

Actividad 11: Simulación Zombie

Alejandro Guirado García.

01 de Mayo del 2016

1. Descripción de la actividad



En esta práctica utilizaremos el código que se nos presenta en el artículo, el cuál desarrollaremos para obtener las gráficas en cada caso. Se trataran los diferentes escenarios en caso de una invasión zombi. Un zombi es la representación de un cadáver que de una u otra manera puede resucitar o volver a la vida. Muchas de las diferentes relaciones que se muestran con uno de ellos es una figura legendaria propia del culto vudú. Se trata de un muerto resucitado por medios mágicos por un hechicero para convertirlo en su esclavo. De acuerdo con la creencia, un houngan, bokor o hechicero vudú, sería capaz, mediante un ritual, de resucitar a un muerto,

que quedaría, sin embargo, sometido en adelante a la voluntad de la persona que le devuelve la vida. También, según una creencia popular, se dice que una persona que es mordida por un zombi, se convierte en zombi.

Existe una gran variedad de videojuegos donde interviene ésta temática de zombis debido a los tipos de comportamientos que serán descritos más adelante.

Parámetros que influyen en esta dinámica poblacional:

- Π : Tasa de natalidad.
- β : Tasa de transmisión
- δ : Tasa de muertes naturales
- ζ : Tasa de resusitación
- ρ : Tasa de conversión de I a Z
- α : Tasa de destrucción Z
- σ : Tasa de ingreso de Z a Q
- κ : Tasa de ingreso de I a Q
- γ : Tasa de muertes en Q
- c : Tasa de cura

Clases de individuos a tomar en cuenta.

- Z: Población Zombie
- S: Población susceptible a infectarse o morir, las personas vivas.
- R: Población de individuos removidos o eliminados
- I: Población de infectados pero que no son zombies
- Q: Población en cuarentena

2. Objetivos

Desarrollar los siguientes escenarios posibles.

- Modelo básico de Zombis.
- Modelo con Infección Latente.
- Modelo con Cuarentena.
- Modelo con Tratamiento.

Modelo básico.

Se utilizan 3 clases:

- **S**(Personas susceptibles)
- **R**(Personas Removidas)
- **Z**(Personas susceptibles)

Los susceptibles son las personas que pueden morir por cualquier causa no relacionada a ataque zombi, los marcaremos con el parámetro δ . Sin embargo, pueden convertirse en zombi por el contacto directo con estos, los marcaremos con el parámetro β . Los removidos son las personas que mueren ya sea por ataque zombi o causa natural pero pueden resucitar y convertirse en zombis, los marcaremos con el parámetro ζ . El modelo y las ecuaciones se definen de la siguiente manera: **Código**

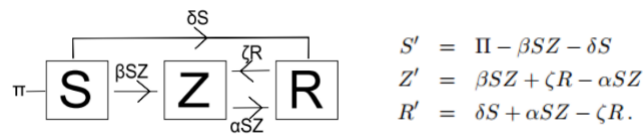


Figura 1: Modelo y ecuaciones, respectivamente.

```

#-*- coding: utf-8 -*-
# APOCALIPSIS ZOMBIE (MODELO BÁSICO)
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint
plt.ion()
plt.rcParams['figure.figsize'] = 10, 8

```

```

P = 0          # taza de nacimientos
d = 0.0001     # porcentaje de muerte natural (por día)
B = 0.0095     # porcentaje de transmisión (por día)
G = 0.0001     # porcentaje de resurrección (por día)
A = 0.0001     # porcentaje de destrucción (por día)

# Resolver el sistema  $dy/dt = f(y, t)$ 
def f(y, t):
    Si = y[0]
    Zi = y[1]
    Ri = y[2]
    # Ecuaciones del modelo
    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]

# condiciones iniciales
S0 = 500.          # población inicial
Z0 = 0             # población zombie inicial
R0 = 100           # población muerta inicial
y0 = [S0, Z0, R0]  # vector condiciones iniciales
t = np.linspace(0, 5., 1000) # tiempo

# resolviendo las ecuaciones diferenciales ordinarias
soln = odeint(f, y0, t)
S = soln[:, 0]
Z = soln[:, 1]
R = soln[:, 2]

# gráfica del modelo básico de apocalipsis zombie
plt.figure()
plt.plot(t, S, "darkcyan", linewidth=2, label='Suceptibles')
plt.plot(t, Z, "darkmagenta", linewidth=2, label='Zombies')
plt.xlabel('Dias a partir del brote de epidemia')
plt.ylabel('Poblacion')
plt.title('Apocalipsis Zombie: modelo basico')
plt.legend(loc=0)

```

Modelo con infección latente.

La diferencia ahora es que existe un periodo de tiempo de aproximadamente 24 horas en el cual el infectado se convierte en zombi. Por lo cual, ahora incluiremos la posibilidad de que un susceptible se infecte antes de ser zombi(I).

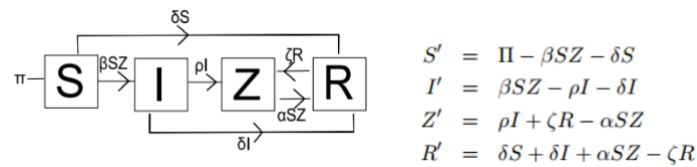


Figura 2: Modelo y ecuaciones, respectivamente.

Código

```

#-*- coding: utf-8 -*-
# MODELO APOCALIPSIS ZOMBIE (INFECCIÓN LATENTE)
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint
plt.ion()
plt.rcParams['figure.figsize'] = 10, 8

P = 0          # taza de nacimiento
d = 0.0001     # porcentaje de muerte (por día)
B = 0.0095     # porcentaje de transmisión (por día)
G = 0.0001     # porcentaje de resurrección (por día)
A = 0.0001     # porcentaje de destrucción (por día)
rho = 0.3      # porcentaje de transformación (por día)

# Resolver el sistema dy/dt = f(y, t)
def f(y, t):
    Si = y[0]
    Ii = y[1]
    Zi = y[2]
    Ri = y[3]

    # Ecuaciones del modelo
    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]

# condiciones iniciales
S0 = 500.                # población inicial
I0 = 1.                  # población infectada inicial
Z0 = 0.                  # población zombie inicial
R0 = 0.                  # población muerta inicial

y0 = [S0, I0, Z0, R0]   # vector condiciones iniciales
t = np.linspace(0, 20, 1000) # tiempo

# resolviendo las ecuaciones diferenciales ordinarias
soln = odeint(f, y0, t)
S = soln[:, 0]
I = soln[:, 1]
Z = soln[:, 2]
R = soln[:, 3]

# gráfica de apocalipsis zombie con infección latente
plt.plot(t, S, "darkcyan",linewidth=2, label='Suceptibles')
plt.plot(t, Z, "darkmagenta",linewidth=2, label='Zombies')
plt.xlabel('Dias a partir del brote de epidemia')
plt.ylabel('Poblacion')
plt.title('Apocalipsis Zombie: infeccion latente')
plt.legend(loc=0)

```

Modelo con cuarentena.

Ahora se utiliza un modelo más real donde una cierta cantidad de individuos infectados se pondrán en cuarentena. La población en cuarentena(Q) solo contiene miembros de la población infectada (I) o de zombis(Z) (con tasas de κ y σ , respectivamente). También si algún individuo en cuarentena intenta escapar será eliminado (γ). Estos individuos muertos entran en la clase de removidos (R), sin embargo después pueden ser reanimados y convertirse en zombis(Z).

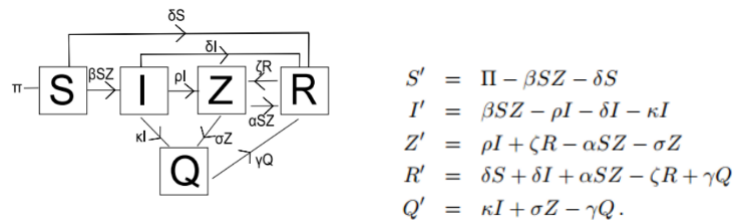


Figura 3: Modelo y ecuaciones, respectivamente.

Código

```

#-*- coding: utf-8 -*-
# MODELO APOCALIPSIS ZOMBIE (CUARENTENA)
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint
plt.ion()
plt.rcParams['figure.figsize'] = 10, 8

P = 0          # tasa de nacimiento
d = 0.0001     # porcentaje de muerte natural (por día)
B = 0.0095     # porcentaje de transmisión (por día)
G = 0.0001     # porcentaje de resurrección (por día)
A = 0.0001     # porcentaje de destrucción (por día)
rho=1          # porcentaje de transformación
k=0.001        # porcentaje de infectados a cuarentena
sigma=0.009    # porcentaje de zombies a cuarentena
Ga=0.004       # porcentaje de muertos en cuarentena

# Resolver el sistema dy/dt = f(y, t)
def f(y, t):

```

```

    Si = y[0]
    Ii = y[1]
    Zi = y[2]
    Ri = y[3]
    Qi = y[4]
# Ecuaciones del modelo
    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]

# condiciones iniciales
S0 = 500.                # población inicial
Z0 = 0.                  # población zombie inicial
R0 = 0.                  # población muerta inicial
I0 = 100.                # población infectada inicial
Q0 = 130.                # población en cuarentena inicial
y0 = [S0, Z0, R0, I0, Q0] # vector condiciones iniciales
t = np.linspace(0, 10., 1000) # tiempo

# resolviendo las ecuaciones diferenciales ordinarias
soln = odeint(f, y0, t)
S = soln[:, 0]
I = soln[:, 1]
Z = soln[:, 2]
R = soln[:, 3]
Q = soln[:, 4]

# gráfica del apocalipsis zombie (cuarentena)
plt.figure()
plt.plot(t, S, "darkcyan",linewidth=2, label='Suceptibles')
plt.plot(t, Z, "darkmagenta",linewidth=2, label='Zombies')
plt.xlabel('Dias a partir del brote de epidemia')
plt.ylabel('Poblacion')
plt.title('Apocalipsis Zombie: brote con cuarentena')
plt.legend(loc=0)

```


Modelo con tratamiento.

Ahora con el desarrollo de la ciencia hemos obtenido una cura para los zombies, con lo cual dichos zombies podrán ser regresados a humanos(S). Es importante aclarar que la cura no proporciona inmunidad por lo tanto, un zombi que ha recibido la cura, puede volver a ser zombi. Debido a la cura, la cuarentena ya no es necesaria, por lo tanto se elimina de las clases a tomar en cuenta.

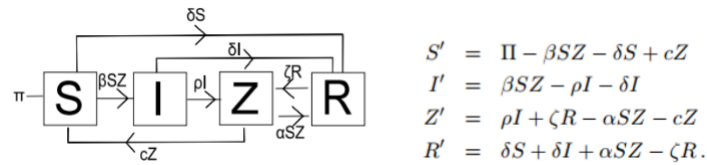


Figura 4: Modelo y ecuaciones, respectivamente.

Código

```

#-*- coding: utf-8 -*-
#MODELO DE APOCALIPSIS ZOMBIE (Tratamiento)
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint
plt.ion()
plt.rcParams['figure.figsize'] = 10, 8

P = 0          # taza de nacimiento
d = 0.0001     # porcentaje de muerte (por día)
B = 0.0095     # porcentaje de transmisión (por día)
G = 0.0001     # porcentaje de resurrección (por día)
A = 0.0001     # porcentaje de destrucción (por día)

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)
rho = 0.5      # porcentaje de transformación (por día)
c = 0.3

```

```

# Resolver el sistema  $dy/dt = f(y, t)$ 
def f(y, t):
    Si = y[0]
    Ii = y[1]
    Zi = y[2]
    Ri = y[3]

# Ecuaciones del modelo
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]

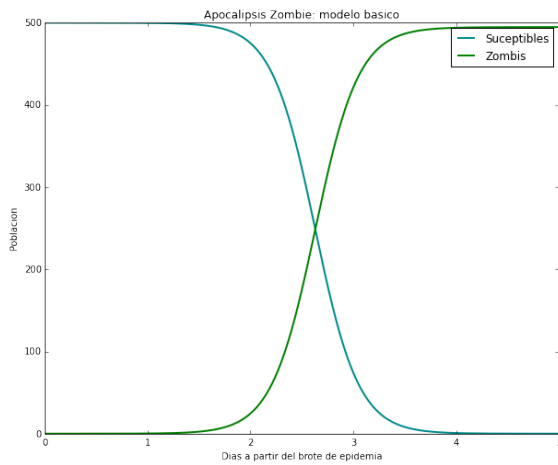
# condiciones iniciales
S0 = 500.          # población inicial
I0 = 5.            # población infectada inicial
Z0 = 0.            # población zombie inicial
R0 = 0.            # población muerta inicial

y0 = [S0, I0, Z0, R0] # initial condition vector
t = np.linspace(0, 10, 1000) # time grid

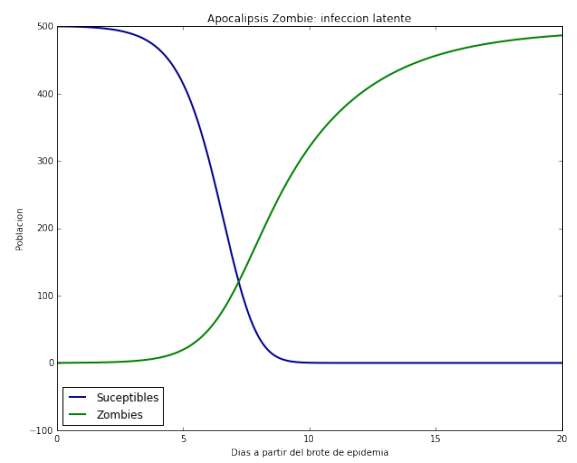
# resolviendo las ecuaciones diferenciales ordinarias
soln = odeint(f, y0, t)
S = soln[:, 0]
I = soln[:, 1]
Z = soln[:, 2]
R = soln[:, 3]

# gráfica de apocalipsis zombie con tratamiento
plt.plot(t, S, "darkcyan", linewidth=2, label='Suceptibles')
plt.plot(t, Z, "darkmagenta", linewidth=2, label='Zombies')
plt.xlabel('Dias a partir del brote de epidemia')
plt.ylabel('Poblacion')

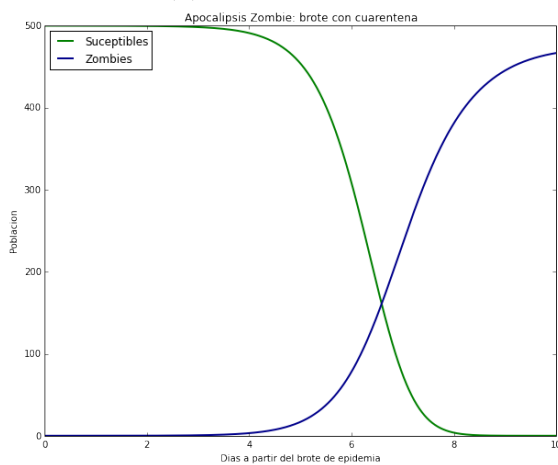
```



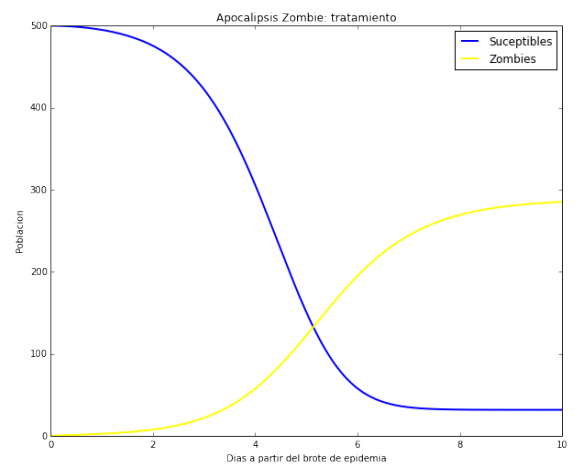
(a) Modelo Básico.



(b) Modelo con ineccción latente.



(c) Modelo cuarentena.



(d) Modelo con cura.

```
plt.title('Apocalipsis Zombie: tratamiento')
plt.legend(loc=0)
```

Gráficas recolectadas.

Referencias

- [1] Lizárraga, C. *Actividad 11 (2016-1)*. Recuperado en Abril de 2016 de <http://computacional1.pbworks.com/w/page/107247876/Actividad%2011%20%282016-1%29>
- [2] *Modeling a Zombie Apocalypse*. Recuperado en Mayo de 2016 de http://scipy-cookbook.readthedocs.io/items/Zombie_Apocalypse_ODEINT.html