*########################################*

*### MIDAS-CCDD Outreach Conference ###*

*### March 4th 2024 ###*

*### Breakout session #1 : SIR Basics ###*

*### SOLUTIONS ###*

*########################################*

*### Install required package*

*# Install deSolve package*

*# You only need to do this once on any computer.*

*install.packages("deSolve", dep = TRUE)*

*# Load deSolve package (so we can use the lsoda() command below)*

*# You'll need to do this very time you open a new R session.*

*library(deSolve)*

*### Question 1*

*### Set time steps, initial values, and parameters based on information provided in the question*

*dt <- seq(0, 365, 1)*

*inits <- c(S= 999999, I= 1, R= 0)*

*parms <- c(b= 0.75, k= 12/7 , r= 1/7)*

*### Create an ODE model*

*SIR <- function(t, x, parms){*

 *with(as.list(c(parms,x)),{*

 *N <- S+I+R*

 *dS <- - (b\*k\*S\*I)/N*

 *dI <- + (b\*k\*S\*I)/N - r\*I*

 *dR <- r\*I*

 *der <- c(dS, dI,dR)*

 *list(der)*

 *})*

*}*

*### Run the ode model*

*simulation <- as.data.frame(lsoda(y = inits,*

 *times = dt,*

 *func = SIR,*

 *parms = parms))*

*## Plot results*

*matplot(x = simulation[,1], y = simulation[,2:4],*

 *type= "l", lty = 1,*

 *xlab = "Time", ylab = "People (count)",*

 *main = "Simulation results")*

*# Add a legend*

*legend(x = "right", legend = c('S', 'I', 'R'),*

 *col = 1:3, lty = 1)*

*### Question 2*

*### chickenpox*

*### Set time steps, initial values, and parameters based on information provided in the question*

*dt <- seq(0, 365, 1)*

*inits <- c(S= 999999, I= 1, R= 0)*

*parms <- c(b= 0.51, k= 12/7 , r= 1/7)*

*### Run the ode model*

*simulation <- as.data.frame(lsoda(y = inits,*

 *times = dt,*

 *func = SIR,*

 *parms = parms))*

*## Plot results*

*matplot(x = simulation[,1], y = simulation[,2:4],*

 *type= "l", lty = 1,*

 *xlab = "Time", ylab = "People (count)",*

 *main = "Simulation results")*

*# Add a legend*

*legend(x = "right", legend = c('S', 'I', 'R'),*

 *col = 1:3, lty = 1)*

*### mumps*

*### Set time steps, initial values, and parameters based on information provided in the question*

*dt <- seq(0, 365, 1)*

*inits <- c(S= 999999, I= 1, R= 0)*

*parms <- c(b= 0.39, k= 12/7 , r= 1/7)*

*### Run the ode model*

*simulation <- as.data.frame(lsoda(y = inits,*

 *times = dt,*

 *func = SIR,*

 *parms = parms))*

*## Plot results*

*matplot(x = simulation[,1], y = simulation[,2:4],*

 *type= "l", lty = 1,*

 *xlab = "Time", ylab = "People (count)",*

 *main = "Simulation results")*

*# Add a legend*

*legend(x = "right", legend = c('S', 'I', 'R'),*

 *col = 1:3, lty = 1)*

*### Question 3*

*### Set time steps, initial values, and parameters based on information provided in the question*

*dt <- seq(0, 365, 1)*

*inits <- c(S= 999999, I= 1, R= 0)*

*parms <- c(b= 0.75, k= 12/7 , r= 1/3)*

*### Run the ode model*

*simulation <- as.data.frame(lsoda(y = inits,*

 *times = dt,*

 *func = SIR,*

 *parms = parms))*

*## Plot results*

*matplot(x = simulation[,1], y = simulation[,2:4],*

 *type= "l", lty = 1,*

 *xlab = "Time", ylab = "People (count)",*

 *main = "Simulation results")*

*# Add a legend*

*legend(x = "right", legend = c('S', 'I', 'R'),*

 *col = 1:3, lty = 1)*

*### Question 4*

*### Set time steps, initial values, and parameters based on information provided in the question*

*dt <- seq(0, 365, 1)*

*inits <- c(S= (1000000-1), E= 0 , I= 1, R= 0) # E: new 'Exposed' compartment*

*parms <- c(b= 0.75, k= 12/7, a=1/12, r= 1/7) # a = latency period*

*### Create an ODE model*

*SEIR <- function(t, x, parms){ # do not change the order of these inputs*

 *# t is the vector of time-steps;*

 *# x is the current state of the model;*

 *# parms is the vector of parameters*

 *with(as.list(c(parms,x)),{ # "with" allows us to refer to parms and x by shorthand*

 *N <- S+E+I+R*

 *dS <- - (b\*k\*S\*I)/N*

 *dE <- + (b\*k\*S\*I)/N - a\*E*

 *dI <- + (a\*E) - r\*I*

 *dR <- r\*I*

 *der <- c(dS, dE, dI, dR)*

 *list(der) # the output must be returned as a list*

 *}) # end of 'with'*

*} # end of function definition*

*### Run the ode model*

*simulation <- as.data.frame(lsoda(y = inits,*

 *times = dt,*

 *func = SEIR,*

 *parms = parms))*

*## Plot results*

*matplot(x = simulation[,1], y = simulation[,2:5],*

 *type= "l", lty = 1,*

 *xlab = "Time", ylab = "People (count)",*

 *main = "Simulation results",*

 *col = c(1,4,2,3))*

*# Add a legend*

*legend(x = "right", legend = c('S', 'E', 'I', 'R'),*

 *col = c(1,4,2,3), lty = 1)*

*### Question 5a*

*### Set time steps, initial values, and parameters based on information provided in the question*

*dt <- seq(0, 365, 1)*

*inits <- c(S= (1000000-1), I= 1, R= 0)*

*parms <- c(b= 0.75, k= 12/7 , r= 1/7, birth = 0.013, death = 0.013) # Add birth and death rates*

*### Create an ODE model*

*SIR\_steadystate <- function(t, x, parms){*

 *with(as.list(c(parms,x)),{*

 *N <- S+I+R*

 *dS <- - (b\*k\*S\*I)/N + (birth\*N) - (death\*S)*

 *dI <- + (b\*k\*S\*I)/N - r\*I - (death\*I)*

 *dR <- r\*I - (death\*R)*

 *der <- c(dS, dI,dR)*

 *list(der)*

 *})*

*}*

*### Run the ode model*

*simulation <- as.data.frame(lsoda(y = inits,*

 *times = dt,*

 *func = SIR\_steadystate,*

 *parms = parms))*

*## Plot results*

*matplot(x = simulation[,1], y = simulation[,2:4],*

 *type= "l", lty = 1,*

 *xlab = "Time", ylab = "People (count)",*

 *main = "Simulation results")*

*# Add a legend*

*legend(x = "right", legend = c('S', 'I', 'R'),*

 *col = 1:3, lty = 1)*

*### Question 5b*

*### Set time steps, initial values, and parameters based on information provided in the question*

*dt <- seq(0, 365, 1)*

*inits <- c(S= (1000000-1), I= 1, R= 0)*

*parms <- c(b= 0.75, k= 12/7 , r= 1/7, birth = 0.025, death = 0.025) # Change birth and death rates*

*### Create an ODE model*

*SIR\_steadystate <- function(t, x, parms){*

 *with(as.list(c(parms,x)),{*

 *N <- S+I+R*

 *dS <- - (b\*k\*S\*I)/N + (birth\*N) - (death\*S)*

 *dI <- + (b\*k\*S\*I)/N - r\*I - (death\*I)*

 *dR <- r\*I - (death\*R)*

 *der <- c(dS, dI,dR)*

 *list(der)*

 *})*

*}*

*### Run the ode model*

*simulation <- as.data.frame(lsoda(y = inits,*

 *times = dt,*

 *func = SIR\_steadystate,*

 *parms = parms))*

*## Plot results*

*matplot(x = simulation[,1], y = simulation[,2:4],*

 *type= "l", lty = 1,*

 *xlab = "Time", ylab = "People (count)",*

 *main = "Simulation results")*

*# Add a legend*

*legend(x = "right", legend = c('S', 'I', 'R'),*

 *col = 1:3, lty = 1)*