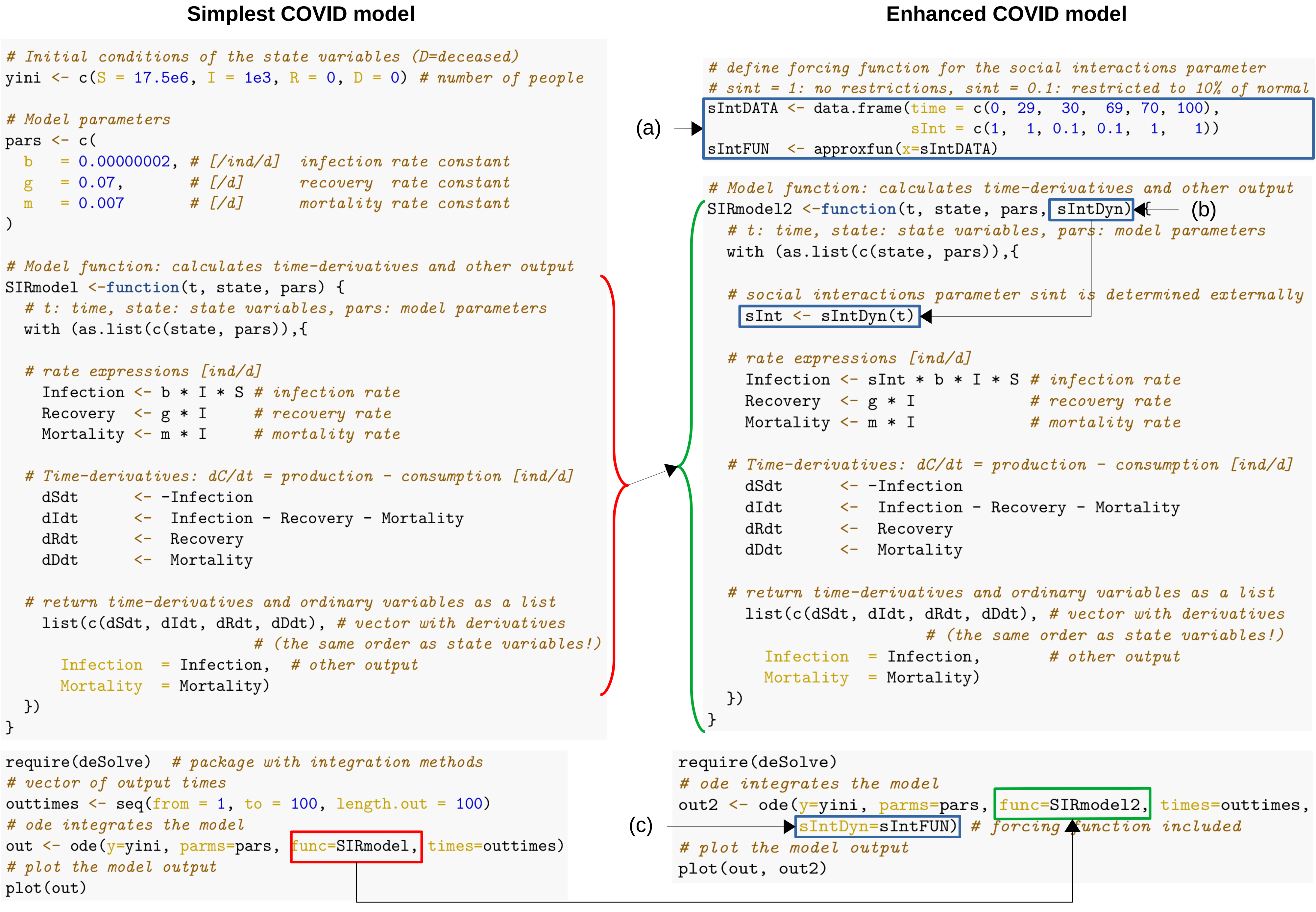

Simplest COVID model

```
# Initial conditions of the state variables (D=deceased)
yini <- c(S = 17.5e6, I = 1e3, R = 0, D = 0) # number of people

# Model parameters
pars <- c(
  b   = 0.00000002, # [/ind/d]  infection rate constant
  g   = 0.07,       # [/d]      recovery  rate constant
  m   = 0.007       # [/d]      mortality rate constant
)

# Model function: calculates time-derivatives and other output
SIRmodel <-function(t, state, pars) {
  # t: time, state: state variables, pars: model parameters
  with (as.list(c(state, pars)),{

  # rate expressions [ind/d]
    Infection <- b * I * S # infection rate
    Recovery  <- g * I     # recovery rate
    Mortality <- m * I     # mortality rate

  # Time-derivatives: dC/dt = production - consumption [ind/d]
    dSdt      <- -Infection
    dIdt      <-  Infection - Recovery - Mortality
    dRdt      <-  Recovery
    dDdt      <-  Mortality

  # return time-derivatives and ordinary variables as a list
    list(c(dSdt, dIdt, dRdt, dDdt), # vector with derivatives
                           # (the same order as state variables!)
      Infection  = Infection,  # other output
      Mortality  = Mortality)
  })
}
```

```
require(deSolve)   # package with integration methods
# vector of output times
outtimes <- seq(from = 1, to = 100, length.out = 100)
# ode integrates the model
out <- ode(y=yini, parms=pars, func=SIRmodel, times=outtimes)
# plot the model output
plot(out)
```

Enhanced COVID model

```
# define forcing function for the social interactions parameter
# sint = 1: no restrictions, sint = 0.1: restricted to 10% of normal
sIntDATA <- data.frame(time = c(0, 29,  30,  69, 70, 100),
                       sInt = c(1,  1, 0.1, 0.1,  1,   1))
sIntFUN  <- approxfun(x=sIntDATA)
```

```
# Model function: calculates time-derivatives and other output
SIRmodel2 <-function(t, state, pars, sIntDyn){
  # t: time, state: state variables, pars: model parameters
  with (as.list(c(state, pars)),{

  # social interactions parameter sint is determined externally
    sInt <- sIntDyn(t)

  # rate expressions [ind/d]
    Infection <- sInt * b * I * S # infection rate
    Recovery  <- g * I            # recovery rate
    Mortality <- m * I            # mortality rate

  # Time-derivatives: dC/dt = production - consumption [ind/d]
    dSdt      <- -Infection
    dIdt      <-  Infection - Recovery - Mortality
    dRdt      <-  Recovery
    dDdt      <-  Mortality

  # return time-derivatives and ordinary variables as a list
    list(c(dSdt, dIdt, dRdt, dDdt), # vector with derivatives
                          # (the same order as state variables!)
      Infection  = Infection,       # other output
      Mortality  = Mortality)
  })
}
```

```
require(deSolve)
# ode integrates the model
out2 <- ode(y=yini, parms=pars, func=SIRmodel2, times=outtimes,
            sIntDyn=sIntFUN) # forcing function included
# plot the model output
plot(out, out2)
```