Practical 2: Compartmental Models to Equations

Sam Abbott

# Learning Objectives

1. How to translate a simple model flow diagram into equations
2. Understand how changing parameter values can change model dynamics
3. Using a simple model scaffold to develop a more complex model

# Timeline for Session

1. Set up (5 minutes)
2. Explore the dynamics of a simple SEIR model (10 minutes)
3. Add high and low risk latency to a SEIR model (10 minutes)
4. Explore model dynamics (10 minutes)
5. Extension: Translate your Model Flow Diagram to Equations (20 minutes)
6. Extension: Explore the parameter space of multiple models (20 minutes)(If having trouble with the previous excercises then skip to this point for an R free exercise)
7. Session wrap up (10 minutes)

# Excercises

## A Simple SEIR Model of Tuberculosis (TB)

library(biddmodellingcourse)  
library(tidyverse)  
library(knitr)

### Populations and Initialisation

We first set up the initial populations for the Susceptible (), Latent (), Infected (), and Recovered () compartments. We have initialised the model as early stage epidemic with a single case of TB.

inits = c(  
 S = 999,  
 E = 0,  
 I = 1,  
 R = 0  
)

### Parameters

We then specify the model parameters (with the units being years), varying these paremeters will impact the model dynamics.

parameters <- c(  
 beta = 3, # Rate of transmission  
 gamma = 2, # Rate of progression to active symptoms   
 tau = 1/2, # Rate of recovery  
 mu = 1/81 # Rate of natural mortality  
)

### Equations

Finally we specify the model equations for each population compartment. This is model incorperates simple demographic processes with a constant natural death rate from all compartments which is equal to the birth rate into the susceptible compartment.

SEIR\_demo\_ode <- function(t, x, params) {  
  
 ## Specify model compartments  
 S <- x[1]  
 E <- x[2]  
 I <- x[3]  
 R <- x[4]  
  
 with(as.list(params),{  
  
 ## Specify total population  
 N = S + E + I + R  
  
 ## Derivative Expressions  
 dS = - beta \* S \* I / N - mu \* S + mu \* N  
 dE = beta \* S \* I / N - gamma \* E - mu \* E  
 dI = gamma \* E - tau \* I - mu \* I  
 dR = tau \* I - mu \* R  
  
 ## output  
 derivatives <- c(dS, dE, dI, dR)  
  
 list(derivatives)  
 })  
}

### Simulate and Summarise

To simulate the model we specify the starting year (begin\_time) and final year (end\_time) and define a sequence over all intervening years. The model is then solved using deSolve::lsoda which is used within a simple wrapper function (see ?solve\_ode for details). The resulting table summarises the simulation results for the first 5 years.

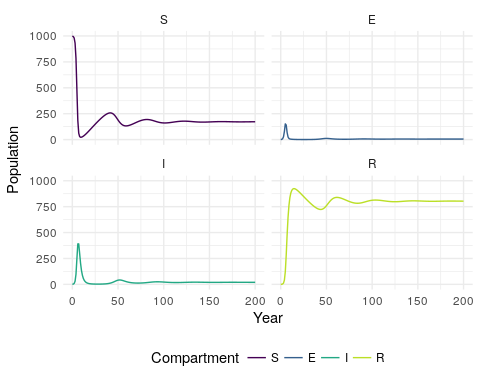
begin\_time <- 0  
end\_time <- 200  
times <- seq(begin\_time, end\_time, 1)  
  
## see ?solve\_ode for details  
SEIR\_sim <- biddmodellingcourse::solve\_ode(model = SEIR\_demo\_ode,   
 inits = inits,   
 params = parameters,  
 times = times,   
 as.data.frame = TRUE)  
SEIR\_sim %>%   
 head %>%   
 kable

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| time | S | E | I | R |
| 0 | 999.0000 | 0.000000 | 1.000000 | 0.0000000 |
| 1 | 994.8003 | 2.123367 | 2.372180 | 0.7041888 |
| 2 | 980.6200 | 7.697258 | 8.583578 | 3.0991770 |
| 3 | 931.5074 | 26.491249 | 30.334819 | 11.6665502 |
| 4 | 783.2648 | 78.184595 | 97.968856 | 40.5817168 |
| 5 | 479.7156 | 151.492041 | 246.217573 | 122.5747792 |
| Summar | ise the epi | demic peak (` | epi\_peak`) an | d epidemic duration (epi\_dur). |

summarise\_model <- function(sim) {  
 epi\_peak <- sim %>%   
 filter(I == max(I)) %>%   
 pull(time)  
   
 epi\_dur <- sim %>%   
 filter(I == 0) %>%   
 arrange(I) %>%   
 pull(time)  
   
 if (length(epi\_dur) == 0) {  
 epi\_dur <- Inf  
 }  
   
 sum\_stat <- tibble(epi\_peak = epi\_peak, epi\_dur = epi\_dur)  
  
 return(sum\_stat)  
}  
  
summarise\_model(SEIR\_sim) %>%   
 kable

|  |  |
| --- | --- |
| epi\_peak | epi\_dur |
| 6 | Inf |

order <- colnames(SEIR\_sim)[-1]  
  
SEIR\_sim %>%   
 gather(key = "Compartment", value = "Population", -time) %>%   
 mutate(Compartment = factor(Compartment, levels = order)) %>%   
 ggplot(aes(x = time, y = Population, col = Compartment)) +  
 geom\_line() +  
 theme\_minimal() +  
 labs(x = "Year") +  
 scale\_color\_viridis\_d(end = 0.9) +  
 theme(legend.position = "bottom") +   
 facet\_wrap(~Compartment)



### Explore

* What impact does varying parameters have?

## Add High and Low Risk Compartments

* outline simple model function with text indicating where solution should go
* code to plot model etc

## Extension: Translate your Model Flow Diagram to Equations

* basic model outline
* suggestions for order in which to add complexity

## Extension: Explore the Parameter Space of Multiple Models

* link to shiny app
* help with shiny app
* question ideas

# References