # transmission percent (per day)
G = 0.0001 # resurrect percent (per day)
A = 0.0001 # destroy percent (per day)
```

```

# solve the system dy/dt = f(y, t)
def f(y, t):
    Si = y[0]
    Zi = y[1]
    Ri = y[2]
    # the model equations (see Munz et al. 2009)
    f0 = P - B*Si*Zi - d*Si
    f1 = B*Si*Zi + G*Ri - A*Si*Zi
    f2 = d*Si + A*Si*Zi - G*Ri
    return [f0, f1, f2]

# initial conditions
S0 = 500. # initial population
Z0 = 0 # initial zombie population
R0 = 0.05*S0 # 1% of initial pop is dead
y0 = [S0, Z0, R0]
t = np.linspace(0, 5., 1000) # time grid

# solve the DEs
soln = odeint(f, y0, t)
S = soln[:, 0]
Z = soln[:, 1]
R = soln[:, 2]

# plot results
plt.figure()
plt.plot(t, S, "Darkblue", label='Humanos vivos')
plt.plot(t, Z, "Orange", label='Zombies')
plt.xlabel('Dias despues del brote')
plt.ylabel('Poblacion')
plt.title('Apocalipsis zombie - Con el 5% de la poblacion aniquilada')
plt.legend(loc=0)

```

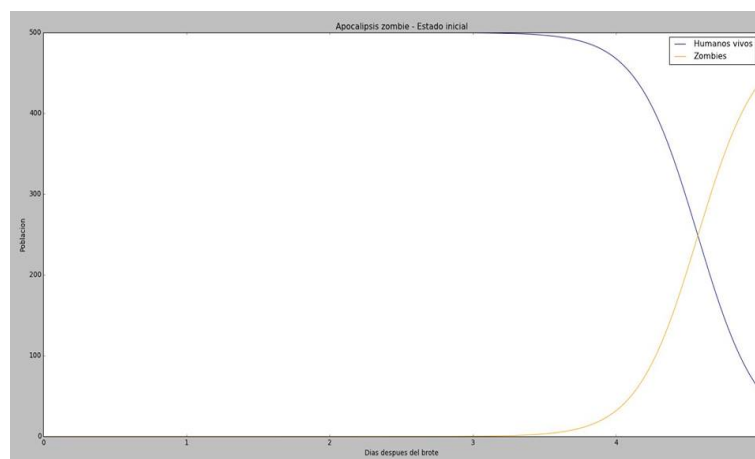


FIGURE 1 –

Este es el modelo base que se nos brindó. Ahora hay que modelar los casos mencionados arriba.

El siguiente caso es para el **Modelo con Infeccion Latente**

```
# zombie apocalypse modeling
```

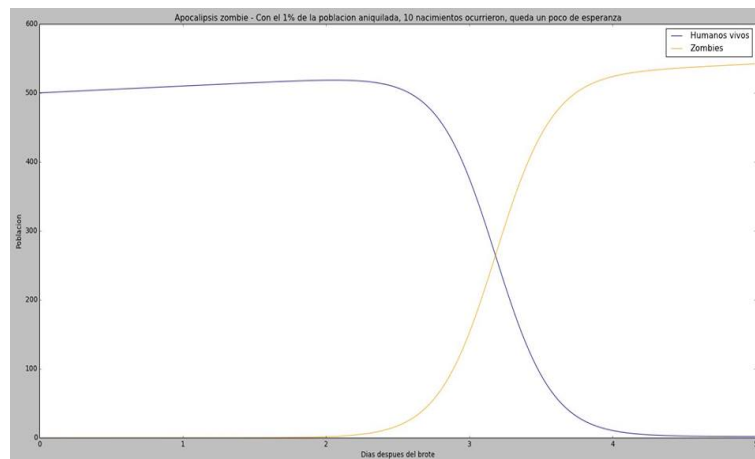


FIGURE 2 –

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint
plt.ion()
plt.rcParams['figure.figsize'] = 10, 8

P = 0 # birth rate
d = 0.0001 # natural death percent (per day)
B = 0.0095 # transmission percent (per day)
G = 0.0001 # resurect percent (per day)
A = 0.0001 # destroy percent (per day)
rho = 0.3

# solve the system dy/dt = f(y, t)
def f(y, t):
    Si = y[0]
    Ii = y[1]
    Zi = y[2]
    Ri = y[3]
    # the model equations (see Munz et al. 2009)
    f0 = P - B*Si*Zi - d*Si
    f1 = B*Si*Zi - rho*Ii - d*Ii
    f2 = rho*Ii + G*Ri - A*Si*Zi
    f3 = d*Si + d*Ii + A*Si*Zi - G*Ri
    return [f0, f1, f2, f3]

# initial conditions
S0 = 500. # initial population
I0 = 1
Z0 = 0 # initial zombie population
R0 = 0 # 1% of initial pop is dead
y0 = [S0, I0, Z0, R0]
t = np.linspace(0, 20, 1000) # time grid

# solve the DEs
soln = odeint(f, y0, t)
S = soln[:, 0]
I = soln[:, 1]
Z = soln[:, 2]
R = soln[:, 3]
```

```
# plot results
plt.figure()
plt.plot(t, S,"Darkblue", label='Humanos vivos')
plt.plot(t, Z,"Orange", label='Zombies')
plt.xlabel('Dias despues del brote')
plt.ylabel('Poblacion')
plt.title('Apocalipsis zombie - Infeccion latente')
plt.legend(loc=0)
```

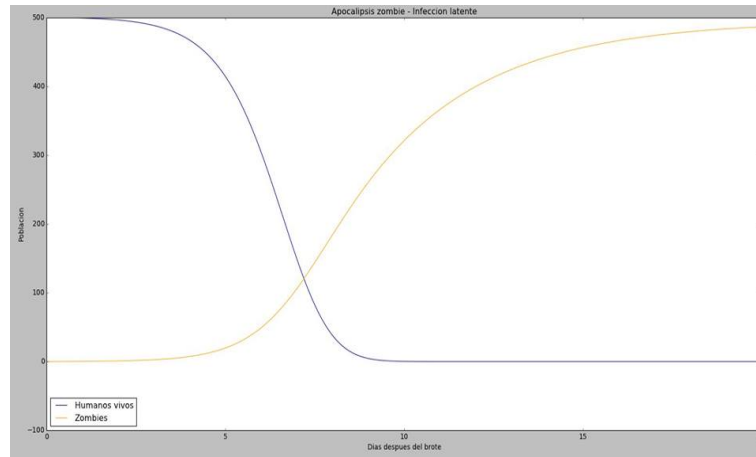


FIGURE 3 –

Ahora para el caso de **Modo Cuarentena** :

```
# zombie apocalypse modeling
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint
plt.ion()
plt.rcParams['figure.figsize'] = 10, 8

P = 0 # birth rate
d = 0.0001 # natural death percent (per day)
B = 0.0095 # transmission percent (per day)
G = 0.0001 # resurect percent (per day)
A = 0.0001 # destroy percent (per day)
rho = 0.3
k = 0.001
sigma = 0.0009
Ga = 0.004

# solve the system dy/dt = f(y, t)
def f(y, t):
    Si = y[0]
    Ii = y[1]
    Zi = y[2]
    Ri = y[3]
    Qi = y[4]
    # the model equations (see Munz et al. 2009)
    f0 = P - B*Si*Zi - d*Si
    f1 = B*Si*Zi - rho*Ii - d*Ii - k*Ii
    f2 = rho*Ii + G*Ri - A*Si*Zi - sigma*Zi
    f3 = d*Si + d*Ii + A*Si*Zi - G*Ri + Ga* Qi
    f4 = k*Ii + sigma*Zi - Ga*Qi
```

```

return [f0, f1, f2, f3, f4]

# initial conditions
S0 = 500. # initial population
I0 = 100
Z0 = 0 # initial zombie population
Q0 = 130
R0 = 0 # 1% of initial pop is dead
y0 = [S0, I0, Z0, R0, Q0]
t = np.linspace(0, 20, 1000) # time grid

# solve the DEs
soln = odeint(f, y0, t)
S = soln[:, 0]
I = soln[:, 1]
Z = soln[:, 2]
R = soln[:, 3]
Q = soln[:, 4]

# plot results
plt.figure()
plt.plot(t, S, "Darkblue", label='Humanos vivos')
plt.plot(t, Z, "Orange", label='Zombies')
plt.xlabel('Dias despues del brote')
plt.ylabel('Poblacion')
plt.title('Apocalipsis zombie - Modo cuarentena')
plt.legend(loc=0)

```

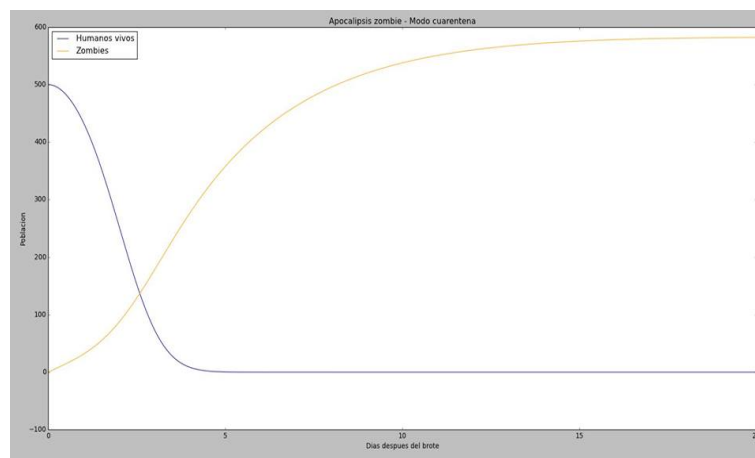


FIGURE 4 –

Y por último el caso de **Modo Tratamiento** :

```

# zombie apocalypse modeling
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint
plt.ion()
plt.rcParams['figure.figsize'] = 10, 8

P = 0 # birth rate
d = 0.0001 # natural death percent (per day)
B = 0.0095 # transmission percent (per day)
G = 0.0001 # resurect percent (per day)

```

```

A = 0.0001 # destroy percent (per day)
rho = 0.3
C = 0.1

# solve the system dy/dt = f(y, t)
def f(y, t):
    Si = y[0]
    Ii = y[1]
    Zi = y[2]
    Ri = y[3]
    # the model equations (see Munz et al. 2009)
    f0 = P - B*Si*Zi - d*Si + C*Zi
    f1 = B*Si*Zi - rho*Ii - d*Ii
    f2 = rho*Ii + G*Ri - A*Si*Zi - C*Zi
    f3 = d*Si + d*Ii + A*Si*Zi - G*Ri
    return [f0, f1, f2, f3]

# initial conditions
S0 = 500. # initial population
I0 = 50
Z0 = 0 # initial zombie population
R0 = 0 # 1% of initial pop is dead
y0 = [S0, I0, Z0, R0]
t = np.linspace(0, 20, 1000) # time grid

# solve the DEs
soln = odeint(f, y0, t)
S = soln[:, 0]
I = soln[:, 1]
Z = soln[:, 2]
R = soln[:, 3]

# plot results
plt.figure()
plt.plot(t, S, "Darkblue", label='Humanos vivos')
plt.plot(t, Z, "Orange", label='Zombies')
plt.xlabel('Dias despues del brote')
plt.ylabel('Poblacion')
plt.title('Apocalipsis zombie - Modo tratamiento')
plt.legend(loc=0)

```

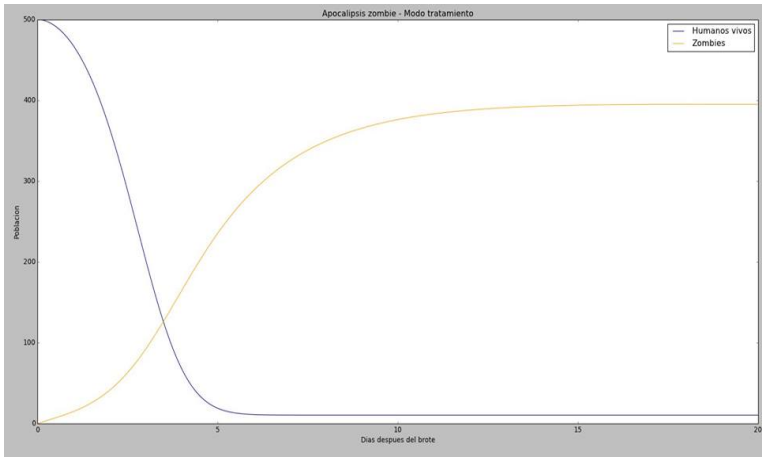


FIGURE 5 –