Actividad 2: Modelo SIR

Arif Morán Velázquez/A01234442Mendoza Muraira/A01383399Fernando Sandoval Silva/A01742322

Lunes 21 de Noviembre del 2022

## El modelo SIR

Consideremos un modelo para describir la dinámica de un grupo de individuos de una población con exposición a una enfermedad que puede contagiarse entre los miembros de la población. Esto puede modelarse como un sistema dinámico denominado para una población de individuos en la que se considera la interacción entre un conjunto de individuos *suceptibles* de contraer la enfermedad, un conjunto de individuos *infectados* y uno conjunto de individuos *recuperados* de la enfermedad.

Este modelo tiene los siguientes supuestos:

* la probabilidades de infectarse son iguales para todos los individuos de la población;
* la población es homogénea, es decir que los riesgos de infectarse son iguales para toos los suceptibles y que los tiempos para recuperarse son iguales para todos los infectados; y
* el tamaño de la población es constante.

El modelo maneja los diferentes conjuntos , y como si fueran compartimentos bien separados y considera que los individuos pueden pasr de uno a otro en el caso de que se enfermen (cambio ) o que una vez enfermos se recuperen (cambio ). Ademas, se asume que un individuo no puede pasar del conjunto de suceptibles directamente al conjunto de recuperados.

Con estos supuestos y consideraciones, las ecuaciones diferenciales del modelo SIR son:

donde:

* la cantidad representa la razón con que las personas salen del compartimento S (se infectan);
* en la primera ecuación representa el cambio debido a las personas que salen del compartimento (el signo negativo se debe a que las personas salen)
* en la segunda ecuación representa el cambio debido a las personas que salen del compartimento (una parte se debe a las personas que del compartimento pasan al compartimento , y otra parte se debe a las personas que salen del compartimento porque se recuperan);
* la cantidad representa la razón con que las personas se recuperan.

# PACKAGES:  
library(deSolve)  
library(reshape2)  
#library(ggplot2)  
  
mb=function(beta,gamma,t,v){  
initial\_state\_values <- c(S = 999999, # Número de susceptibles inicial  
 #   
 I = 1, # Se inicia con una persona infectada  
 R = 0) #   
  
  
#razones en unidades de días^-1  
parameters <- c(beta,gamma) # razón de recuperación  
  
#valores de tiempo para resolver la ecuación, de 0 a 60 días  
times <- seq(from = 0, to = t, by = 0.01)   
  
sir\_model <- function(time, state, parameters) {   
 with(as.list(c(state, parameters)), {# R obtendrá los nombres de variables a  
 # partir de inputs de estados y parametros  
 N <- S+I+R   
 lambda <- beta \* I/N  
 dS <- -lambda \* S-v\*S   
 dI <- lambda \* S - gamma \* I   
 dR <- gamma \* I+v\*S   
 return(list(c(dS, dI, dR)))   
 })  
}  
  
# poner la solución del sistema de ecuaciones en forma de un dataframe  
output <- as.data.frame(ode(y = initial\_state\_values,   
 times = times,   
 func = sir\_model,  
 parms = parameters))  
return(output)  
}

**Segundo Modelo**

# PACKAGES:  
library(deSolve)  
library(reshape2)  
#library(ggplot2)  
  
hh=function(beta,gamma,b,mu,t,v){  
initial\_state\_values <- c(S = 999999, # Número de susceptibles inicial  
 #   
 I = 1, # Se inicia con una persona infectada  
 R = 0) #   
  
  
#razones en unidades de días^-1  
parameters <- c(beta,gamma,b,mu,v) # razón de recuperación  
  
#valores de tiempo para resolver la ecuación, de 0 a 60 días  
times <- seq(from = 0, to = t, by = 0.001)   
  
sir\_model <- function(time, state, parameters) {   
 with(as.list(c(state, parameters)), {# R obtendrá los nombres de variables a  
 # partir de inputs de estados y parametros  
 N <- S+I+R   
 lambda <- beta \* I/N  
 dS <- -lambda \* S+b\*N-mu\*S-v\*S   
 dI <- lambda \* S - gamma \* I-mu\*I   
 dR <- gamma \* I-mu\*R+v\*S   
 return(list(c(dS, dI, dR)))   
 })  
}  
  
# poner la solución del sistema de ecuaciones en forma de un dataframe  
output <- as.data.frame(ode(y = initial\_state\_values,   
 times = times,   
 func = sir\_model,  
 parms = parameters))  
return(output)  
}

## Gráficos de la evolución del sistema

#```{r } output\_long <- melt(as.data.frame(output), id = “time”)

ggplot(data = output\_long,  
aes(x = time, y = value, colour = variable, group = variable)) +  
geom\_line() +  
xlab(“Tiempo (días)”)+  
ylab(“Número de individuos”) +  
labs(colour = “Subconjunto”) + theme(legend.position = “bottom”) #```

Con el modelo SIR se define la constante

que representa el número de personas que cada contagiado infecta. Para que la enfermedad analizada logre dispararse en forma de una epidemia debe cumplirse que .

También se define

que corresponde al número promedio de personas que cada contagiado infecta. Este segundo valor toma en cuenta de que durante la evolución de la pandemia, al aumentar del número de personas inmunes en la población cada persona contagiada infectará a un número de personas cada vez menor.

## Pregunta 1

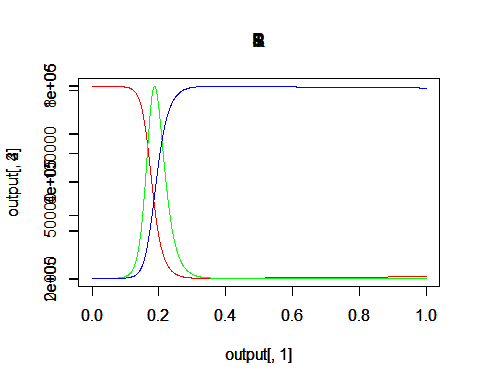
Haga cambios en el modelo para tomar en cuenta el hecho de que la población no es constante:

* agregar un término de incremento en para tomar en cuenta los individuos nacidos
* agregar un término de decremento en para tomar en cuenta las personas susceptibles que mueren -
* agregar un término de decremento en para tomar en cuenta las personas infectadas que mueren -
* agregar un término de decremento en para tomar en cuenta las personas recuperadas que fallecen

Usar ahora los paámetros

y considerar una duración de 1 año.

output=hh(beta = 0.4\*365,gamma = 0.2\*365,b=1/70,mu=1/70,t=1,v=0)  
####GGplot##############################################################  
plot(output[,1],output[,2],type="l",main="S",col="red",lwd=1.5)  
par(new=TRUE)  
plot(output[,1],output[,3],type="l",main="I",col="green",lwd=1.5)  
par(new=TRUE)  
plot(output[,1],output[,4],type="l",main="R",col="blue",lwd=1.5)



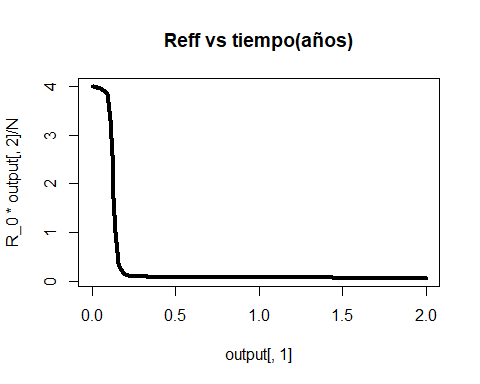
## Pregunta 2

Considerando el modelo SIR básico, haga cambios para tomar en cuenta un programa de vacunación. Suponga que una fracción de susceptibles se vacuna de manera que queda inmune (y entra ahora directamente en el conjunto de los recuperados). Calcule la dinámica de la epidemia en este caso usando los parámetros , y considere un periodo de 2 años.

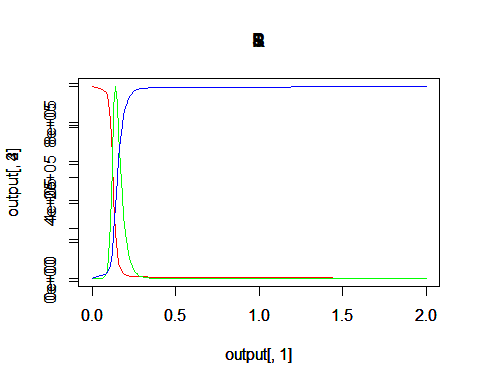
Su modelo debe ser capaz de mostrar que si la fracción es suficiente, no es necesario vacunar a todos los suceptibles para evitar la epidemia. A este efecto se le conoce como *inmunidad de rebaño* y se refiere a que si un sector grande de la población es inmune, entonces los contagios se mantienen a un nivel en el que la enfermedad es eliminada.

¿Cómo se puede calcular la fracción mínima de personas que se deben vacunar para poder evitar una epidemia? La inmunidad de rebaño ocurre cuando .

g=0.1  
b=0.4  
R\_0 =b/g  
  
output=mb(beta = b\*365,gamma = g\*365,t=2,v=1/4)  
##Reff  
N=output[,2]+output[,3]+output[,4]  
plot(output[,1],R\_0\*output[,2]/N,main="Reff vs tiempo(años)",type="l",col="black",lwd="4")



###GGplot#################################################  
plot(output[,1],output[,2],type="l",main="S",col="red",lwd=1.5)  
par(new=TRUE)  
plot(output[,1],output[,3],type="l",main="I",col="green",lwd=1.5)  
par(new=TRUE)  
plot(output[,1],output[,4],type="l",main="R",col="blue",lwd=1.5)



## Pregunta 3

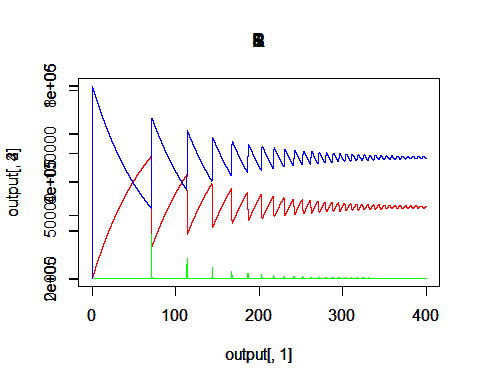
Haga cambios en el modelo para tomar en cuenta de que la población no es constante:

* agregar un término de incremento en para tomar en cuenta los nacidos
* agregar un término de decremento en para tomar en cuenta las personas susceptibles que mueren -
* agregar un término de decremento en para tomar en cuenta las personas infectadas que mueren -
* agregar un término de decremento en para tomar en cuenta las personas recuperadas que fallecen

Use los parámetros

y considere una duración de 400 años en sus cálculos.

output=hh(beta = 0.4\*365,gamma = 0.2\*365,b=1/70,mu=1/70,t=400,v=0)  
plot(output[,1],output[,2],type="l",main="S",col="red",lwd=1.5)  
par(new=TRUE)  
plot(output[,1],output[,3],type="l",main="I",col="green",lwd=1.5)  
par(new=TRUE)  
plot(output[,1],output[,4],type="l",main="R",col="blue",lwd=1.5)



## Pregunta 4

Considerando el modelo SIR básico, haga cambios para tomar en cuenta un programa de vacunación. Suponga que una fracción de susceptibles se vacuna de manera que queda inmune (y entra ahora directamente en el conjunto de los recuperados), mientras que la fracción sigue siendo susceptible.

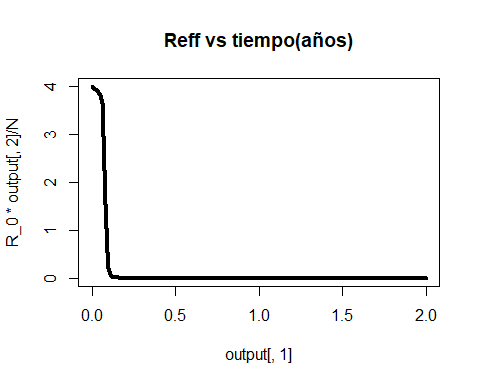
Calcule la dinámica de la epidemia en este caso, estudiando cómo cambia la dinámica variando la fracción . Utilice , y considere un periodo de 2 años.

Su modelo debe ser capaz de mostrar que si la fracción es suficiente, no es necesario vacunar a todos los suceptibles para evitar la epidemia. A este efecto se le conoce como *inmunidad de rebaño* y se refiere a que si un sector grande de la población es inmune, entonces los contagios se mantienen a un nivel en el que la enfermedad es eliminada.

¿Cómo se puede calcular la fracción mínima de personas que se deben vacunar para poder evitar una epidemia? La inmunidad de rebaño ocurre cuando .

# PACKAGES:  
library(deSolve)  
library(reshape2)  
#library(ggplot2)  
  
m3=function(beta,gamma,t,v){  
initial\_state\_values <- c(S = 999999, # Número de susceptibles inicial  
 #   
 I = 1, # Se inicia con una persona infectada  
 R = 0) #   
  
  
#razones en unidades de días^-1  
parameters <- c(beta,gamma) # razón de recuperación  
  
#valores de tiempo para resolver la ecuación, de 0 a t años  
times <- seq(from = 0, to = t, by = 0.01)   
  
sir\_model <- function(time, state, parameters) {   
 with(as.list(c(state, parameters)), {# R obtendrá los nombres de variables a  
 # partir de inputs de estados y parametros  
 N <- S+I+R   
 lambda <- beta \* I/N  
 dS <- -lambda \* S-(1-v)\*S   
 dI <- lambda \* S - gamma \* I   
 dR <- gamma \* I+v\*S   
 return(list(c(dS, dI, dR)))   
 })  
}  
  
# poner la solución del sistema de ecuaciones en forma de un dataframe  
output <- as.data.frame(ode(y = initial\_state\_values,   
 times = times,   
 func = sir\_model,  
 parms = parameters))  
return(output)  
}

output=m3(beta = 0.6\*365,gamma = 0.1\*365,t=2,v=3/4)  
###Reff  
N=output[,2]+output[,3]+output[,4]  
plot(output[,1],R\_0\*output[,2]/N,main="Reff vs tiempo(años)",type="l",col="black",lwd="4")



###GGplot###########################################################  
plot(output[,1],output[,2],type="l",main="S",col="red",lwd=1.5)  
par(new=TRUE)  
plot(output[,1],output[,3],type="l",main="I",col="green",lwd=1.5)  
par(new=TRUE)  
plot(output[,1],output[,4],type="l",main="R",col="blue",lwd=1.5)

