In [1]:
```python
import numpy as np
from scipy.integrate import odeint
import matplotlib.pyplot as plt
```

Model I:

$$s(t) + e(t) + i(t) + r(t) + p(t) = 1$$

Model I:

$$\frac{ds(t)}{dt} = -\alpha_e s(t)e(t) - \alpha_i s(t)i(t) + \gamma r(t)$$

$$\frac{de(t)}{dt} = \alpha_e s(t)e(t) + \alpha_i s(t)i(t) - \kappa e(t) - \rho e(t)$$

$$\frac{di(t)}{dt} = \kappa e(t) - \beta i(t) - \mu i(t)$$

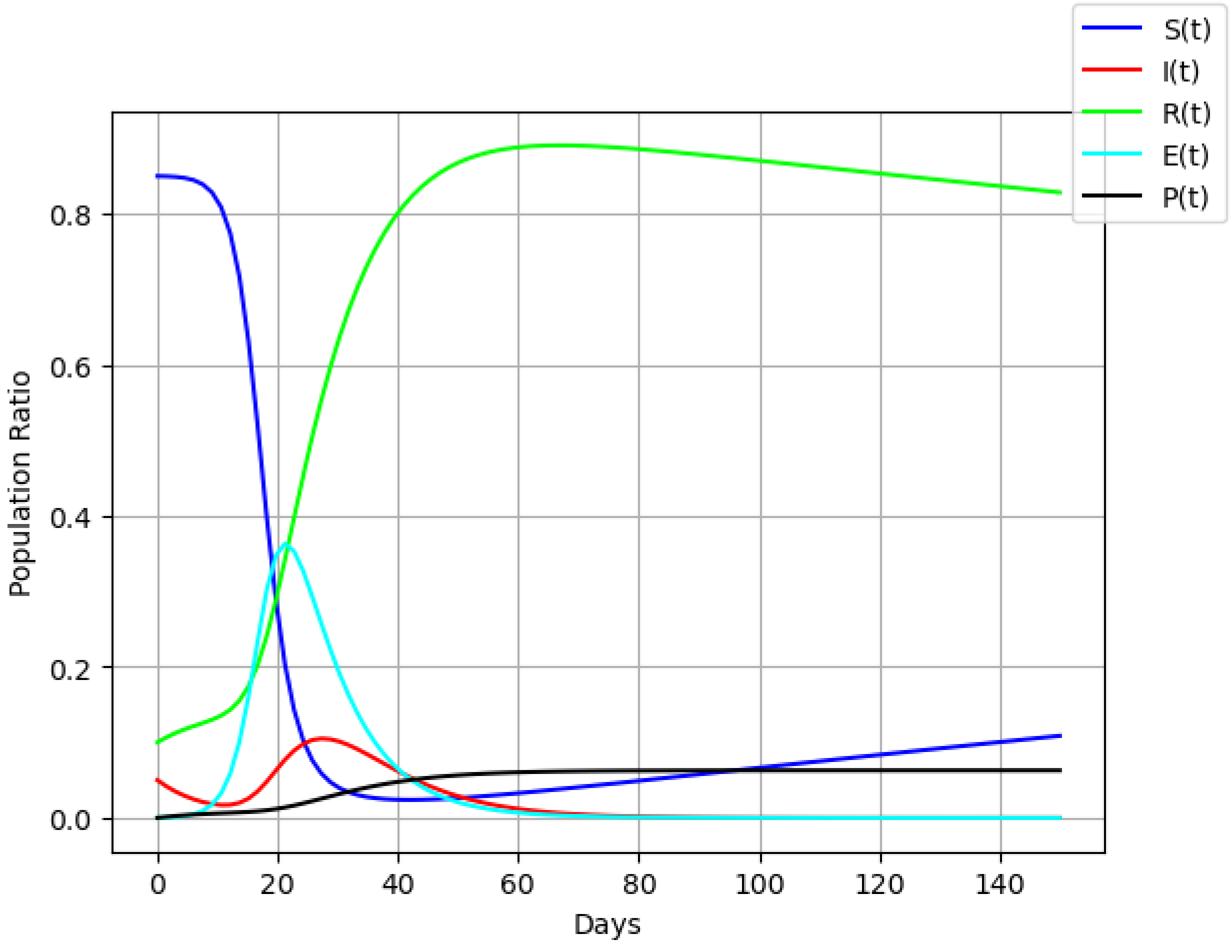
$$\frac{dr(t)}{dt} = \beta i(t) + \rho e(t) - \gamma r(t)$$

$$\frac{dp(t)}{dt} = \mu i(t)$$

In [2]:
```python
# The SIR model differential equations.
def deriv(y, t, alpha_E, alpha_I, gamma, k, q ,beta, landa):
    S, E, I, R, P = y
    dSdt = (-1) * alpha_E * S * E + (-1) * alpha_I * S * I + landa * R
    dEdt =        alpha_E * S * E +        alpha_I * S * I - k * E - q * E
    dIdt = k * E  - beta * I - gamma * I
    dRdt = beta * I + q * E - landa * R
    dPdt = gamma * I
    return dSdt, dEdt, dIdt, dRdt, dPdt
```

In [3]:
```python
# Initial number of S, E, I, R, P
E0, I0, R0, P0 =  0, 0.05, 0.1, 0
S0 = 1 - E0 - I0 - R0 - P0

# Initial number of alpha_E, alpha_I, gamma, k, q ,beta, landa
alpha_E, alpha_I, gamma, k, q ,beta, landa = 0.65, 0.005, 0.02, 0.05, 0.08, 0.1, 0.001

# A grid of time in days
t = np.linspace(0, 150, 100)
```

In [4]:
```python
# Initial conditions vector
y0 = S0, E0, I0, R0, P0

# Integrate the SIR equations over the time grid, t.
Solution = odeint(deriv, y0, t, args=(alpha_E, alpha_I, gamma, k, q ,beta, landa))
S, E, I, R, P = Solution.T

# Plot the data
fig = plt.figure()
plt.grid(True)
plt.plot(t, S, 'b'    , label='S(t)')
plt.plot(t, I, 'r'    , label='I(t)')
plt.plot(t, R, 'lime' , label='R(t)')
plt.plot(t, E, 'cyan' , label='E(t)')
plt.plot(t, P, 'k'    , label='P(t)')
plt.xlabel('Days')
plt.ylabel('Population Ratio')
legend = fig.legend()
```



In [ ]:

In [29]:

In [ ]: