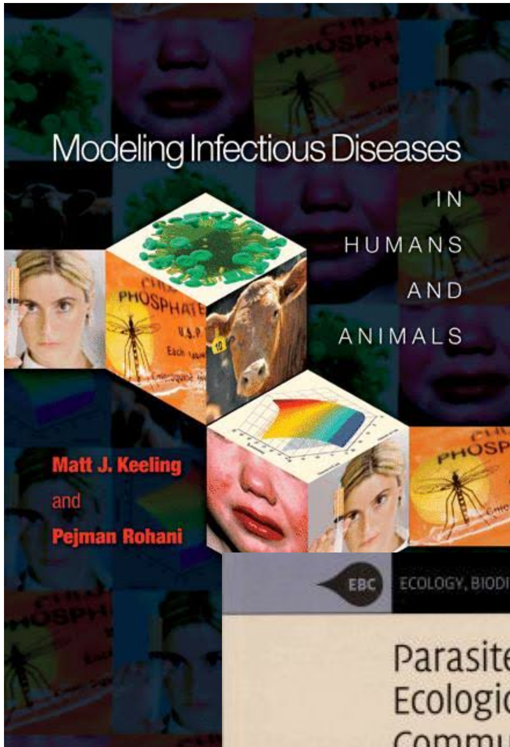
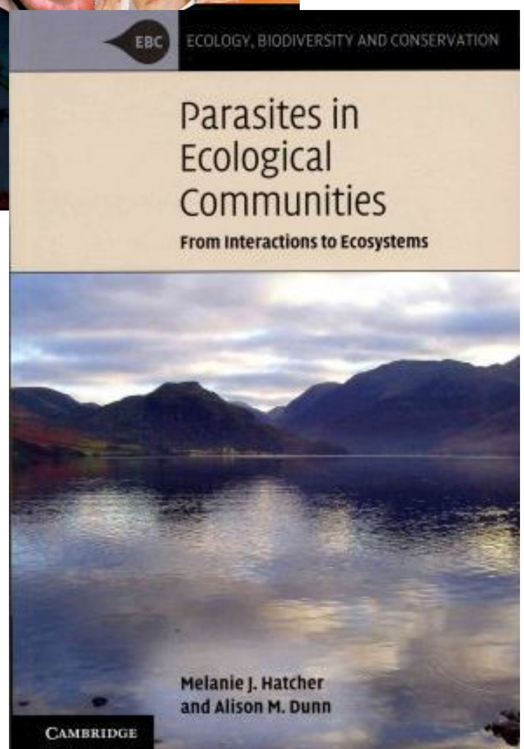
A practical guide to modelling epidemics:

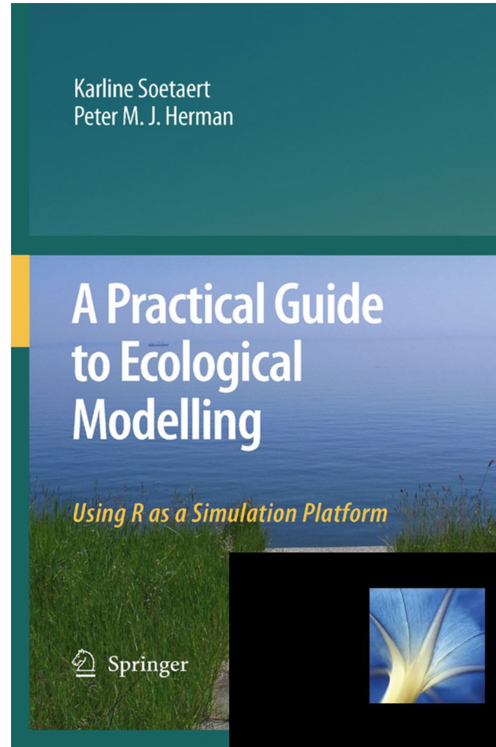
Numerical simulations in R



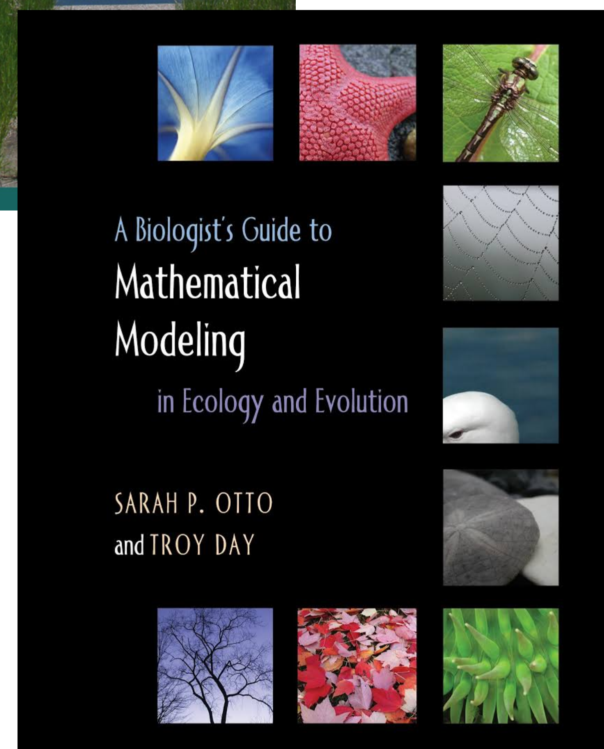
Modeling Infectious Diseases
IN
HUMANS
AND
ANIMALS
Matt J. Keeling
and
Pejman Rohani



EBC
ECOLOGY, BIODIVERSITY AND CONSERVATION
Parasites in Ecological Communities
From Interactions to Ecosystems
Melanie J. Hatcher
and Alison M. Dunn
CAMBRIDGE



Karline Soetaert
Peter M. J. Herman
A Practical Guide to Ecological Modelling
Using R as a Simulation Platform
Springer



A Biologist's Guide to
Mathematical Modeling
in Ecology and Evolution
SARAH P. OTTO
and TROY DAY

Overview

1. An introduction to epidemic models (SIS model)
2. Implementation in R

1. An introduction to epidemic models (SIS model)

An epidemic model

- SIS model (ODEs)

$$\begin{aligned}\frac{dS}{dt} &= -\beta SI + \gamma I \\ \frac{dI}{dt} &= \beta SI - \gamma I\end{aligned}$$

An epidemic model

- SIS model (ODEs)

Change in number of susceptibles & Infecteds over time

$$\frac{dS}{dt} = -\beta S I + \gamma I$$

$$\frac{dI}{dt} = \beta S I - \gamma I$$

An epidemic model

- SIS model (ODEs)

Change in number of susceptibles & Infecteds over time

Number of new infections per unit time

$$\frac{dS}{dt} = -\beta SI + \gamma I$$

$$\frac{dI}{dt} = \beta SI - \gamma I$$

An epidemic model

- SIS model (ODEs)

Change in number of susceptibles & Infecteds over time

$$\frac{dS}{dt} = -\beta S I + \gamma I$$

$$\frac{dI}{dt} = \beta S I - \gamma I$$

Number of new infections per unit time

Number of recovered hosts per unit time

An epidemic model

- SIS model (ODEs)

Change in number of susceptibles & Infecteds over time

$$\frac{dS}{dt} = -\beta SI + \gamma I$$

$$\frac{dI}{dt} = \beta SI - \gamma I$$

Number of new infections per unit time

Number of recovered hosts per unit time

Assumptions:

- No births, deaths, and migration
- Mass action
- Recovery without immunity memory

An epidemic model

- On β:
  - β is a rate, NOT a probability
  - Each infected individual makes β infectious contacts per unit time
  - $\beta \times I$ is the incoming transmission rate per susceptible individual, the force of infection

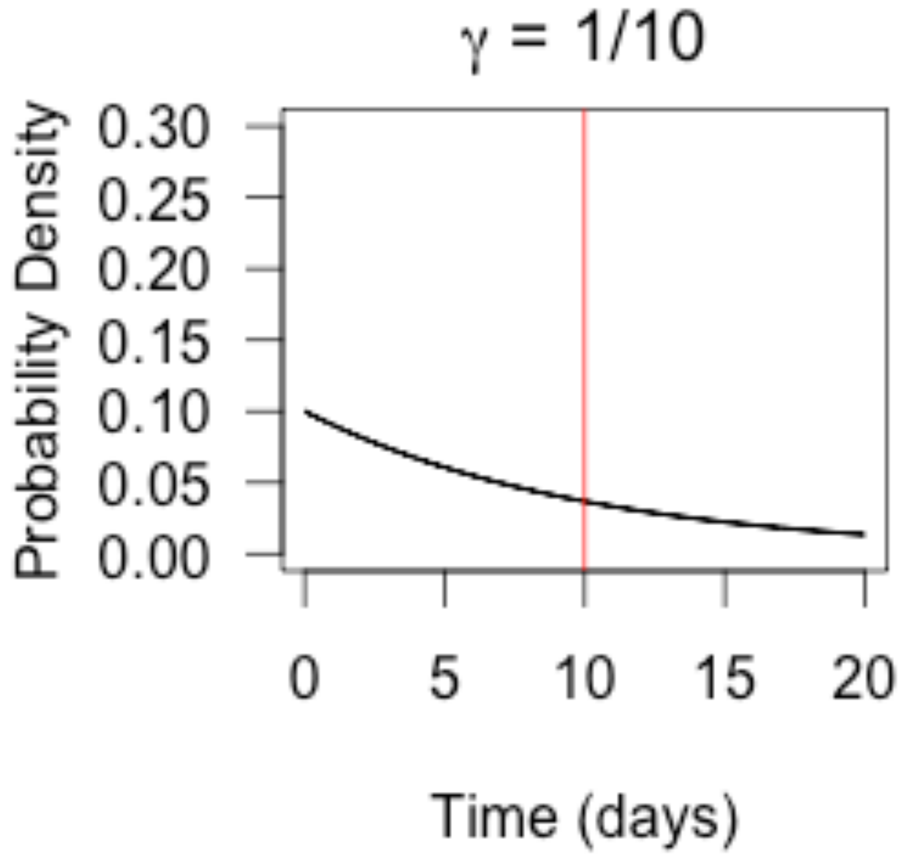
More on β: Keeling and Rohani: Box 2.1

An epidemic model

- On ɣ:
  - ɣ is also a constant rate
  - Exponential distribution:
    - E[waiting time] = 1/rate
    - 1/ ɣ = expected time until recovery
    - 1/ ɣ = duration of infection (if there is no mortality)

An epidemic model

- On ɣ:
  - ɣ is also a constant rate
  - Exponential distribution:
    - E[waiting time] = 1/rate
    - 1/ ɣ = expected time until recovery
    - 1/ ɣ = duration of infection (if there is no mortality)



An epidemic model

- On R_0:
  - The number of secondary infections per infected host in a fully susceptible population
  - Maximum reproductive potential for the parasite population
  - Tons of different methods and they rarely agree with each other (for complex models).

  - Deriving intuitively from an ODE model:
    - $R_0 = Transmission\ rate \times duration\ of\ infection$
    - For a SIS model: $R_0 = \beta/\gamma$

Modelling considerations

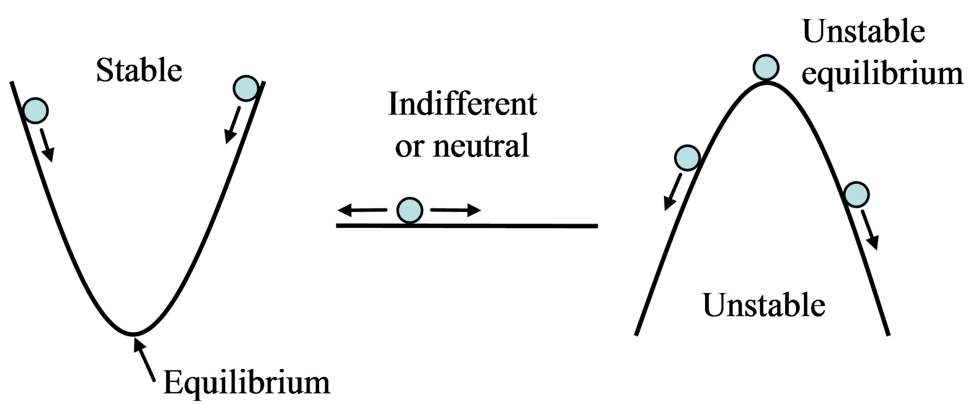
- Time
  - Continuous
  - Discrete
- Deterministic (large pop.) vs. stochastic (small pop.)
- Host demography
  - Constant population size
  - Constant input (strong externally regulation)
  - Logistic growth (similar to constant population size near K)
  - Per capita growth rate
- Host immunity
  - (waning) immune memory (SIR model)
  - No immune memory (SIS model)
- Mode of transmission
  - Density or frequency (STI and vector-borne) dependent
  - Vertical or horizontal
  - Direct, indirect or environmental
- Parasite development
  - Infected, but not infectious (exposed class)
- Parasite-induced harm
  - Mortality through infection?

Assessing epidemiological consequences

- Examples of useful metrics
  - R_0 and its relatives (emergence)
  - Emergence probability (stochastic models)
  - Peak infection prevalence (SIR models, in particular)
  - Endemic equilibrium prevalence
  - Stability (disease free & endemic equilibrium)

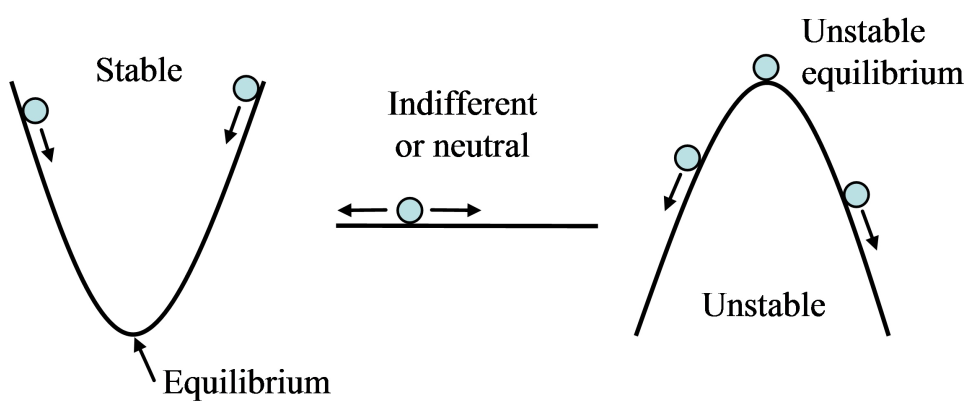
Assessing epidemiological consequences

- Examples of useful metrics
  - R_0 and its relatives (emergence)
  - Emergence probability (stochastic models)
  - Peak infection prevalence (SIR models, in particular)
  - Endemic equilibrium prevalence
  - Stability (disease free & endemic equilibrium)



Assessing epidemiological consequences

- Examples of useful metrics
  - R_0 and its relatives (emergence)
  - Emergence probability (stochastic models)
  - Peak infection prevalence (SIR models, in particular)
  - **Endemic equilibrium prevalence**
  - **Stability (disease free & endemic equilibrium)**



Calculating infection prevalence

Rate of change

$$\frac{dS}{dt} \Longrightarrow S(t)$$

$$\frac{dI}{dt} \Longrightarrow I(t)$$

Density at time t

Calculating infection prevalence

$$\frac{dS}{dt} \Longrightarrow S(t)$$

Rate of change Integration **Density at time t**

$$\frac{dI}{dt} \Longrightarrow I(t)$$

Calculating infection prevalence

Rate of change

$$\frac{dS}{dt} \Longrightarrow S(t)$$

Integration

$$\frac{dI}{dt} \Longrightarrow I(t)$$

Density at time t

- But, often impossible to integrate complex systems models analytically
  - So we resort to calculating numerical values that approximate the system
  - R offers ready-to-use numerical integrators

2. Implementation in R

Useful R packages for ODE models

- deSolve
  - Numerical integration algorithms: lsoda()
- rootSolve
  - Finding steady states (equilibrium): runsteady()
  - Stability: jacobian.full()
- GillespieSSA
  - Stochastic model
- FME
  - Sensitivity analysis

R implementation (SIS model):

1. Describing the model
2. Setting up initial conditions
3. Running the model through time
4. Finding a steady state (equilibrium)
5. Assessing stability
6. (Sensitivity analysis)

R implementation (SIS model):

1. Describing the model

```
#####  Model description  #####
SISmodel=function(t,y,parameters)
{
  ## Variables
  S=y[1]; I=y[2];

  ## Parameters
  beta = parameters[1]; gamma = parameters[2]

  ## Ordinary differential equations
  dSdt = - beta * S * I + gamma * I
  dIdt = beta * S * I - gamma * I

  return(list(c(dSdt,dIdt)));
}
###############################
```

R implementation (SIS model):

2. Setting up initial conditions

```
#####  Intial condition  #####
## Parameters
parameters=c(beta=0.001, gamma=1/10)

## Initial state
variables0=c(S0=999,I0=1)

## Times at which estimates of the variables are returned
timevec=seq(0,30,1)
################################
```

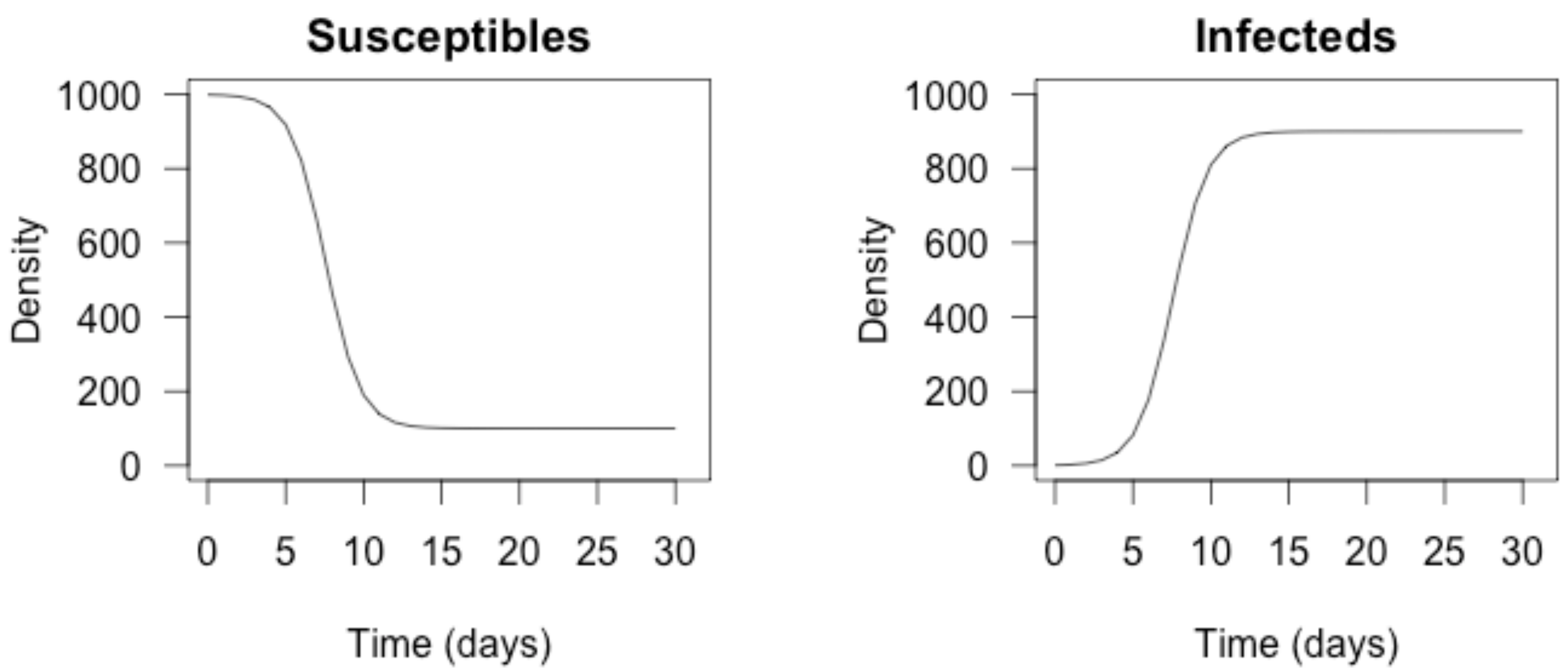
R implementation (SIS model):

3. Running the model through time

```
#####  Running the model   #####
## Numerical integration through time
output=lsoda(y = variables0,    # intial values
             times = timevec,   # time vector
             func = SISmodel,   # model
             parms = parameters # constant parameters
             )
################################
```

R implementation (SIS model):

3. Running the model through time



R implementation (SIS model):

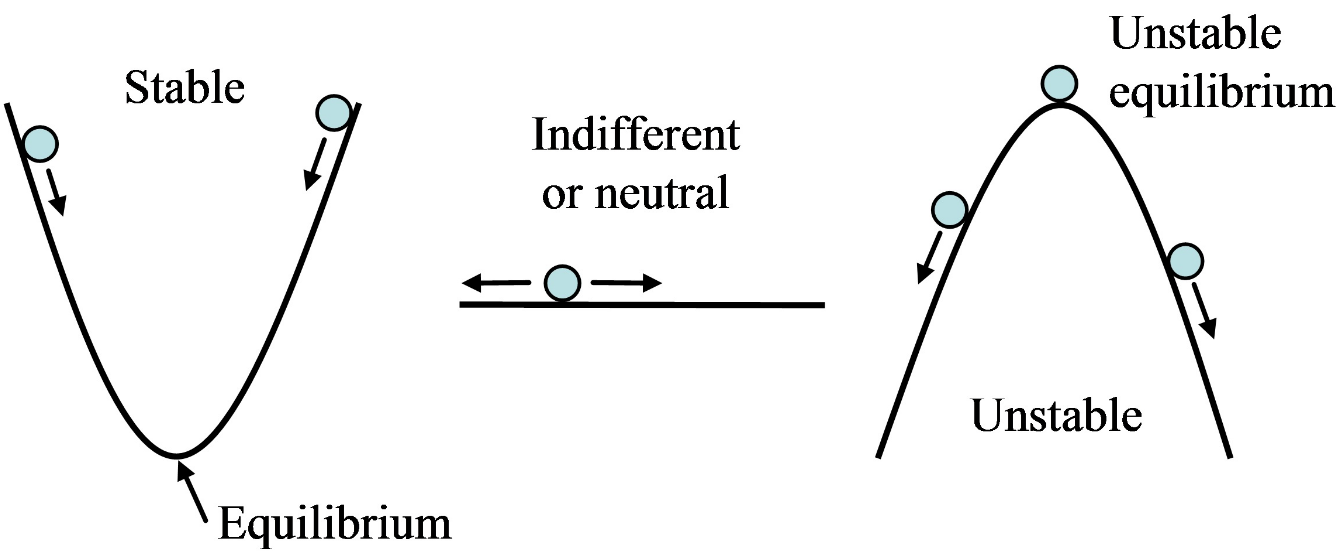
4. Finding a steady state (equilibrium)

```
##### Finding steady state #####
steadyState=runsteady(y = variables0,
                      fun = SISmodel,
                      parms = parameters)
steadyState$y
################################
```

```
> steadyState$y
 S0  I0
100 900
```

R implementation (SIS model):

5. Assessing stability of the equilibrium



More on stability analysis: Otto and Day, Chapter 5, Chapter 8

R implementation (SIS model):

5. Assessing stability of the equilibrium
   - When the dominant Eigenvalue of the Jacobian matrix is negative -> the equilibrium is stable

$$J = \begin{bmatrix} \frac{\partial S'}{\partial S} & \frac{\partial S'}{\partial I} \\ \frac{\partial I'}{\partial S} & \frac{\partial I'}{\partial I} \end{bmatrix}$$

R implementation (SIS model):

5. Assessing stability of the equilibrium
   - When the dominant Eigenvalue of the Jacobian matrix is negative -> the equilibrium is stable

```
##### Assessing the stability of the steady state #####
## Stability
Jmatrix=jacobian.full(y = steadyState$y,
                      func = SISmodel,
                      parms = parameters)
dEigenV=max(Re(eigen(Jmatrix)$values))
#######################################################
```

R implementation (SIS model):

6. (Sensitivity analysis: FME package)
   - Local sensitivity (elasticity) analysis

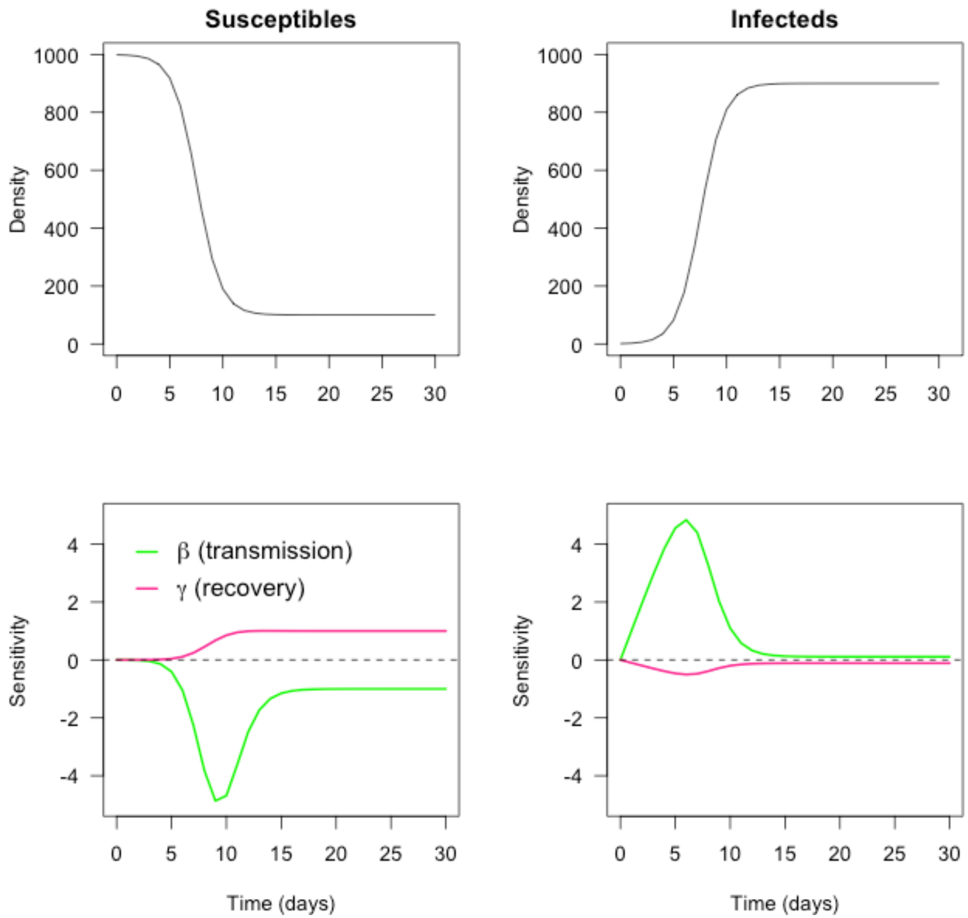
$$\frac{\partial y_i}{\partial \Theta_j} \cdot \frac{w_{\Theta_j}}{w_{y_i}}$$

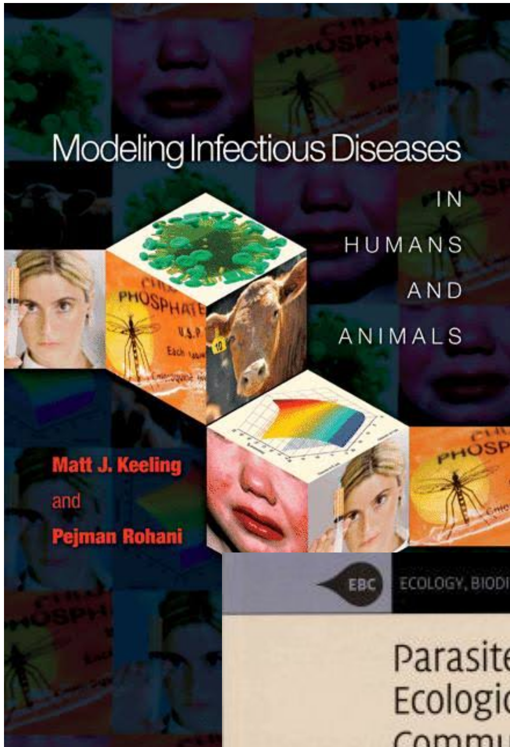
     - Where y_i is the i^{th} variable and Θ_j is the j^{th} parameter and w is a scaling factor.

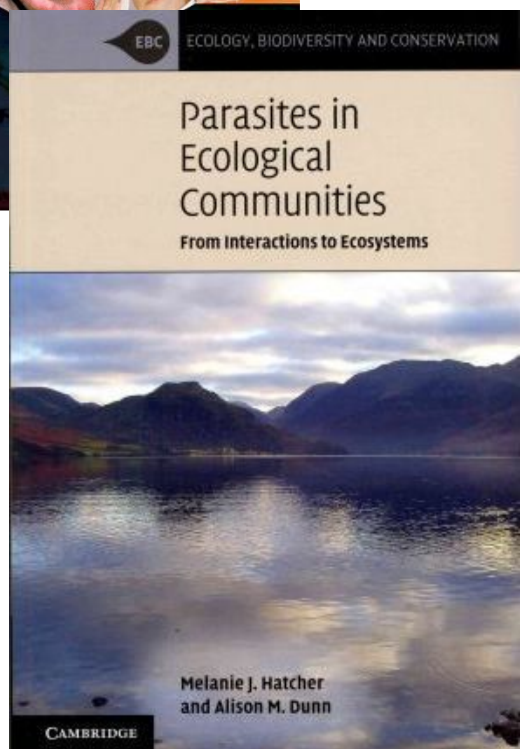
   - Soetaert, K. and Petzoldt, T., 2010. Inverse modelling, sensitivity and Monte Carlo analysis in R using package FME. *Journal of Statistical Software*, *33*(3), pp.1-28.

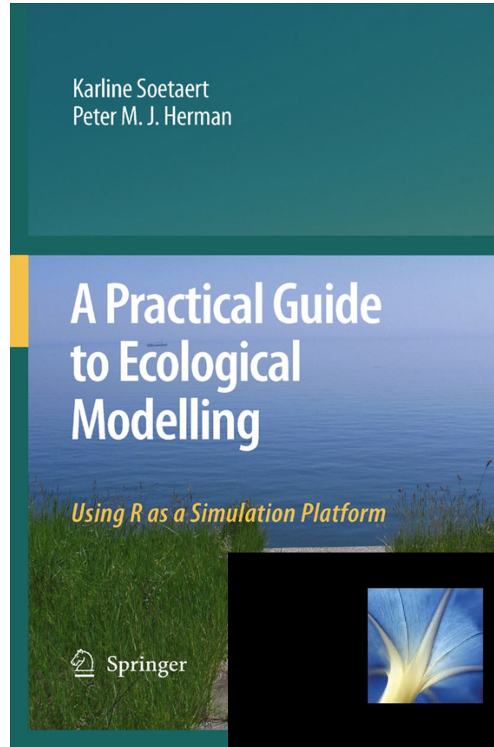
R implementation (SIS model):

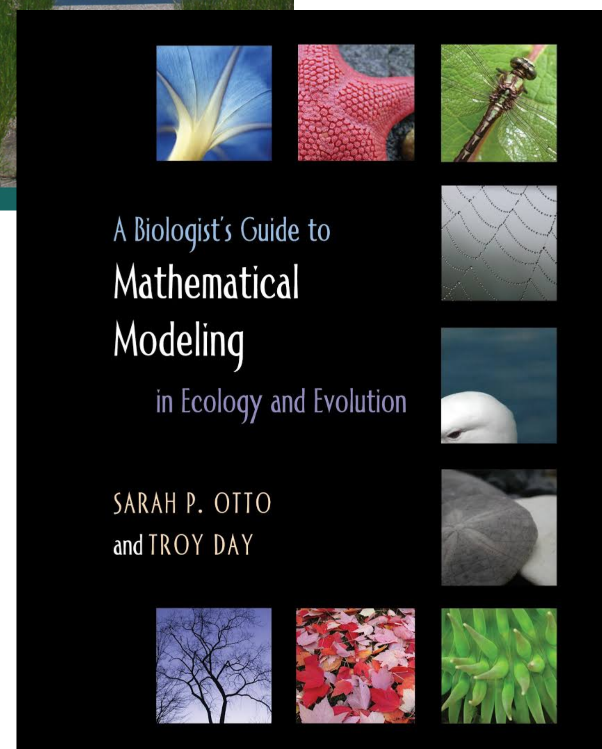
6. (Sensitivity analysis: FME package)




Modeling Infectious Diseases
IN
HUMANS
AND
ANIMALS
Matt J. Keeling
and
Pejman Rohani


EBC
ECOLOGY, BIODIVERSITY AND CONSERVATION
Parasites in
Ecological
Communities
From Interactions to Ecosystems
Melanie J. Hatcher
and Alison M. Dunn
CAMBRIDGE


Karline Soetaert
Peter M. J. Herman
A Practical Guide
to Ecological
Modelling
Using R as a Simulation Platform
Springer


A Biologist's Guide to
Mathematical
Modeling
in Ecology and Evolution
SARAH P. OTTO
and TROY DAY