In [122…

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.integrate as integrate
from ipywidgets import interact
```

In [152…

```
#Make paraters global
global c
global mu
global nu
global alpha
global beta
global gamma
global delta

c = 0.19
mu = 0.03
nu = 0.003
alpha = 800
beta = 1.5
gamma = 0.004
delta = 2.2
```

In [144…

```
def xdot(state: tuple , t : np.ndarray, r: float):
    x, y = state


    cross = (alpha*x*y) * (1/(beta + x))

    dxdt = x*(r - c*x)*(x - mu)* (1/(nu + x)) - cross
    dydt = gamma * cross - delta * y
    return [dxdt, dydt]
```

In [ ]:

```
num_r = 100
rspan = np.linspace(1.4, 2.6, num_r)
timespan = np.linspace(0, 100, 1000)
x0, y0 = 3, 0.010
init = [x0, y0]
fig, axs = plt.subplots(1, figsize = (12, 12))

period_limits = []
for i, r in enumerate(rspan):
    sol = integrate.odeint(xdot, init, timespan, args=(r, ))
    x, y = sol.T
    xmax = x / max(x)

    #finding period ->Taking derivative and plotting zeros

    x_max_pts = []

    prev_diff = 0
    for j, diff in enumerate(np.diff(xmax)):
        if prev_diff >= 0 and diff <= 0:
            x_max_pts.append((j, xmax[j]))
        prev_diff = diff

    period_limits.append(np.diff(np.array(x_max_pts)[:, 0])[-1])

axs.plot(period_limits)
axs.set_ylabel("Period")
x_ticks = np.round(np.linspace(1.4, 2.6, 7), 2)
axs.set_xticklabels(x_ticks)
axs.set_xlabel("r-value")
axs.set_title("Period Times for r in (1.4, 2.6)")
plt.tight_layout(pad = 0.7)
```