ModelosMunzConParámetrosCorrectos

December 9, 2020

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
```

Resolver ODEs usando el Método de Euler. A continucación se van a reproducir los modelos de *When zombies attack!: Mathematical modelling of an outbreak of zombie infection* de Munz usando los parámetros dados por Witkowski en *Zombie Apocalypse: An Epidemic Model* ya que los dados por Munz no dan los modelos mostrados.

[2]:
```
#Parámetros dados por Munz
a = 0.005  #Zombies move to the removed class upon being ‘defeated’.
b = 0.0095 # Susceptibles can become zombies through transmission via an␣
 ↪encounter with a zombie
d =  0.0 # non-zombierelated death
z = 0.0001  # Humans in the removed class can resurrect and become a zombie
p = 0.005
```

[3]:
```
#Parámetros dados por Witkowski
a = 0.005  #Zombies move to the removed class upon being ‘defeated’.
b = 0.0028 # Susceptibles can become zombies through transmission via an␣
 ↪encounter with a zombie
d =  0.0 # non-zombierelated death
z = 5  # Humans in the removed class can resurrect and become a zombie
p = 5
```

[4]:
```
def euler(y, t, dt, derivadas):
    y_next = y + derivadas(y, t)*dt
    return y_next
```

1 SIR Basic Model

- $Susceptible(S)$
- $Zombie(Z)$
- $Removed(R)$

Only humans can become infected through contact with zombies, and zombies only have a craving for human flesh so we do not consider any other lifeforms in the model.

New zombies can only come from two sources:

- The resurrected from the newly deceased (removed group)
- Susceptibles who have 'lost' an encounter with a zombie

$$\dot{S} = -SZ - S$$
$$\dot{Z} = SZ + R - SZ$$
$$\dot{R} = S + SZ - R$$

[5]:

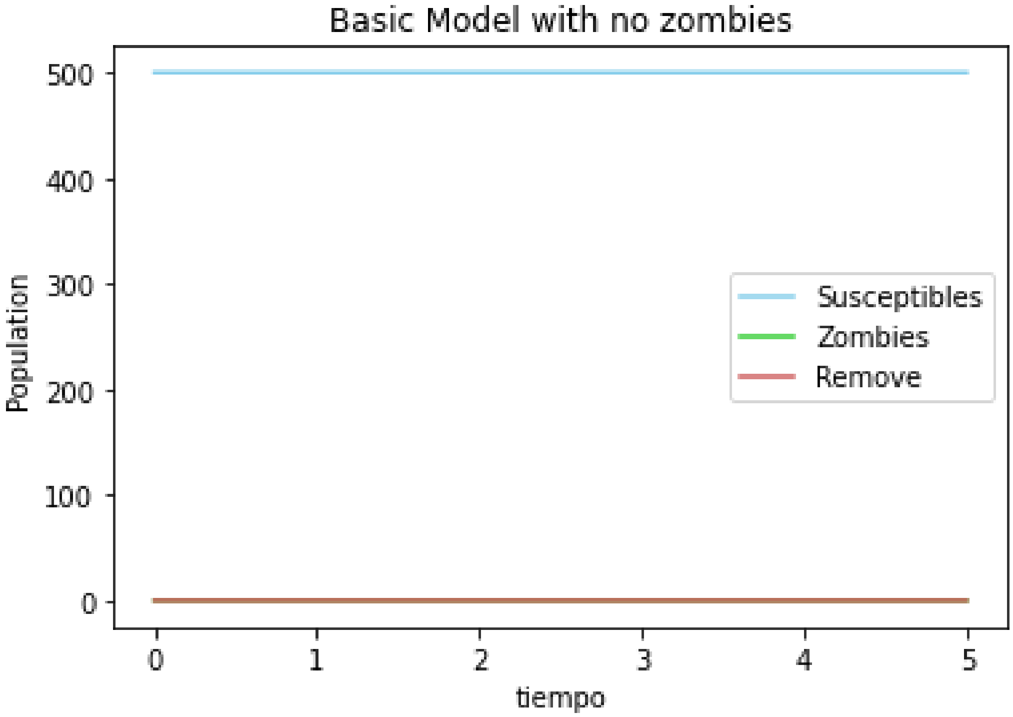
```
def bModel(estado, sistema):
    g0 = -b*estado[0]*estado[1] #assume birth rate = background deathrate
    g1 = b*estado[0]*estado[1] + z*estado[2]  - a*estado[0] * estado[1]
    g2 = d*estado[0] - z*estado[2] + a*estado[0] * estado[1]
    return np.array([g0, g1,g2])
```

1.1 No zombies

[6]:

```
N = 10000 # número de pasos
s0 = 500  # susceptibles inicial
z0 = 0.0  # infectados
r0 = 0.0  # remove
tau = 5.0 # tiempo de la simulación
dt = tau/(float)(N-1)  # tamaño del paso
time = np.linspace(0, tau, N)
y = np.zeros([N,3])

y[0,0] = 500
y[0,1] = 0 #No zombies
y[0,2] = 0

for j in range(N-1):
    y[j+1] = euler(y[j], time[j], dt, bModel)

susdata = [y[j,0] for j in range(N)]
zomdata = [y[j,1] for j in range(N)]
remdata = [y[j,2] for j in range(N)]

plt.plot(time, susdata, label="Susceptibles", color = 'skyblue')
plt.plot(time, zomdata, label="Zombies", color = 'limegreen')
plt.plot(time, remdata, label="Remove", color = 'indianred')
plt.xlabel("tiempo")
plt.ylabel("Population")
plt.legend(loc="best")
plt.title("Basic Model with no zombies")
```

```
[6]: Text(0.5, 1.0, 'Basic Model with no zombies')
```



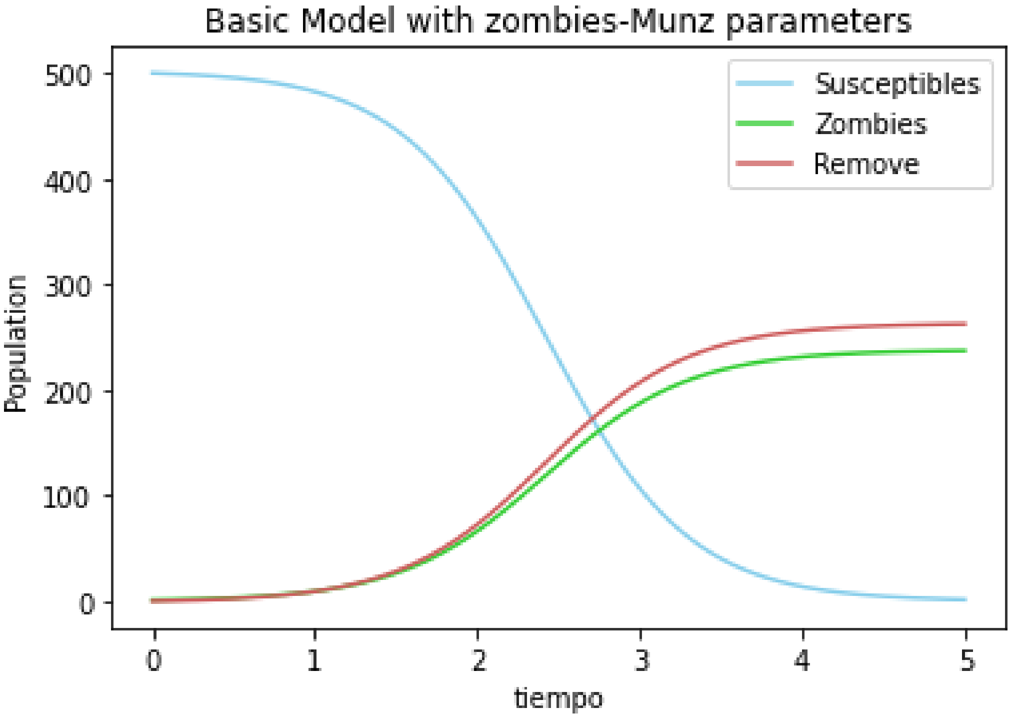
1.2 Con zombies parámetros de Munz

```
[7]: a = 0.005  #Zombies move to the removed class upon being ‘defeated’.
     b = 0.0095 # Susceptibles can become zombies through transmission via an␣
      ↪encounter with a zombie
     d =  0.0001 # non-zombierelated death
     z = 0.0001  # Humans in the removed class can resurrect and become a zombie

     N = 10000 # número de pasos
     s0 = 500  # susceptibles inicial
     z0 = 1.0  # infectados
     r0 = 0.0  # remove
     tau = 5.0 # tiempo de la simulación
     dt = tau/(float)(N-1)  # tamaño del paso
     time = np.linspace(0, tau, N)
     y = np.zeros([N,3])

     y[0,0] = 500
     y[0,1] = 1 #Hay zombies
     y[0,2] = 0
```

```
for j in range(N-1):
    y[j+1] = euler(y[j], time[j], dt, bModel)
susdata = [y[j,0] for j in range(N)]
zomdata = [y[j,1] for j in range(N)]
remdata = [y[j,2] for j in range(N)]

plt.plot(time, susdata, label="Susceptibles", color = 'skyblue')
plt.plot(time, zomdata, label="Zombies", color = 'limegreen')
plt.plot(time, remdata, label="Remove", color = 'indianred')
plt.xlabel("tiempo")
plt.ylabel("Population")
plt.legend(loc="best")
plt.title("Basic Model with zombies-Munz parameters")
```

[7]:

```
Text(0.5, 1.0, 'Basic Model with zombies-Munz parameters')
```



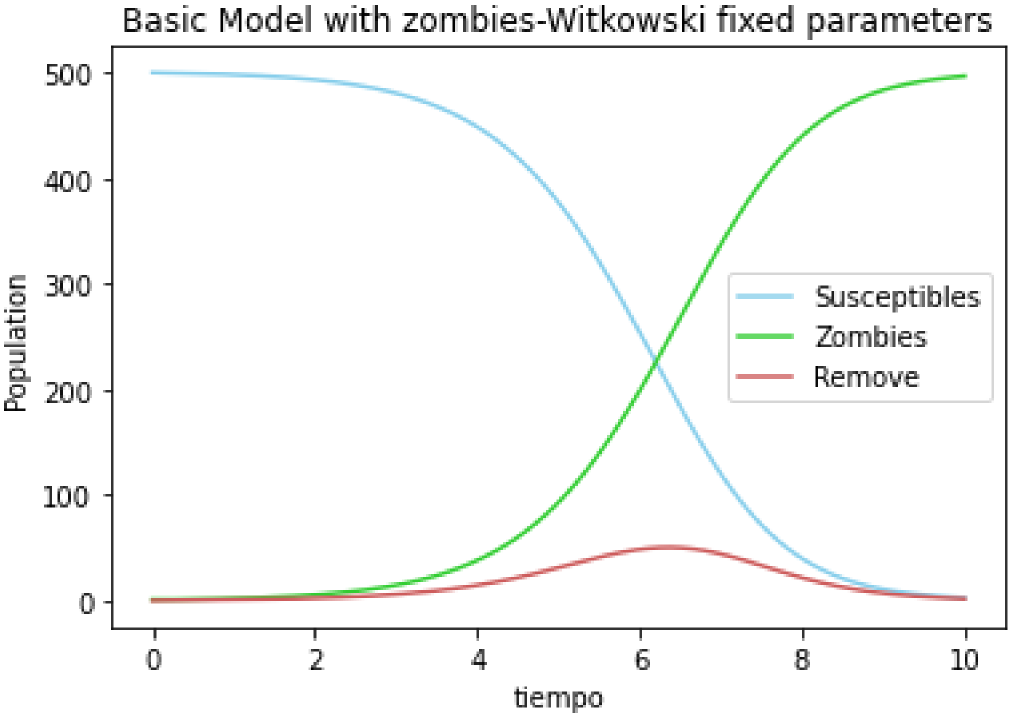
1.3 Con zombies parámetros de Witkowski

[8]:

```
b = 0.0028 # Susceptibles can become zombies through transmission via an␣
↪encounter with a zombie
z = 5  # Humans in the removed class can resurrect and become a zombie
p = 5
```

```
N = 20000 # número de pasos
s0 = 500  # susceptibles inicial
z0 = 1.0  # infectados
r0 = 0.0  # remove
tau = 10.0 # tiempo de la simulación
dt = tau/(float)(N-1)  # tamaño del paso
time = np.linspace(0, tau, N)
y = np.zeros([N,3])

y[0,0] = 500
y[0,1] = 1 #Hay zombies
y[0,2] = 0

for j in range(N-1):
    y[j+1] = euler(y[j], time[j], dt, bModel)
susdata = [y[j,0] for j in range(N)]
zomdata = [y[j,1] for j in range(N)]
remdata = [y[j,2] for j in range(N)]

plt.plot(time, susdata, label="Susceptibles", color = 'skyblue')
plt.plot(time, zomdata, label="Zombies", color = 'limegreen')
plt.plot(time, remdata, label="Remove", color = 'indianred')
plt.xlabel("tiempo")
plt.ylabel("Population")
plt.legend(loc="best")
plt.title("Basic Model with zombies-Witkowski fixed parameters")
```

```
[8]: Text(0.5, 1.0, 'Basic Model with zombies-Witkowski fixed parameters')
```



2 The model with latent infection

• Susceptibles first move to an infected class once infected and remain there for some period of time.

• Infected individuals can still die a 'natural' death before becoming a zombie; otherwise, they become a zombie.

$$\dot{S} = -SZ - S$$
$$\dot{I} = SZ - I - I$$
$$\dot{Z} = I + R - SZ$$
$$\dot{R} = S + I + SZ - R$$

Short timescale and hence $\Pi = \quad = 0$

[9]:
```
def infModel(estado, sistema):
    g0 = -b*estado[0]*estado[2] #assume birth rate = background deathrate
    g1 =  b*estado[0]*estado[2] - p*estado[1]
    g2 =  p*estado[1]+ z*estado[3]  - a*estado[0] * estado[2]
    g3 = - z*estado[3] + a*estado[0] * estado[2]
    return np.array([g0, g1,g2,g3])
```

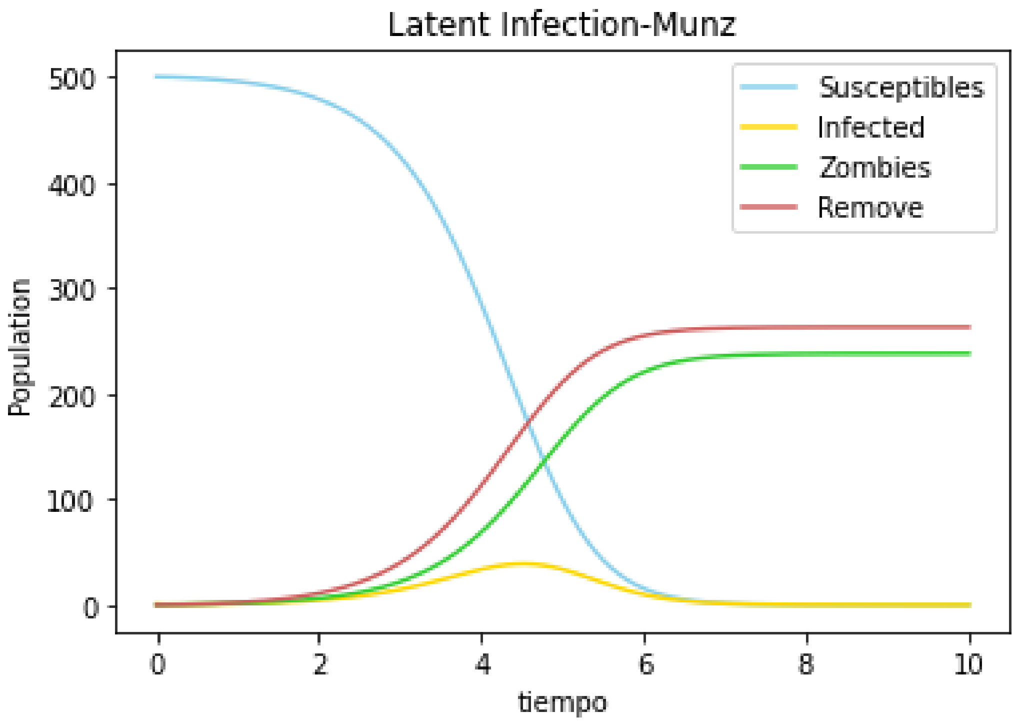
2.1 Infección latente con zombies parámetros de Munz

[10]:

```
a = 0.005  #Zombies move to the removed class upon being ‘defeated’.
b = 0.0095 # Susceptibles can become zombies through transmission via an␣
 ↪encounter with a zombie
d =  0.0001 # non-zombierelated death
z = 0.0001  # Humans in the removed class can resurrect and become a zombie
p = 5 #p = 5 y se ve la gráfica bien
N = 20000 # número de pasos
s0 = 500  # susceptibles inicial
z0 = 1.0  # infectados
r0 = 0.0  # remove
tau = 10.0 # tiempo de la simulación
dt = tau/(float)(N-1)  # tamaño del paso
time = np.linspace(0, tau, N)
y = np.zeros([N,4])

y[0,0] = 500
y[0,1] = 1 #Hay zombies
y[0,2] = 0
y[0,3] = 0

for j in range(N-1):
    y[j+1] = euler(y[j], time[j], dt, infModel)
susdata = [y[j,0] for j in range(N)]
infdata = [y[j,1] for j in range(N)]
zomdata = [y[j,2] for j in range(N)]
remdata = [y[j,3] for j in range(N)]

plt.plot(time, susdata, label="Susceptibles", color = 'skyblue')
plt.plot(time, infdata, label="Infected", color = 'gold')
plt.plot(time, zomdata, label="Zombies", color = 'limegreen')
plt.plot(time, remdata, label="Remove", color = 'indianred')
plt.xlabel("tiempo")
plt.ylabel("Population")
plt.legend(loc="best")
plt.title("Latent Infection-Munz")
```

[10]:

```
Text(0.5, 1.0, 'Latent Infection-Munz')
```



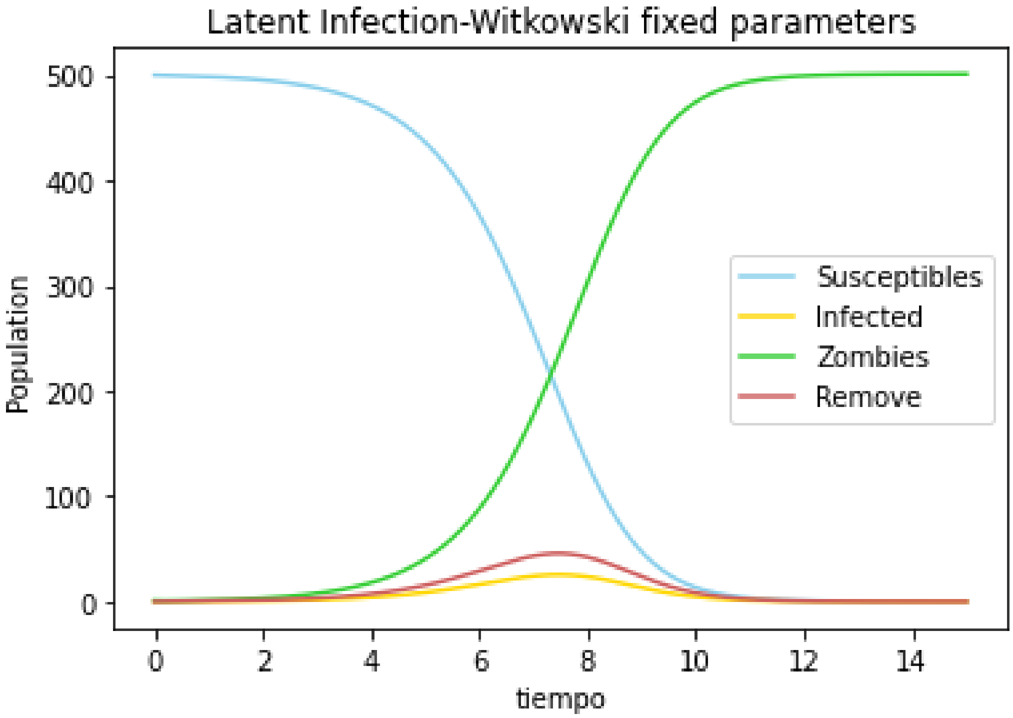
2.2 Infección latente con zombies parámetros de Witkowski

[11]:

```
b =  0.0028 #once infected what for some period of time become a zombie
p = 5
z = 5

N = 30000
tau = 15.0 # tiempo de la simulación
dt = tau/(float)(N-1)  # tamaño del paso
time = np.linspace(0, tau, N)
y = np.zeros([N,4])
y[0,0] = 500
y[0,1] = 0
y[0,2] = 1
y[0,3] = 0

for j in range(N-1):
    y[j+1] = euler(y[j], time[j], dt, infModel)

susdata = [y[j,0] for j in range(N)]
infdata = [y[j,1] for j in range(N)]
zomdata = [y[j,2] for j in range(N)]
remdata = [y[j,3] for j in range(N)]
```

```
plt.plot(time, susdata, label="Susceptibles", color = 'skyblue')
plt.plot(time, infdata, label="Infected", color = 'gold')
plt.plot(time, zomdata, label="Zombies", color = 'limegreen')
plt.plot(time, remdata, label="Remove", color = 'indianred')
plt.xlabel("tiempo")
plt.ylabel("Population")
plt.legend(loc="best")
plt.title("Latent Infection-Witkowski fixed parameters")
```

```
[11]: Text(0.5, 1.0, 'Latent Infection-Witkowski fixed parameters')
```



3 The model with quarantine

- The quarantined area only contains members of the infected or zombie populations (entering at rates and , rspectively).

- There is a chance some members will try to escape, but any that tried to would be killed before finding their 'freedom' (parameter).

- These killed individuals enter the removed class and may later become reanimated as 'free' zombies.

$$\dot{S} = -SZ - S$$

$$\dot{I} = SZ - I - I - I$$
$$\dot{Z} = I + R - SZ - Z$$
$$\dot{R} = S + I + SZ - R + Q$$
$$\dot{Q} = I + Z - Q$$

[12]:

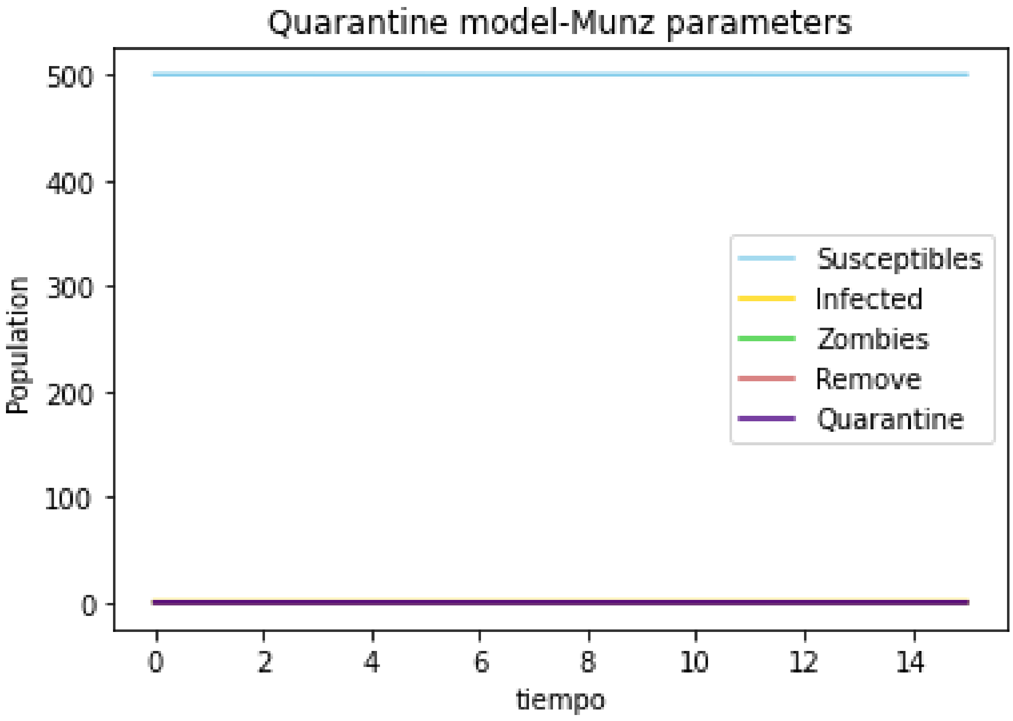
```
k = 0.003
sig = 0.001
gam = 0.001
def quaModel(estado, sistema):
    g0 = -b*estado[0]*estado[2]
    g1 = b*estado[0]*estado[2] - p*estado[1] - k *estado[1]
    g2 = (p)*estado[1] + z*estado[3]  - a*estado[0] * estado[2] -sig*estado[2]
    g3 = - z*estado[3] + a*estado[0] * estado[2] + gam * estado[4]
    g4 = k * estado[1] + sig * estado[2]- gam*estado[4]

    return np.array([g0, g1,g2,g3,g4])
```

3.1 Cuarentena con parámetros de Munz

[13]:

```
b = 0.0095 # Susceptibles can become zombies through transmission via an␣
 ↪encounter with a zombie
d =  0.0001 # non-zombierelated death
z = 0.0001
p = .005
N = 30000
tau = 15.0 # tiempo de la simulación
time = np.linspace(0, tau, N)
dt = tau/(float)(N-1)  # tamaño del paso
y = np.zeros([N,5])
y[0,0] = 500
y[0,1] = 1
y[0,2] = 0
y[0,3] = 0
y[0,4] = 0

for j in range(N-1):
    y[j+1] = euler(y[j], time[j], dt, quaModel)

susdata = [y[j,0] for j in range(N)]
infdata = [y[j,1] for j in range(N)]
zomdata = [y[j,2] for j in range(N)]
remdata = [y[j,3] for j in range(N)]
quadata = [y[j,4] for i in range(N)]

plt.plot(time, susdata, label="Susceptibles", color = 'skyblue')
plt.plot(time, infdata, label="Infected", color = 'gold')
plt.plot(time, zomdata, label="Zombies", color = 'limegreen')
plt.plot(time, remdata, label="Remove", color = 'indianred')
plt.plot(time, quadata, label="Quarantine", color = 'indigo')
plt.xlabel("tiempo")
plt.ylabel("Population")
plt.legend(loc="best")
plt.title("Quarantine model-Munz parameters")
```

```
[13]: Text(0.5, 1.0, 'Quarantine model-Munz parameters')
```

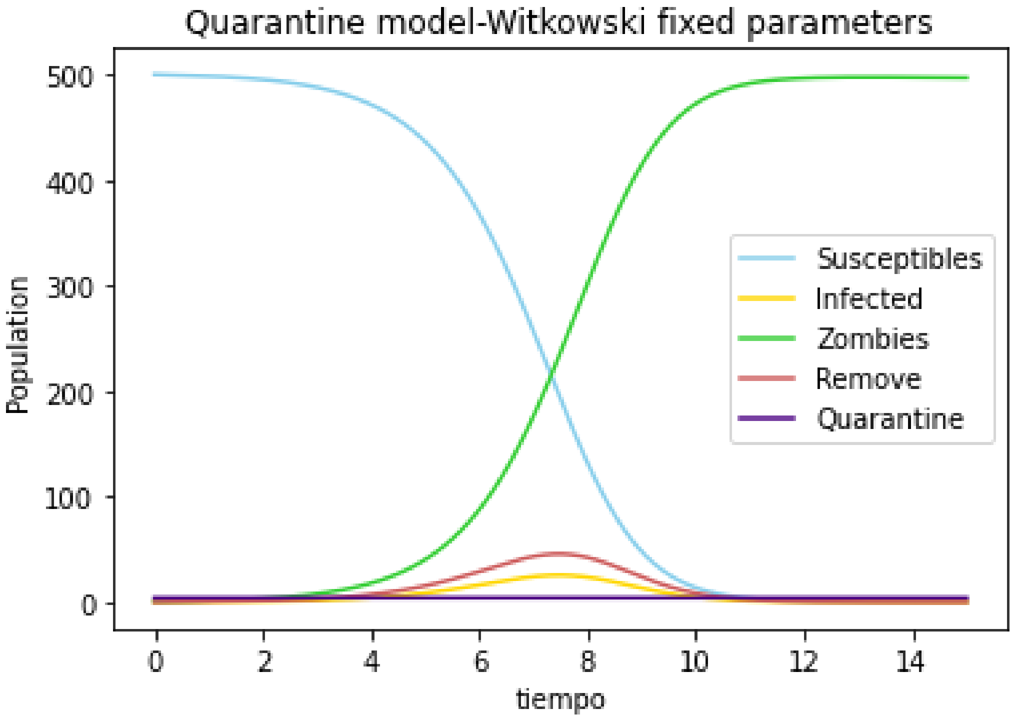


3.2 Cuarentena con parámetros de Witkowski

```
[14]: b =  0.0028 #once infected what for some period of time become a zombie
p = 5
z = 5

N = 30000
tau = 15.0 # tiempo de la simulación
time = np.linspace(0, tau, N)
dt = tau/(float)(N-1)  # tamaño del paso
y = np.zeros([N,5])
y[0,0] = 500
y[0,1] = 0
```

```
y[0,2] = 1
y[0,3] = 0
y[0,4] = 0

for j in range(N-1):
    y[j+1] = euler(y[j], time[j], dt, quaModel)

susdata = [y[j,0] for j in range(N)]
infdata = [y[j,1] for j in range(N)]
zomdata = [y[j,2] for j in range(N)]
remdata = [y[j,3] for j in range(N)]
quadata = [y[j,4] for i in range(N)]

plt.plot(time, susdata, label="Susceptibles", color = 'skyblue')
plt.plot(time, infdata, label="Infected", color = 'gold')
plt.plot(time, zomdata, label="Zombies", color = 'limegreen')
plt.plot(time, remdata, label="Remove", color = 'indianred')
plt.plot(time, quadata, label="Quarantine", color = 'indigo')
plt.xlabel("tiempo")
plt.ylabel("Population")
plt.legend(loc="best")
plt.title("Quarantine model-Witkowski fixed parameters")
```

```
[14]: Text(0.5, 1.0, 'Quarantine model-Witkowski fixed parameters')
```



4 A model with treatment

Things that need to be considered now include:

- Since we have treatment, we no longer need the quarantine.

- The cure will allow zombies to return to their original human form regardless of how they became zombies in the first place.

- Any cured zombies become susceptible again; the cure does not provide immunity.

$$
\begin{aligned}
\dot{S} &= -SZ - S + cZ \\
\dot{I} &= SZ - I - I \\
\dot{Z} &= I + R - SZ - cZ \\
\dot{R} &= S + I + SZ - R
\end{aligned}
$$

[15]:

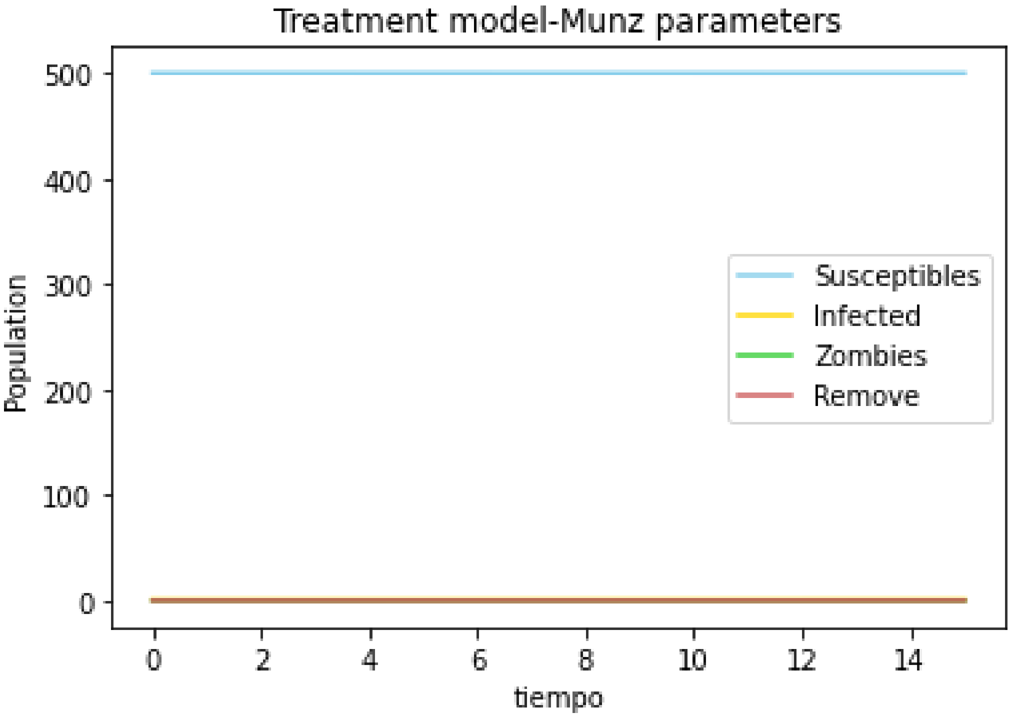
```
c = 0.3 #cure

def treatModel(estado, sistema):
    g0 = -b*estado[0]*estado[2]  + c * estado[2]
    g1 = b*estado[0]*estado[2] - p*estado[1]
    g2 = (p)*estado[1] + z*estado[3]  - a*estado[0] * estado[2] - c * estado[2]
    g3 = - z*estado[3] + a*estado[0] * estado[2]

    return np.array([g0, g1,g2,g3])
```

4.1 Tratamiento con parámetros de Munz

[16]:

```
b = 0.0095 # Susceptibles can become zombies through transmission via an␣
 ↪encounter with a zombie
d =  0.0001 # non-zombierelated death
z = 0.0001
p = .005

N = 30000
tau = 15.0 # tiempo de la simulación
time = np.linspace(0, tau, N)
dt = tau/(float)(N-1)  # tamaño del paso
y = np.zeros([N,4])
y[0,0] = s0
y[0,1] = 1
y[0,2] = 0
y[0,3] = r0

for j in range(N-1):
    y[j+1] = euler(y[j], time[j], dt, treatModel)

susdata = [y[j,0] for j in range(N)]
infdata = [y[j,1] for j in range(N)]
zomdata = [y[j,2] for j in range(N)]
remdata = [y[j,3] for j in range(N)]

plt.plot(time, susdata, label="Susceptibles", color = 'skyblue')
plt.plot(time, infdata, label="Infected", color = 'gold')
plt.plot(time, zomdata, label="Zombies", color = 'limegreen')
plt.plot(time, remdata, label="Remove", color = 'indianred')
plt.xlabel("tiempo")
plt.ylabel("Population")
plt.legend(loc="best")
plt.title("Treatment model-Munz parameters")
```

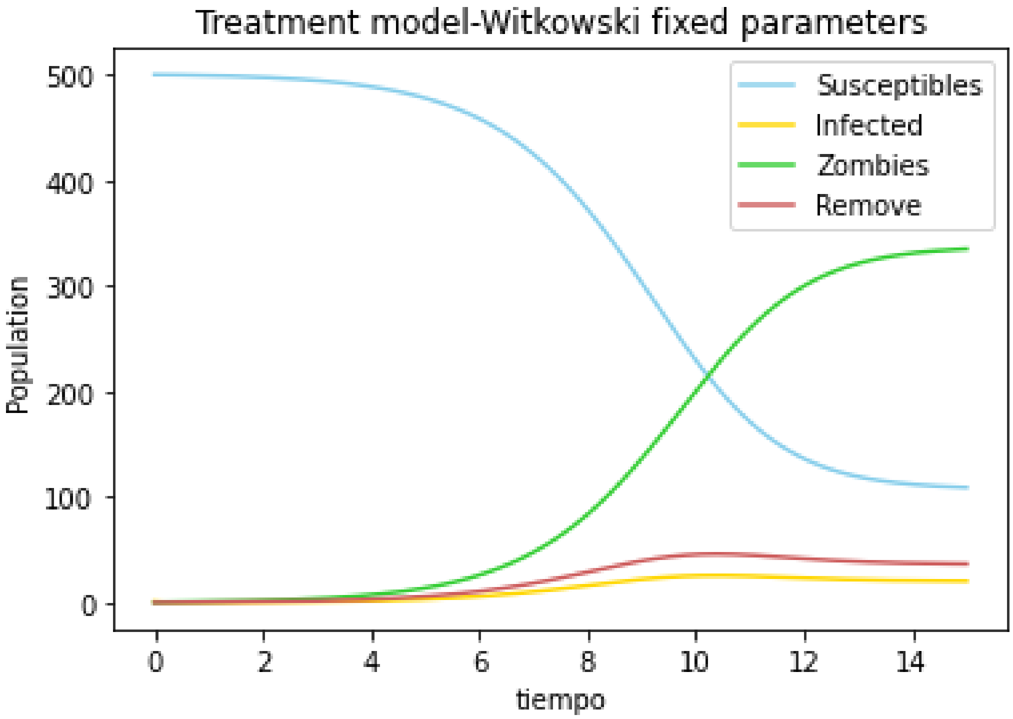
[16]:
```
Text(0.5, 1.0, 'Treatment model-Munz parameters')
```



[17]:
```
b =  0.0028 #once infected what for some period of time become a zombie
p = 5
z = 5

N = 30000
```

```
tau = 15.0 # tiempo de la simulación
time = np.linspace(0, tau, N)
dt = tau/(float)(N-1)  # tamaño del paso
y = np.zeros([N,4])
y[0,0] = s0
y[0,1] = 1
y[0,2] = 0
y[0,3] = r0

for j in range(N-1):
    y[j+1] = euler(y[j], time[j], dt, treatModel)

susdata = [y[j,0] for j in range(N)]
infdata = [y[j,1] for j in range(N)]
zomdata = [y[j,2] for j in range(N)]
remdata = [y[j,3] for j in range(N)]

plt.plot(time, susdata, label="Susceptibles", color = 'skyblue')
plt.plot(time, infdata, label="Infected", color = 'gold')
plt.plot(time, zomdata, label="Zombies", color = 'limegreen')
plt.plot(time, remdata, label="Remove", color = 'indianred')
plt.xlabel("tiempo")
plt.ylabel("Population")
plt.legend(loc="best")
plt.title("Treatment model-Witkowski fixed parameters")
```

```
[17]: Text(0.5, 1.0, 'Treatment model-Witkowski fixed parameters')
```



5 Impulsive Eradication

We attempted to control the zombie population by strategically destroying them at such times that our resources permit.

$$\dot{S} = -SZ - S$$

$$\dot{Z} = SZ + R - SZ$$

$$\dot{R} = S + SZ - R$$

$$Z = -knZ$$

[18]:
```
k = 0.25
a = 0.0075
b = 0.0055
z = 0.09
d = 0.0001
```

[19]:
```
def impEraModel(estado, sistema):
    g0 = -b*estado[0]*estado[1] #assume birth rate = background deathrate
    g1 = b*estado[0]*estado[1] + z*estado[2]  - a*estado[0] * estado[1]
    g2 = d*estado[0] - z*estado[2] + a*estado[0] * estado[1]
    return np.array([g0, g1,g2])
```

[20]:

```python
from ipywidgets import interact, fixed, widgets
```

[21]:

```python
class ModeloZombie:

    def __init__(self, a,b,z, p, N = 20000, s0 = 500, z0 = 1, r0 = 0):
        self.a = a
        self.b = b
        self.z = z
        self.p = p
        self.N = N
        self.s0 = s0
        self.z0 = z0
        self.r0 = r0
        self.tau = (10/20000) * N
        self.dt = self.tau//(float)(N-1)
        self.time = np.linspace(0, self.tau, N)
        self.y = np.zeros([N,4])
        self.y[0,0] = s0 # susceptibles inicial
        self.y[0,1] = z0 #Hay zombies
        self.y[0,2] = r0 # remove

    def bModel(self,estado, sistema):
        g0 = -self.b*estado[0]*estado[1] #assume birth rate = background deathrate
        g1 = self.b*estado[0]*estado[1] + self.z*estado[2]  - self.a*estado[0] * estado[1]
        g2 =  - self.z*estado[2] + self.a*estado[0] * estado[1]
        return np.array([g0, g1,g2])

    def infModel(self,estado, sistema):
        g0 = -self.b*estado[0]*estado[2] #assume birth rate = background deathrate
        g1 = self.b*estado[0]*estado[2] - self.p*estado[1]
        g2 = self.p*estado[1]+ self.z*estado[3]  - self.a*estado[0] * estado[2]
        g3 = - self.z*estado[3] + self.a*estado[0] * estado[2]
        return np.array([g0, g1,g2,g3])


    def quaModel(self,estado, sistema):
        k = .02
        sig = .03
        gam = .001
        g0 = -self.b*estado[0]*estado[2]
        g1 = self.b*estado[0]*estado[2] - self.p*estado[1] - k *estado[1]
        g2 = (self.p)*estado[1] + self.z*estado[3]  - self.a*estado[0] * estado[2] -sig*estado[2]
        g3 = - self.z*estado[3] + self.a*estado[0] * estado[2] + gam * estado[4]
        g4 = k * estado[1] + sig * estado[2]- gam*estado[4]

        return np.array([g0, g1,g2,g3,g4])

    def euler(self,y, t, dt, derivadas):
        y_next = y + derivadas(y, t)*dt
        return y_next

    def susceptibles(self):
        return [self.y[j,0] for j in range(N)]

    def zombies(self):
        return [self.y[j,1] for j in range(N)]

    def remove(self):
        return [self.y[j,2] for j in range(N)]

    def plotBasic(self,al,be,zz,pp):
        self.a = al
        self.b = be
        self.z = zz
        for j in range(N-1):
            y[j+1] = self.euler(y[j], time[j], dt, self.bModel)
        susdata = [y[j,0] for j in range(N)]
        zomdata = [y[j,1] for j in range(N)]
        remdata = [y[j,2] for j in range(N)]

        plt.plot(time, susdata, label="Susceptibles" , color = 'skyblue')
        plt.plot(time, zomdata, label="Zombies", color = 'limegreen')
        plt.plot(time, remdata, label="Remove", color = 'indianred')
        plt.xlabel("tiempo")
        plt.ylabel("Population")
        plt.legend(loc="best")
        plt.title("Basic Model with zombies")

    def plotLatent(self, al, be,zz,pp):
        self.a = al
        self.b = be
        self.z = zz
        self.p = pp
        y = np.zeros([N,4])
        y[0,0] = 500
        y[0,1] = 0
        y[0,2] = 1
        y[0,3] = 0

        for j in range(N-1):
            y[j+1] = euler(y[j], time[j], dt, self.infModel)

        susdata = [y[j,0] for j in range(N)]
        infdata = [y[j,1] for j in range(N)]
        zomdata = [y[j,2] for j in range(N)]
        remdata = [y[j,3] for j in range(N)]

        plt.plot(time, susdata, label="Susceptibles", color = 'skyblue')
        plt.plot(time, infdata, label="Infected", color = 'gold')
        plt.plot(time, zomdata, label="Zombies", color = 'limegreen')
        plt.plot(time, remdata, label="Remove", color = 'indianred')
        plt.xlabel("tiempo")
        plt.ylabel("Population")
        plt.legend(loc="best")
        plt.title("Latent Infection")
```

[22]:

```
z = ModeloZombie(0.005, 0.0095, 0.0001, 0.005)
```

[23]:

```
interact(z.plotLatent, al =widgets.FloatSlider(value=.005, min=0, max=.01,
                    step=0.001, readout_format='.4f'), be =widgets.
 ↪FloatSlider(value=.0095, min=0, max=.015,
                    step=0.0001, readout_format='.4f'), zz=widgets.
 ↪FloatSlider(value=.0001, min=0, max=5.01,
                    step=0.001, readout_format='.4f'), pp  =widgets.
 ↪FloatSlider(value=.005, min=0, max=5.01,
                    step=0.001, readout_format='1.4f'));
```

```
interactive(children=(FloatSlider(value=0.005, description='al', max=0.01, readout_format='.4f
```

6 Diferencias entre epidemia normal y epidemia zombie

Una epidemia es el **aumento extraordinario del número de casos de una enfermedad infecciosa** que ya existe en una región o población determinada. Puede referirse también a la aparición de un número importante de casos de una enfermedad infecciosa en una región o población habitualmente libre de la enfermedad.

Un **zombie moderno** es un cadáver reanimado que no siente dolor y que tiene un gran apetito por la carne humana. Su objetivo es matar, comer o infectar a las personas.

Las carácterísticas a considerar en un modelo de epidemia normal básico son los susceptibles (S), los infectados (I), recuperados o fallecidos (R). En los recuperados normalmente se toma que el individuo generó inmunidad a la infección.

Munz nos muestra diferentes modelos de epidemia, donde se llegan a contar siempre los suceptibles (S), los zombies (Z) y fallecidos (R). Además, mete a los infectados (I) porque es posible que tarde determinado tiempo en convertirse en zombie. También pone la opción de quarentena (Q) donde los zombies en quarentena no pueden infectar a las personas.

Una de las principales diferencias en estos modelos es la diferencia entre infectado y zombie que se da en el modelo de epidemia zombie. El infectado ya tiene el virus, pero no es capaz de contagiar a otros y tiene que tomarse en cuenta el tiempo que tarda en convertirse en zombie. Y en el modelo de epidemia normal alguien infectado puede contagiar a los demás.

Otra diferencia entre una epidemia zombie y una epidemia normal es que en la epidemia zombie si alguien falleció puede tiene la posibilidad de convertirse en zombie, mientras que en una epidemia normal no se puede regresar a la vida por ningún método.

Otra diferencia es que los zombies contagian mediante mordidas a individuos, y su objetivo no solo es infectar, también es matar. Mientras que en una epidemia normal, los infectados contagias a las demás personas a través del aire, toser o estornudar, tocar a alguien enfermo o un objeto contaminado.

6.1 Se han hecho muchas críticas al artículo (e.g. ***Zombie Apocalypse: An Epidemic Model***) indicando que hay errores y que las gráficas no corresponden ¿Es cierto? Si es afirmativa la respuesta, ¿Cuáles son esos errores?

Es cierto, en las gráficas de arriba se muestran varias gráficas modelando en todas los modelos propuestos por Munz. Por cada modelo se ven 2 gráficas, una con los parámetros de Munz y otra con los parámetros de Witkowski propuestos en *Zombie Apocalypse: An Epidemic Model*.

Los errores que se resaltan en *Zombie Apocalypse: An Epidemic Model* es que los modelados de Munz no coinciden con los parámetros que afirma usar para conseguirlas. En el mismo documento Witkowski nos propone nuevos de los parámetros para ilustar las gráficas de Munz.

$$Parmetros : Munz \Rightarrow Witkowski$$

$$: 0.0095 \Rightarrow .0028$$

$$: 0.0001 \Rightarrow 5$$

$$: 0.005 \Rightarrow 5$$

Donde es la probabilidad de que un susceptible se convierta en zombie mediante un encuentro con un zomie. Donde es la probabilidad de que un fallecido reviva y se convierta en zombie. Donde es la probabilidad de que un infectado se convierta en zombie (es el tiempo que tarda pero se maneja como probabilidad).

```
[ ]:
```