"""COMBINED STOCHASTIC"""

"""IMPORT FUNCTIONS"""

import math

import matplotlib.pyplot as plt

import scipy.integrate as spi

from scipy.integrate import odeint

import numpy as np

import random

import pylab as pl

"""---------------"""

"""VARIABLE DECLARATION"""

"""--------------------"""

"""ICU - USA"""

N = 1000

I0, R0 = 1, 0

S0 = N - I0 - R0

b0, g0, r = 0.26, 0.042, 0.021

t = np.linspace(0, 160, 160)

# The SIR model differential equations.

def usa(y, t, N, beta, gamma, r):

 S, I, R = y

 dSdt = -b(t) \* S \* I / N - r \* I

 dIdt = b(t) \* S \* I / N - g(t) \* I- r \* I

 dRdt = g(t) \* I

 return dSdt, dIdt, dRdt

def b(t):

 global bnewusa

 if t == 0:

 bnewusa = b0

 if t%12 == 0:

 bnewusa += bnewusa\*0.2

 return bnewusa

 else:

 return bnewusa

def g(t):

 global gnewusa

 if t == 0:

 gnewusa = g0

 if t%12 == 0:

 gnewusa += 0.0001

 return gnewusa

 else:

 return gnewusa

# Initial conditions vector

y0 = S0, I0, R0

# Integrate the SIR equations over the time grid, t.

ret = odeint(usa, y0, t, args=(N, b, g, r))

S, I, R = ret.T

# Plot the data on three separate curves for S(t), I(t) and R(t)

fig = plt.figure(facecolor='w')

ax = fig.add\_subplot(111, axis\_bgcolor='#dddddd', axisbelow=True)

ax.plot(t, S/1000, 'b', alpha=0.5, lw=2, label='Susceptible')

ax.plot(t, I/1000, 'r', alpha=0.5, lw=2, label='Infected')

ax.plot(t, R/1000, 'g', alpha=0.5, lw=2, label='Dead')

ax.set\_title('USA')

ax.set\_xlabel('Time (Months)')

ax.set\_ylabel('Individuals (1000s)')

ax.set\_ylim(0,1.2)

ax.yaxis.set\_tick\_params(length=0)

ax.xaxis.set\_tick\_params(length=0)

ax.grid(b=True, which='major', c='w', lw=2, ls='-')

legend = ax.legend()

legend.get\_frame().set\_alpha(0.5)

for spine in ('top', 'right', 'bottom', 'left'):

 ax.spines[spine].set\_visible(False)

plt.show()

"""ICU - CANADA"""

N = 1000

I0, R0 = 1, 0

S0 = N - I0 - R0

b0, g0, r = 0.2, .0208, 0.017

t = np.linspace(0, 160, 160)

dB = 0.01

bnew = b0

# The SIR model differential equations.

def can(y, t, N, b, g, r):

 S, I, R = y

 dSdt = -b(t) \* S \* I / N - r \* I

 dIdt = b(t) \* S \* I / N - g(t) \* I - r \* I

 dRdt = g(t) \* I

 return dSdt, dIdt, dRdt

def b(t):

 global bnew

 if t == 0:

 bnew = b0

 if t%12 == 0:

 bnew += 0.1

 return bnew

 else:

 return bnew

def g(t):

 global gnew

 if t == 0:

 gnew = g0

 if t%12 == 0:

 gnew += 0.0001

 return gnew

 else:

 return gnew

# Initial conditions vector

y0 = S0, I0, R0

# Integrate the SIR equations over the time grid, t.

ret = odeint(can, y0, t, args=(N, b, g,r))

S, I, R = ret.T

print ret.T

# Plot the data on three separate curves for S(t), I(t) and R(t)

fig = plt.figure(facecolor='w')

ax = fig.add\_subplot(111, axis\_bgcolor='#dddddd', axisbelow=True)

ax.plot(t, S/1000, 'b', alpha=0.5, lw=2, label='Susceptible')

ax.plot(t, I/1000, 'r', alpha=0.5, lw=2, label='Infected')

ax.plot(t, R/1000, 'g', alpha=0.5, lw=2, label='Dead')

ax.set\_xlabel('Time (Months)')

ax.set\_ylabel('Individuals (1000s)')

ax.set\_title('CANADA')

ax.set\_ylim(0,1.2)

ax.yaxis.set\_tick\_params(length=0)

ax.xaxis.set\_tick\_params(length=0)

ax.grid(b=True, which='major', c='w', lw=2, ls='-')

legend = ax.legend()

legend.get\_frame().set\_alpha(0.5)

for spine in ('top', 'right', 'bottom', 'left'):

 ax.spines[spine].set\_visible(False)

plt.show()