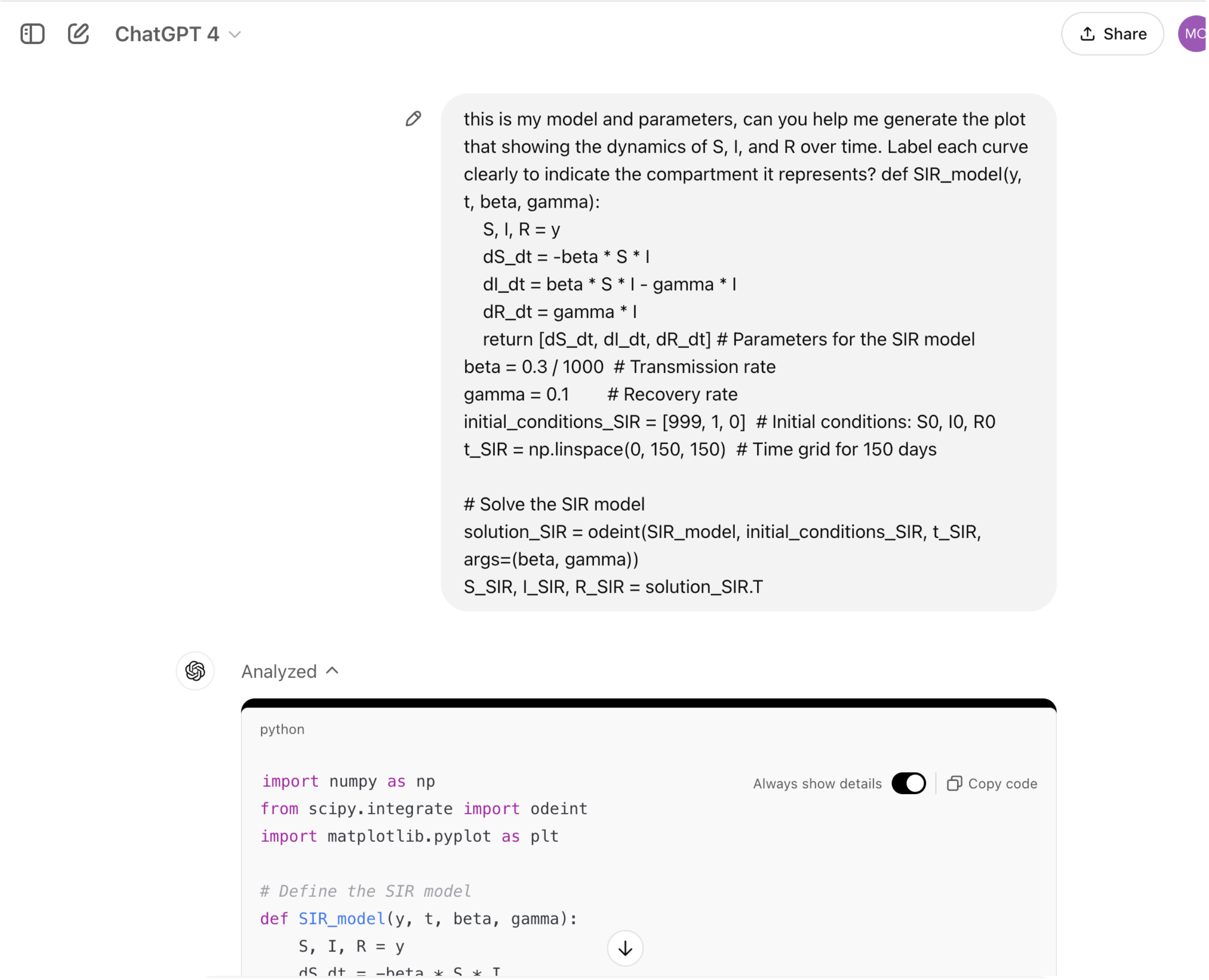
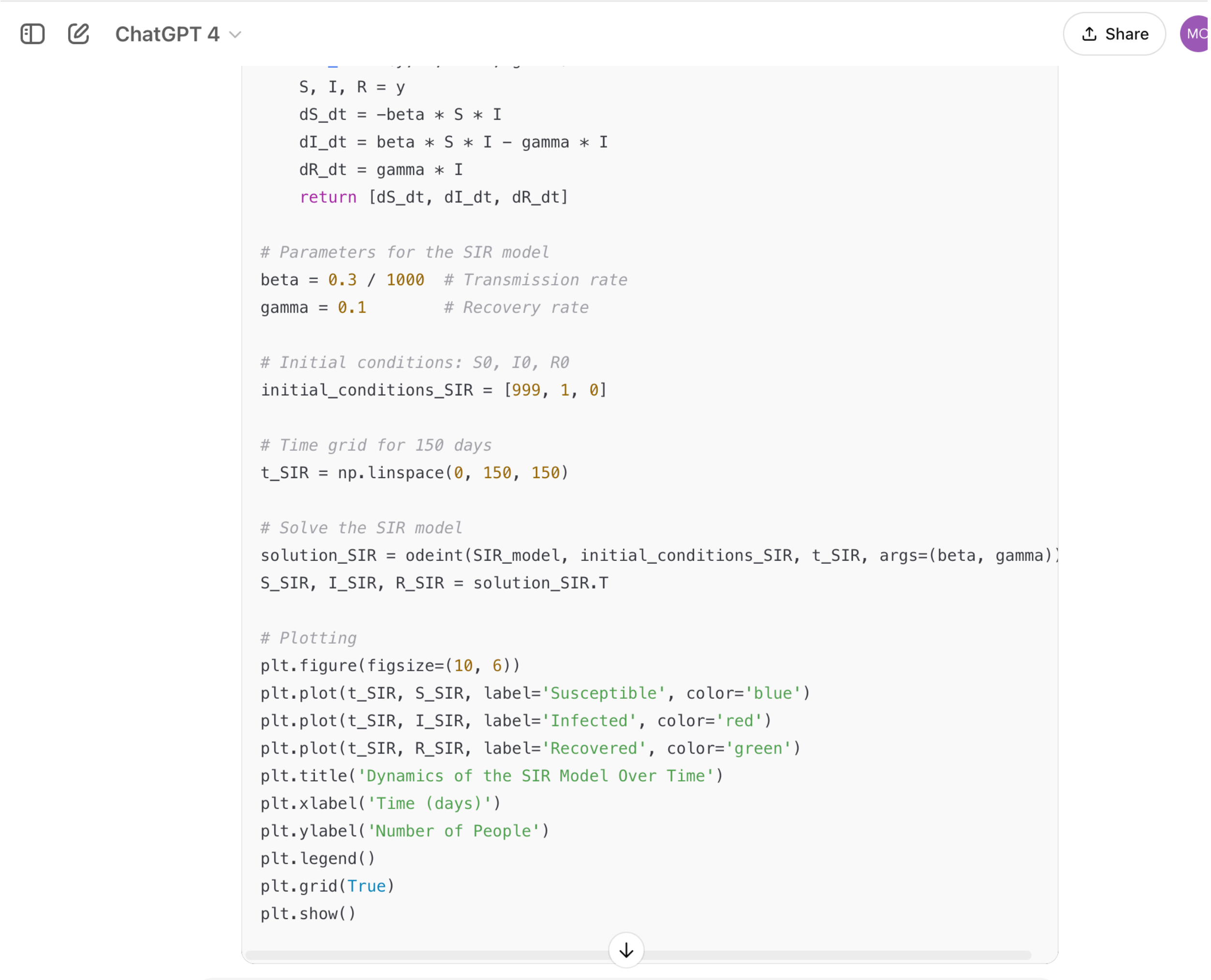


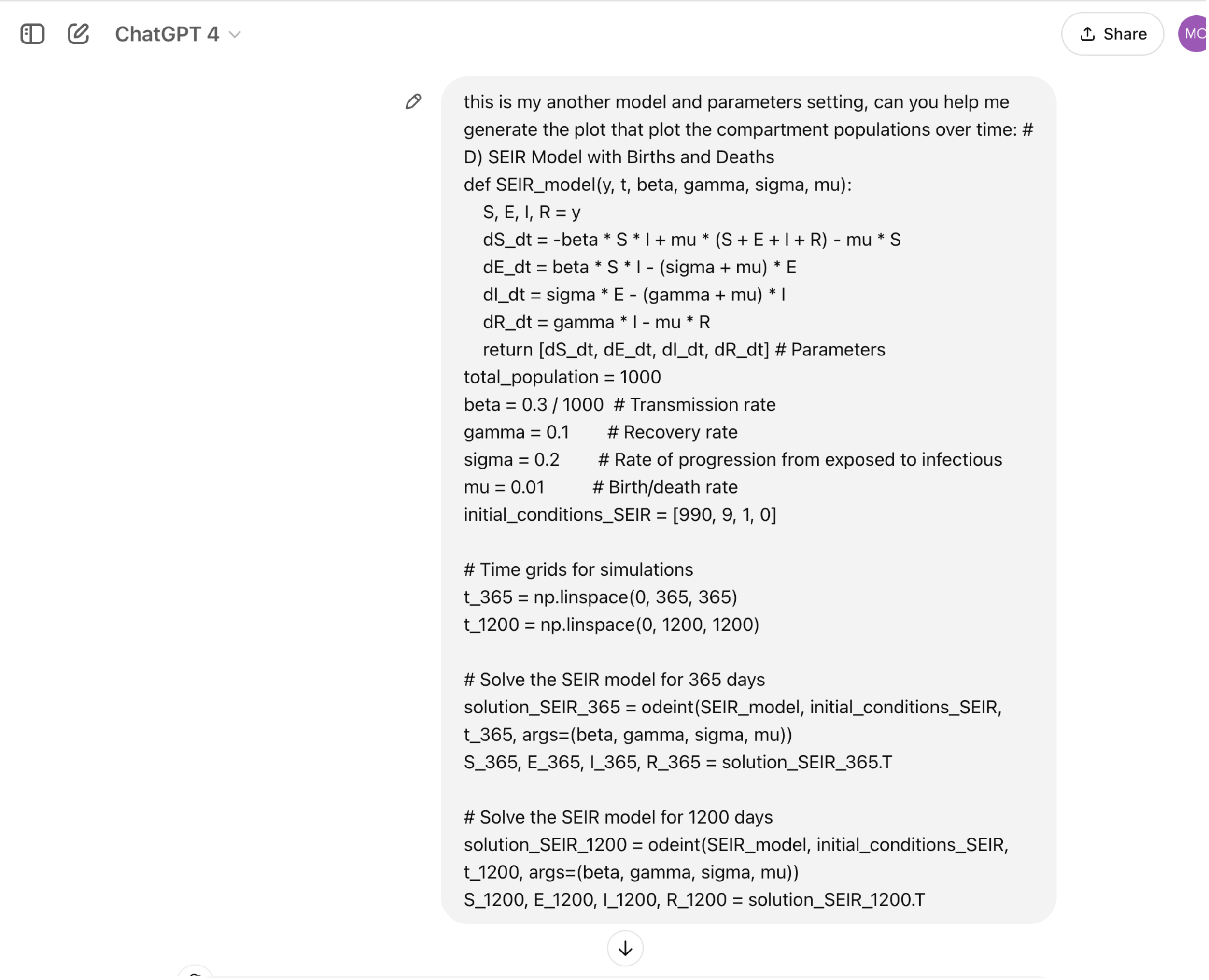
this is my model and parameters, can you help me generate the plot that showing the dynamics of S, I, and R over time. Label each curve clearly to indicate the compartment it represents? def SIR_model(y, t, beta, gamma):
    S, I, R = y
    dS_dt = -beta * S * I
    dI_dt = beta * S * I - gamma * I
    dR_dt = gamma * I
    return [dS_dt, dI_dt, dR_dt] # Parameters for the SIR model
beta = 0.3 / 1000 # Transmission rate
gamma = 0.1   # Recovery rate
initial_conditions_SIR = [999, 1, 0] # Initial conditions: S0, I0, R0
t_SIR = np.linspace(0, 150, 150) # Time grid for 150 days

# Solve the SIR model
solution_SIR = odeint(SIR_model, initial_conditions_SIR, t_SIR, args=(beta, gamma))
S_SIR, I_SIR, R_SIR = solution_SIR.T

```python
import numpy as np
from scipy.integrate import odeint
import matplotlib.pyplot as plt

# Define the SIR model
def SIR_model(y, t, beta, gamma):
    S, I, R = y
    dS_dt = -beta * S * I
```

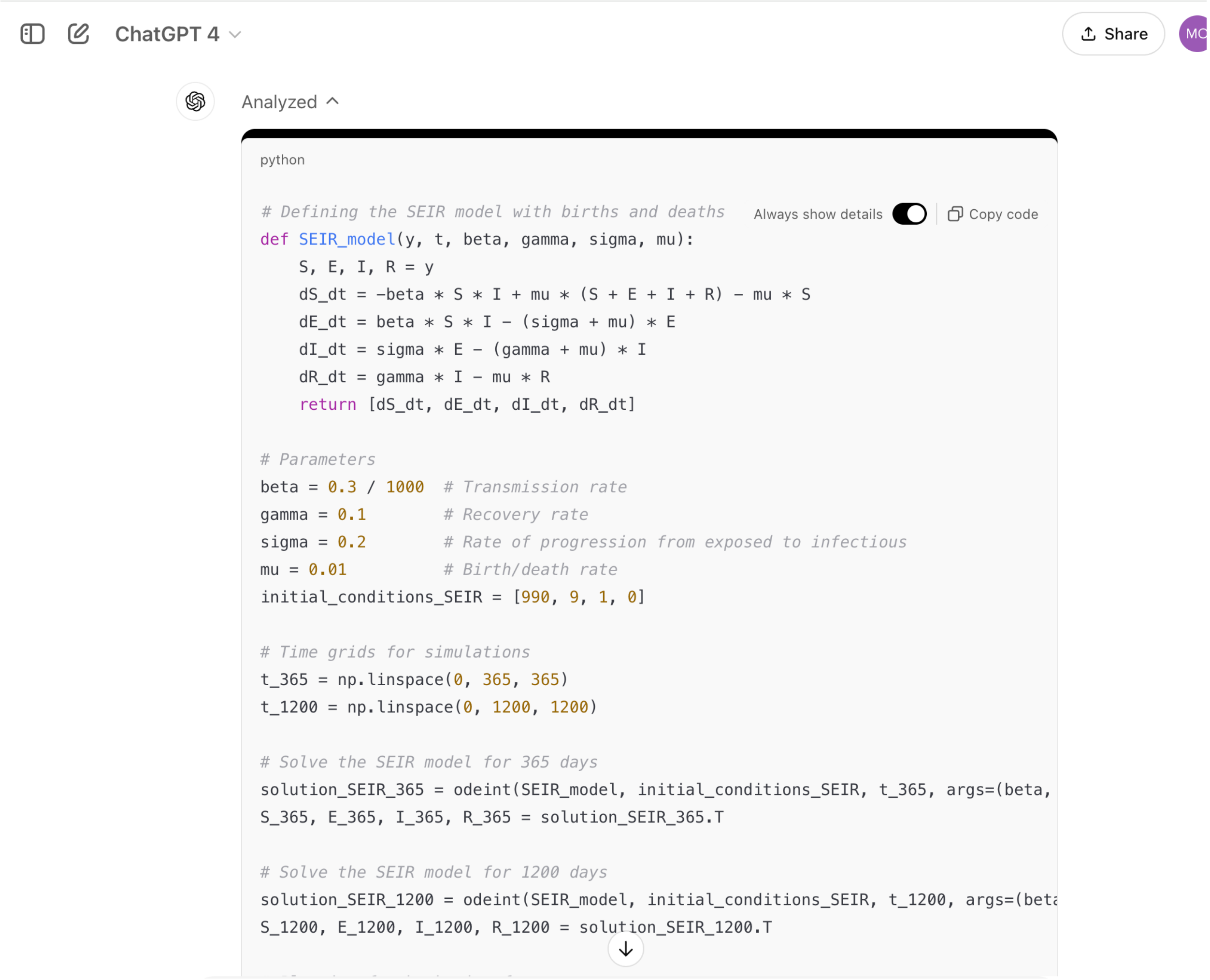


```
    S, I, R = y
    dS_dt = -beta * S * I
    dI_dt = beta * S * I - gamma * I
    dR_dt = gamma * I
    return [dS_dt, dI_dt, dR_dt]

# Parameters for the SIR model
beta = 0.3 / 1000  # Transmission rate
gamma = 0.1        # Recovery rate

# Initial conditions: S0, I0, R0
initial_conditions_SIR = [999, 1, 0]

# Time grid for 150 days
t_SIR = np.linspace(0, 150, 150)

# Solve the SIR model
solution_SIR = odeint(SIR_model, initial_conditions_SIR, t_SIR, args=(beta, gamma)
S_SIR, I_SIR, R_SIR = solution_SIR.T

# Plotting
plt.figure(figsize=(10, 6))
plt.plot(t_SIR, S_SIR, label='Susceptible', color='blue')
plt.plot(t_SIR, I_SIR, label='Infected', color='red')
plt.plot(t_SIR, R_SIR, label='Recovered', color='green')
plt.title('Dynamics of the SIR Model Over Time')
plt.xlabel('Time (days)')
plt.ylabel('Number of People')
plt.legend()
plt.grid(True)
plt.show()
```

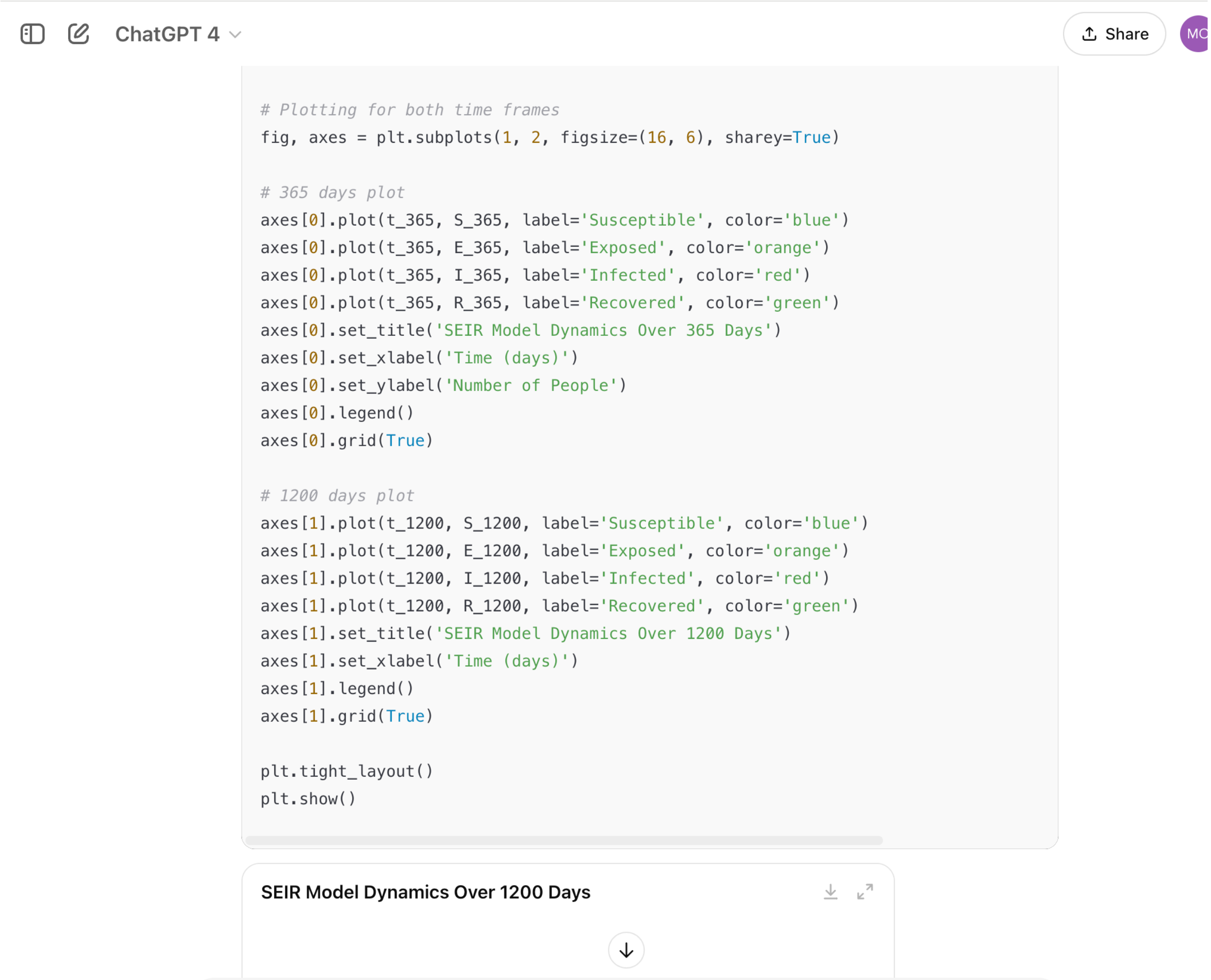


this is my another model and parameters setting, can you help me generate the plot that plot the compartment populations over time:

```
# D) SEIR Model with Births and Deaths
def SEIR_model(y, t, beta, gamma, sigma, mu):
    S, E, I, R = y
    dS_dt = -beta * S * I + mu * (S + E + I + R) - mu * S
    dE_dt = beta * S * I - (sigma + mu) * E
    dI_dt = sigma * E - (gamma + mu) * I
    dR_dt = gamma * I - mu * R
    return [dS_dt, dE_dt, dI_dt, dR_dt] # Parameters
total_population = 1000
beta = 0.3 / 1000  # Transmission rate
gamma = 0.1        # Recovery rate
sigma = 0.2        # Rate of progression from exposed to infectious
mu = 0.01          # Birth/death rate
initial_conditions_SEIR = [990, 9, 1, 0]

# Time grids for simulations
t_365 = np.linspace(0, 365, 365)
t_1200 = np.linspace(0, 1200, 1200)

# Solve the SEIR model for 365 days
solution_SEIR_365 = odeint(SEIR_model, initial_conditions_SEIR, t_365, args=(beta, gamma, sigma, mu))
S_365, E_365, I_365, R_365 = solution_SEIR_365.T

# Solve the SEIR model for 1200 days
solution_SEIR_1200 = odeint(SEIR_model, initial_conditions_SEIR, t_1200, args=(beta, gamma, sigma, mu))
S_1200, E_1200, I_1200, R_1200 = solution_SEIR_1200.T
```

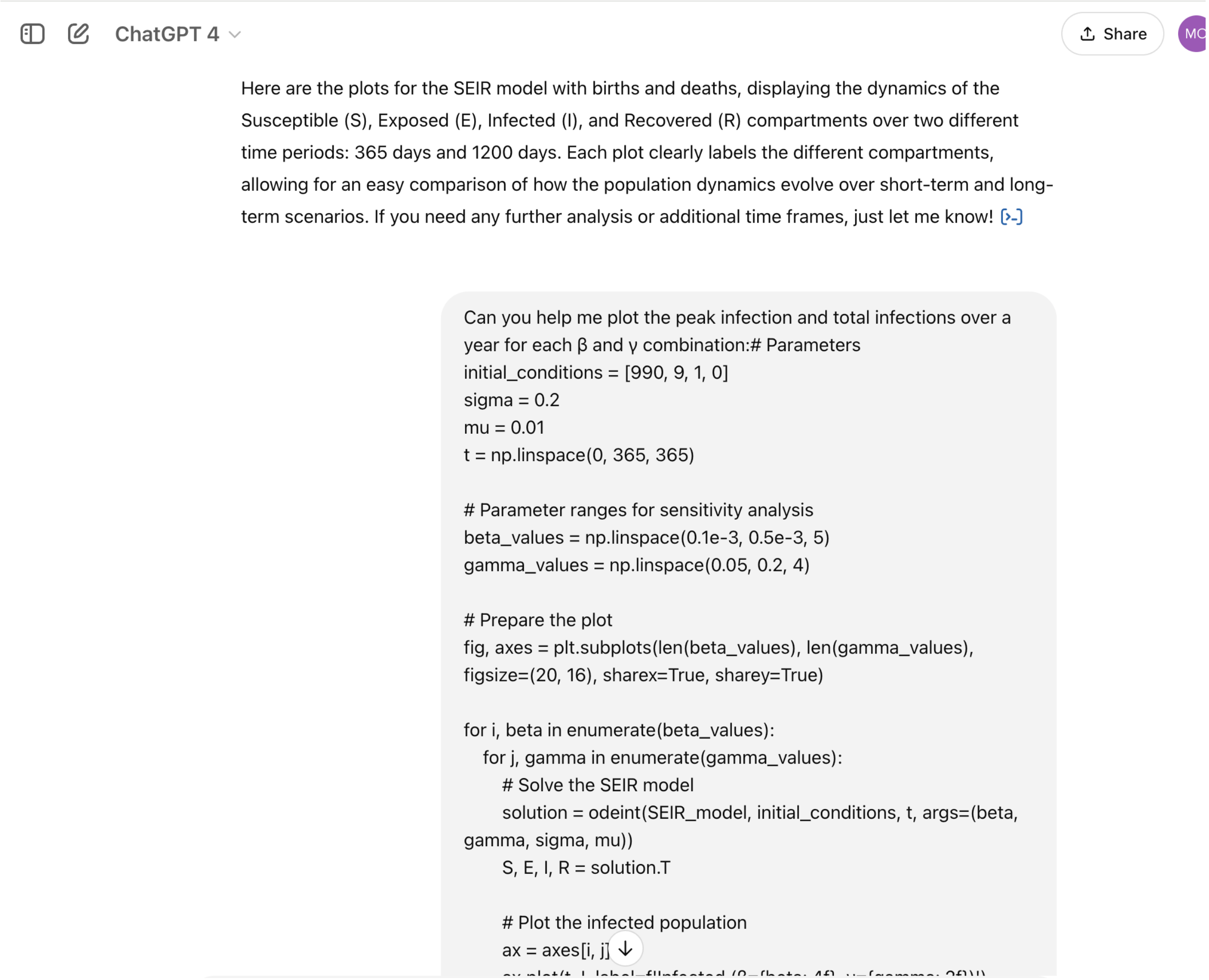


```python
# Defining the SEIR model with births and deaths
def SEIR_model(y, t, beta, gamma, sigma, mu):
    S, E, I, R = y
    dS_dt = -beta * S * I + mu * (S + E + I + R) - mu * S
    dE_dt = beta * S * I - (sigma + mu) * E
    dI_dt = sigma * E - (gamma + mu) * I
    dR_dt = gamma * I - mu * R
    return [dS_dt, dE_dt, dI_dt, dR_dt]

# Parameters
beta = 0.3 / 1000  # Transmission rate
gamma = 0.1        # Recovery rate
sigma = 0.2        # Rate of progression from exposed to infectious
mu = 0.01          # Birth/death rate
initial_conditions_SEIR = [990, 9, 1, 0]

# Time grids for simulations
t_365 = np.linspace(0, 365, 365)
t_1200 = np.linspace(0, 1200, 1200)

# Solve the SEIR model for 365 days
solution_SEIR_365 = odeint(SEIR_model, initial_conditions_SEIR, t_365, args=(beta,
S_365, E_365, I_365, R_365 = solution_SEIR_365.T

# Solve the SEIR model for 1200 days
solution_SEIR_1200 = odeint(SEIR_model, initial_conditions_SEIR, t_1200, args=(bet
S_1200, E_1200, I_1200, R_1200 = solution_SEIR_1200.T
```



```
# Plotting for both time frames
fig, axes = plt.subplots(1, 2, figsize=(16, 6), sharey=True)

# 365 days plot
axes[0].plot(t_365, S_365, label='Susceptible', color='blue')
axes[0].plot(t_365, E_365, label='Exposed', color='orange')
axes[0].plot(t_365, I_365, label='Infected', color='red')
axes[0].plot(t_365, R_365, label='Recovered', color='green')
axes[0].set_title('SEIR Model Dynamics Over 365 Days')
axes[0].set_xlabel('Time (days)')
axes[0].set_ylabel('Number of People')
axes[0].legend()
axes[0].grid(True)

# 1200 days plot
axes[1].plot(t_1200, S_1200, label='Susceptible', color='blue')
axes[1].plot(t_1200, E_1200, label='Exposed', color='orange')
axes[1].plot(t_1200, I_1200, label='Infected', color='red')
axes[1].plot(t_1200, R_1200, label='Recovered', color='green')
axes[1].set_title('SEIR Model Dynamics Over 1200 Days')
axes[1].set_xlabel('Time (days)')
axes[1].legend()
axes[1].grid(True)

plt.tight_layout()
plt.show()
```

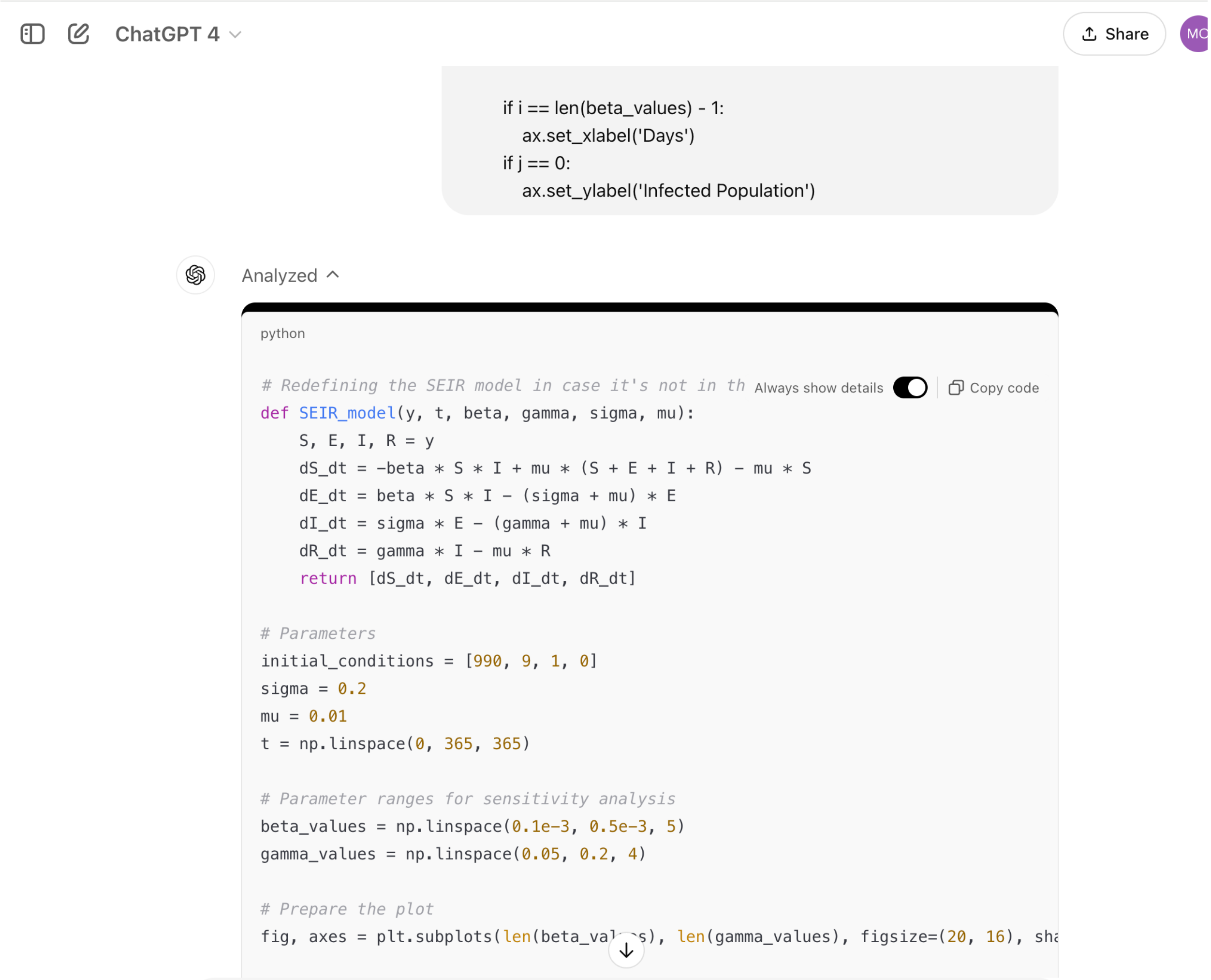
SEIR Model Dynamics Over 1200 Days



Here are the plots for the SEIR model with births and deaths, displaying the dynamics of the Susceptible (S), Exposed (E), Infected (I), and Recovered (R) compartments over two different time periods: 365 days and 1200 days. Each plot clearly labels the different compartments, allowing for an easy comparison of how the population dynamics evolve over short-term and long-term scenarios. If you need any further analysis or additional time frames, just let me know! [>_]

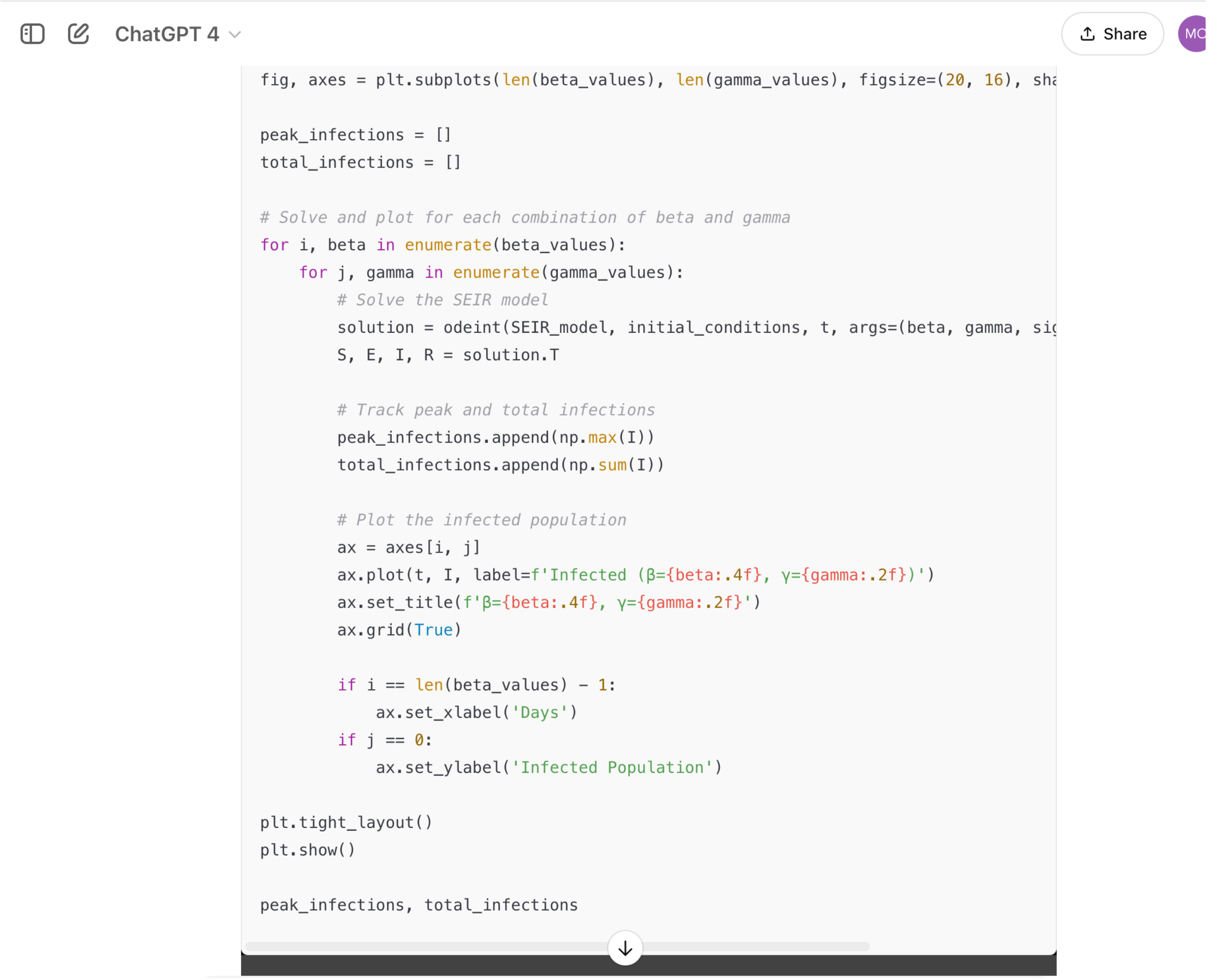
Can you help me plot the peak infection and total infections over a year for each β and γ combination:# Parameters

```
initial_conditions = [990, 9, 1, 0]
sigma = 0.2
mu = 0.01
t = np.linspace(0, 365, 365)

# Parameter ranges for sensitivity analysis
beta_values = np.linspace(0.1e-3, 0.5e-3, 5)
gamma_values = np.linspace(0.05, 0.2, 4)

# Prepare the plot
fig, axes = plt.subplots(len(beta_values), len(gamma_values),
figsize=(20, 16), sharex=True, sharey=True)

for i, beta in enumerate(beta_values):
    for j, gamma in enumerate(gamma_values):
        # Solve the SEIR model
        solution = odeint(SEIR_model, initial_conditions, t, args=(beta,
gamma, sigma, mu))
        S, E, I, R = solution.T

        # Plot the infected population
        ax = axes[i, j]
        ax.plot(t, I, label=f'Infected (β={beta:.4f}, γ={gamma:.2f})')
```

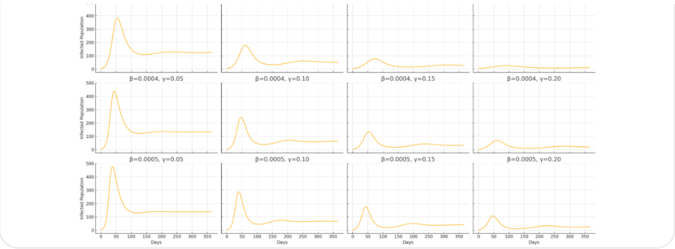


```
if i == len(beta_values) - 1:
    ax.set_xlabel('Days')
if j == 0:
    ax.set_ylabel('Infected Population')
```

```python
# Redefining the SEIR model in case it's not in th
def SEIR_model(y, t, beta, gamma, sigma, mu):
    S, E, I, R = y
    dS_dt = -beta * S * I + mu * (S + E + I + R) - mu * S
    dE_dt = beta * S * I - (sigma + mu) * E
    dI_dt = sigma * E - (gamma + mu) * I
    dR_dt = gamma * I - mu * R
    return [dS_dt, dE_dt, dI_dt, dR_dt]

# Parameters
initial_conditions = [990, 9, 1, 0]
sigma = 0.2
mu = 0.01
t = np.linspace(0, 365, 365)

# Parameter ranges for sensitivity analysis
beta_values = np.linspace(0.1e-3, 0.5e-3, 5)
gamma_values = np.linspace(0.05, 0.2, 4)

# Prepare the plot
fig, axes = plt.subplots(len(beta_val  s), len(gamma_values), figsize=(20, 16), sha
```



```python
fig, axes = plt.subplots(len(beta_values), len(gamma_values), figsize=(20, 16), sha

peak_infections = []
total_infections = []

# Solve and plot for each combination of beta and gamma
for i, beta in enumerate(beta_values):
    for j, gamma in enumerate(gamma_values):
        # Solve the SEIR model
        solution = odeint(SEIR_model, initial_conditions, t, args=(beta, gamma, sig
        S, E, I, R = solution.T

        # Track peak and total infections
        peak_infections.append(np.max(I))
        total_infections.append(np.sum(I))

        # Plot the infected population
        ax = axes[i, j]
        ax.plot(t, I, label=f'Infected (β={beta:.4f}, γ={gamma:.2f})')
        ax.set_title(f'β={beta:.4f}, γ={gamma:.2f}')
        ax.grid(True)

        if i == len(beta_values) - 1:
            ax.set_xlabel('Days')
        if j == 0:
            ax.set_ylabel('Infected Population')

plt.tight_layout()
plt.show()

peak_infections, total_infections
```



The plots for the SEIR model over 365 days for different combinations of the transmission rate (β) and recovery rate (γ) have been generated. Each subplot represents the dynamics of the infected population for a specific β and γ combination.

I also calculated the peak and total infections over the year for each combination:

- **Peak Infections** range from a few individuals to several hundreds depending on the combination of parameters, indicating the highest number of simultaneously infected individuals.
- **Total Infections** represent the cumulative number of infections over the year, reflecting the overall burden of the disease on the population.

Here are a few examples from the results:

- For the lowest β (0.0001) and lowest γ (0.05), the peak infection was approximately 94 individuals with about 21,430 total infections over the year.
- For the highest β (0.0005) and highest γ (0.2), the peak infection was about 107 individuals, but the total infections were around 10,958, showing a quicker but extensive spread.

These insights are critical for understanding how variations in disease transmission and recovery rates can affect the spread and management of an infectious disease. If you need further analysis or specific details from the results, feel free t sk! [>_]