

Simulating an SEIR model with births and deaths

James Mba Azam

In this walk through, we will be simulating an SEIR model with births and deaths.

The SEIR model is a simple compartmental model used to describe the spread of infectious diseases in a population. The model divides the population into four compartments: susceptible (S), exposed (E), infectious (I), and recovered (R) individuals. The model equations describe the flow of individuals between these compartments based on the transmission rate (β), and the recovery rate (γ).

We will be using the `{deSolve}` package to solve the model equations. The `{deSolve}` package is a powerful and flexible package for solving differential equations in R. It provides a wide range of solvers for different types of differential equations.

Here, we will use the `lsoda()` function from the `{deSolve}` package to solve the SIR model. The `lsoda()` function is a wrapper for the Livermore Solver for Ordinary Differential Equations (LSODA) developed by Hindmarsh and Petzold. It is a robust and efficient solver for stiff and non-stiff differential equations.

The `lsoda()` function takes, among others, the following arguments: - **func**: the model function that defines the differential equations, - **times**: the time points at which to return results, - **y**: the initial conditions, and - **parms**: the parameters of the model.

Note the following

- For the model equations, refer to the slides.
- The script on SIR models contains in-depth explanations of coding choices made here, so refer to them if anything looks confusing. If you're not satisfied with the explanations, feel free to ask questions.

The model function

Let's start by defining the model equations.

Recall that the model equations for the SEIR model are given by:

$$\frac{dS}{dt} = \mu N - \beta SI - \mu S \quad (1)$$

$$\frac{dE}{dt} = \beta SI - \sigma E - \mu E \quad (2)$$

$$\frac{dI}{dt} = \sigma E - \gamma I - \mu I \quad (3)$$

$$\frac{dR}{dt} = \gamma I - \mu R \quad (4)$$

where:

- S is the number of susceptible individuals,
- E is the number of exposed individuals,
- I is the number of infectious individuals,
- R is the number of recovered individuals,

- β is the transmission rate,
- σ is the rate of progression from exposed to infectious,
- γ is the recovery rate,
- μ is the birth and death rate,
- N is the total population.

The model function in R is defined as follows:

```
# 1. Define the SEIR model
seir_model <- function(t, y, parms) {
  with(as.list(c(y, parms)), {
    dS <- mu * N - beta * S * I - mu * S
    dE <- beta * S * I - sigma * E - mu * E
    dI <- sigma * E - gamma * I - mu * I
    dR <- gamma * I - mu * R
    return(list(c(dS, dE, dI, dR)))
  })
}
```

Note the following about the function defined above:

- We use the `with()` function to evaluate the expressions in the context of the list created by `as.list(c(state, parameters))`. This allows us to access the state variables and parameters directly without having to use the `$` operator. This makes the code easier to read and write (Run `?with` in the R console and read the documentation of the first three arguments).
- The function returns a list of the derivatives of the state variables. This is the format expected by the `lsoda()` function (Run `?deSolve::lsoda` in the R console and read the documentation of the first three arguments).

Model parameters

Next, we will define the parameter values and initial conditions.

The model has four parameters: β , σ , γ , and μ .

We will define arbitrary values for these parameters for illustrative purposes.

Question: Can you interpret the values of the parameters?

```
# 1. Define the SEIR model
seir_model <- function(t, y, parms) {
  with(as.list(c(y, parms)), {
    dS <- mu * N - beta * S * I - mu * S
    dE <- beta * S * I - sigma * E - mu * E
    dI <- sigma * E - gamma * I - mu * I
    dR <- gamma * I - mu * R
    return(list(c(dS, dE, dI, dR)))
  })
}

# Set parameter values
beta <- 520 / 365
sigma <- 1 / 60
gamma <- 1 / 30
mu <- 774835 / (65640000 * 365)
parms <- c(beta, sigma, gamma, mu)
```

Initial conditions

We will assume the outbreak starts with 0.1 infectious individuals, 0.1 exposed individuals, and 0 recovered individuals. The population is completely susceptible at the start of the simulation, so the number of susceptible individuals is $N - E(0) - I(0) - R(0)$.

```
N <- 1
E_0 <- 0.1
I_0 <- 0.1
R_0 <- 0
inits <- c(S = N - (E_0 + I_0 + R_0), E = E_0, I = I_0, R = R_0)
```

Time steps to return results

`lsoda()` also expects a vector of time points at which to return results. We will run the model for 365 days and return results at each day.

```
dt <- 1:365
```

Solving the model

```
seir_results <- deSolve::lsoda(
  y = inits,
  times = dt,
  func = seir_model,
  parms = params
)
seir_results <- as.data.frame(seir_results)
```

Plotting the results

Now, let's plot the results.

We will use the `ggplot2` package to plot the results. We will reshape the data to long format to make it easier to plot. For this, we will use the `pivot_longer()` function from the `tidyr` package.

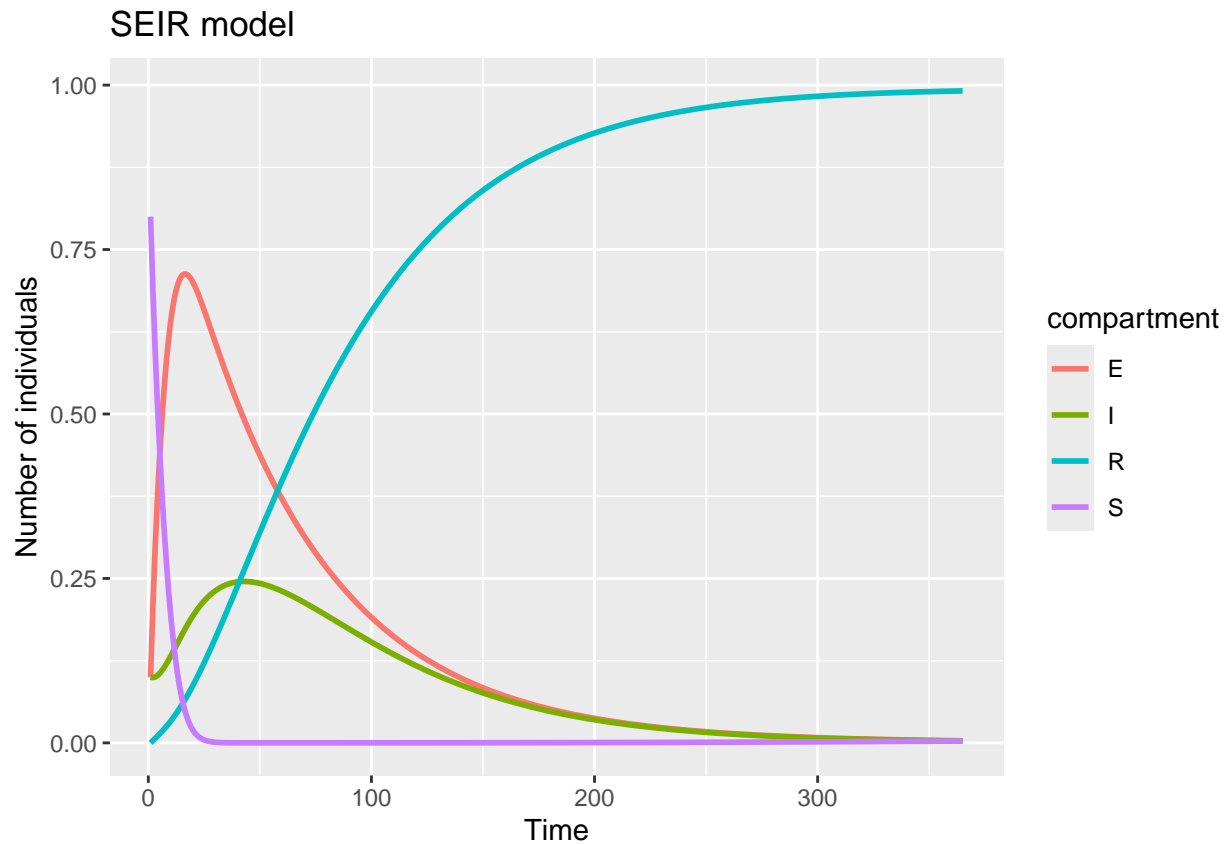
```
# Load the necessary libraries
library(ggplot2)
library(tidyr)
# Create data for ggplot2 by reshaping
results_long <- seir_results |>
  pivot_longer(
    cols = -time,
    names_to = "compartment",
    values_to = "value"
  )

seir_plot <- ggplot(
  data = results_long,
  aes(
    x = time,
    y = value,
    color = compartment
  )
) +
  geom_line(linewidth = 1) +
```

```

labs(
  title = "SEIR model",
  x = "Time",
  y = "Number of individuals"
)
plot(seir_plot)

```



Exercise

Example

- Change beta to 0.0003

```

# Load the necessary libraries
library(ggplot2)
library(tidyr)
library(deSolve)
# 1. Define the SEIR model
seir_model <- function(t, y, parms) {
  with(as.list(c(y, parms)), {
    dS <- mu * N - beta * S * I - mu * S
    dE <- beta * S * I - sigma * E - mu * E
    dI <- sigma * E - gamma * I - mu * I
    dR <- gamma * I - mu * R
    return(list(c(dS, dE, dI, dR)))
  })
}
# Set parameter values

```

```

beta <- 0.0003
sigma <- 1 / 60
gamma <- 1 / 30
mu <- 774835 / (65640000 * 365)
params <- c(beta, sigma, gamma, mu)

# Initial values
N <- 1
E_0 <- 0.1
I_0 <- 0.1
R_0 <- 0
inits <- c(S = N - (E_0 + I_0 + R_0), E = E_0, I = I_0, R = R_0)

dt <- 1:365
# Solve the model

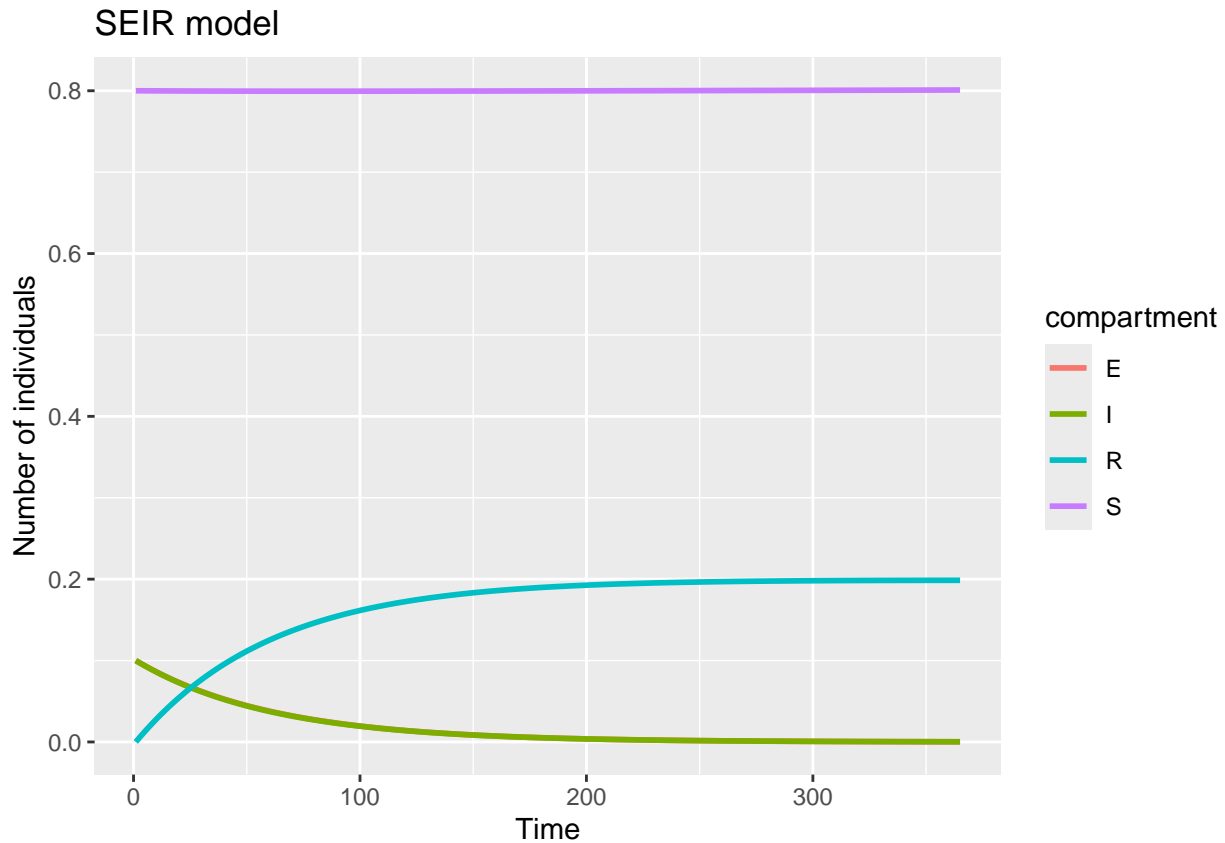
seir_results <- deSolve::lsoda(
  y = inits,
  times = dt,
  func = seir_model,
  parms = params
)

seir_results <- as.data.frame(seir_results)

# Create data for ggplot2 by reshaping
results_long <- seir_results |>
  pivot_longer(
    cols = -time,
    names_to = "compartment",
    values_to = "value"
  )

seir_plot <- ggplot(
  data = results_long,
  aes(
    x = time,
    y = value,
    color = compartment
  )
) +
  geom_line(linewidth = 1) +
  labs(
    title = "SEIR model",
    x = "Time",
    y = "Number of individuals"
  )
plot(seir_plot)

```



- Can you interpret the plot?

The S compartment is not changing.

- What influence does the birth and death rate have on the dynamics of the model?

As μ increases, I observe that S is something and I is something something.

```
# Load the necessary libraries
library(ggplot2)
library(tidyrr)
library(deSolve)

# 1. Define the SEIR model
seir_model <- function(t, y, parms) {
  with(as.list(c(y, parms)), {
    dS <- mu * N - beta * S * I - mu * S
    dE <- beta * S * I - sigma * E - mu * E
    dI <- sigma * E - gamma * I - mu * I
    dR <- gamma * I - mu * R
    return(list(c(dS, dE, dI, dR)))
  })
}

# Set parameter values
beta <- 0.25
sigma <- 1 / 14
gamma <- 1 / 7
# mu <- 774835 / (65640000 * 365)
mu <- 900000 / (65640000 * 365) # higher birth rate
parms <- c(beta, sigma, gamma, mu)
```

```

# Initial values
N <- 1
E_0 <- 0.1
I_0 <- 0.1
R_0 <- 0
inits <- c(S = N - (E_0 + I_0 + R_0), E = E_0, I = I_0, R = R_0)

dt <- 1:365
# Solve the model

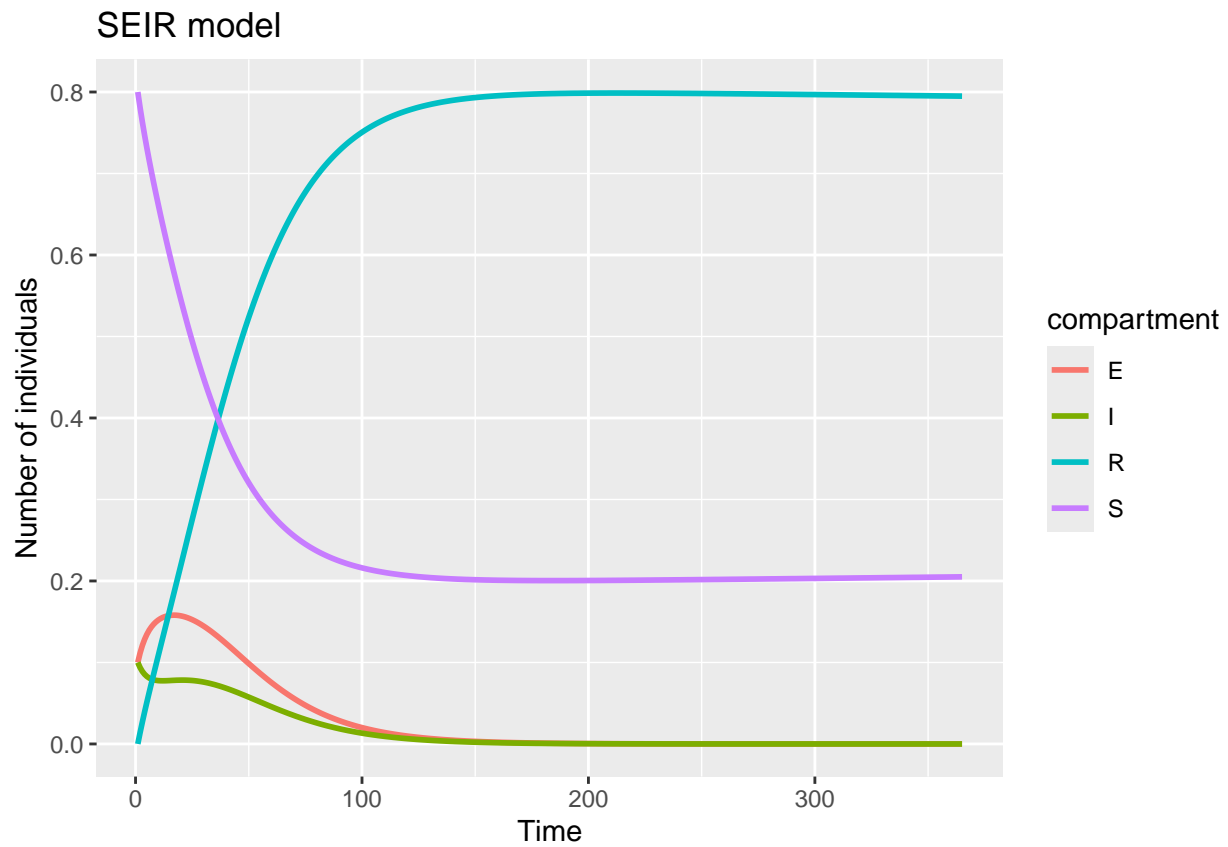
seir_results <- deSolve::lsoda(
  y = inits,
  times = dt,
  func = seir_model,
  parms = params
)

seir_results <- as.data.frame(seir_results)

# Create data for ggplot2 by reshaping
results_long <- seir_results |>
  pivot_longer(
    cols = -time,
    names_to = "compartment",
    values_to = "value"
  )

seir_plot <- ggplot(
  data = results_long,
  aes(
    x = time,
    y = value,
    color = compartment
  )
) +
  geom_line(linewidth = 1) +
  labs(
    title = "SEIR model",
    x = "Time",
    y = "Number of individuals"
  )
plot(seir_plot)

```



- What happens if the birth and death rate is set to zero?
- What happens if the birth and death rate is set to a very high value?
- What happens if the birth and death rate is set to a value higher than the recovery rate?