Modelo SEIRS (proyecto)

Glorivette Gonzalez

30/11/2021

## Modelo SEIRS

### Sistema de Ecuaciones Diferenciales en una Función R

Función en R en la cual están incluidos: tiempo, (cuando hay estacionalidad), vector de las variables de estado (), los parámetros del modelo ().

seirsmod=function(t, y, parms){  
 #Pull state variables from y vector  
 S=y[1]  
 E=y[2]  
 I=y[3]  
 R=y[4]  
 #Pull parameter values from parms vector  
 alpha= parms["alpha"]  
 beta=parms["beta"]  
 gamma=parms["gamma"]  
 mu= parms["mu"]  
 N=parms["N"]  
 ome= parms["ome"]  
 sigma= parms["sigma"]  
 #Define equations  
 dS = mu \* N - beta \* S \* I / N + ome \* R - mu \* S  
 dE = beta \* S \* I / N - sigma \* E - mu \* E  
 dI = sigma \* E - gamma \* I - (mu + alpha) \* I  
 dR = gamma \* I - ome \* R - mu \* R  
 res=c(dS, dE, dI, dR)  
 #Return list of gradients  
 list(res)  
 }

### Parámetros del Modelo y Condiciones Iniciales

times = seq(0, 300, by=1)  
 parms = c(alpha = 0, beta = 0.21, gamma = 0.07, mu = 0.000003, N = 1,   
 ome = 0.003, sigma = 0.14)  
 start = c(S = 0.999, E = 0.001, I = 0, R = 0)

### Aplicación de la función ode para obtener los valores de las variables de estado

library(deSolve)  
out = ode(y = start, times = times, func = seirsmod,   
 parms = parms)  
out=as.data.frame(out)  
head(round(out, 3))

## time S E I R  
## 1 0 0.999 0.001 0 0  
## 2 1 0.999 0.001 0 0  
## 3 2 0.999 0.001 0 0  
## 4 3 0.999 0.001 0 0  
## 5 4 0.999 0.001 0 0  
## 6 5 0.999 0.001 0 0

tail(round(out, 3))

## time S E I R  
## 296 295 0.342 0.003 0.006 0.649  
## 297 296 0.343 0.003 0.006 0.648  
## 298 297 0.345 0.003 0.006 0.646  
## 299 298 0.346 0.003 0.006 0.645  
## 300 299 0.348 0.003 0.006 0.643  
## 301 300 0.349 0.003 0.006 0.642

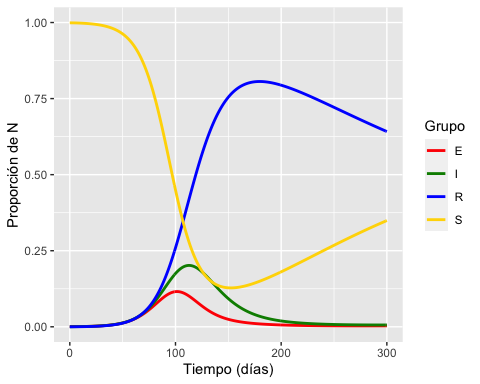
### Rearreglo de los resultados para gráficas y otros análisis

library(tidyverse)  
new.out <- as.data.frame(out) %>% gather(key, value, -time)   
head(new.out)

## time key value  
## 1 0 S 0.9990000  
## 2 1 S 0.9989864  
## 3 2 S 0.9989486  
## 4 3 S 0.9988903  
## 5 4 S 0.9988148  
## 6 5 S 0.9987240

### Gráfica de las variables de estado

library(ggplot2)  
ggplot(data = new.out,   
 aes(x = time,   
 y = value,   
 group = key,   
 col = key)) +   
 ylab("Proporción de N") +   
 xlab("Tiempo (días)") +  
 geom\_line(size = 1) +   
 scale\_colour\_manual(values = c("red", "green4","blue", "gold"),  
 name = "Grupo") +  
 scale\_y\_continuous(labels = waiver(), limits = c(0, 1))



### Cálculo del valor máximo de infectados, cuando ocurren y el valor umbral de S

# valor máximo de I y tiempo usando un pipe  
new.out %>%   
 filter(key=="I") %>%   
 filter(value==max(value)) %>%   
 mutate(maxI = round(value, 2)) %>%  
 select(time, maxI)

## time maxI  
## 1 113 0.2

# valor máximo de E y tiempo usando un pipe  
new.out%>%   
 filter(key=="E") %>%   
 filter(value==max(value)) %>%   
 mutate(maxE = round(value, 2)) %>%  
 select(time, maxE)

## time maxE  
## 1 101 0.12

# valor de S\_umbral  
out$S[which.max(out$I)]

## [1] 0.2761707

## EJERCICIO 1

(Copiar los códigos anteriores abajo para no perder el original) *Examinar los resultados al cambiar los parámetros y valores iniciales y relacionarlos a escenarios epidemiológicos hipotéticos.*

* cambiar el valor de S: ¿qué ocurre cuando se disminuye? ¿cuándo tiene un valor menor del S umbral?

#parametros   
times = seq(0, 300, by=1)  
 parms = c(N = 1, beta = 0.21, gamma = 0.07, mu = 0.000003, ome = 0.003,   
 alpha = 0, sigma = 0.14)  
 start = c(S = 0.799, E = 0.001, I = 0, R = 0.2)  
   
#ode  
 out = ode(y = start, times = times, func = seirsmod,   
 parms = parms)  
out=as.data.frame(out)  
head(round(out, 3))

## time S E I R  
## 1 0 0.799 0.001 0 0.200  
## 2 1 0.800 0.001 0 0.199  
## 3 2 0.800 0.001 0 0.199  
## 4 3 0.801 0.001 0 0.198  
## 5 4 0.801 0.001 0 0.198  
## 6 5 0.802 0.001 0 0.197

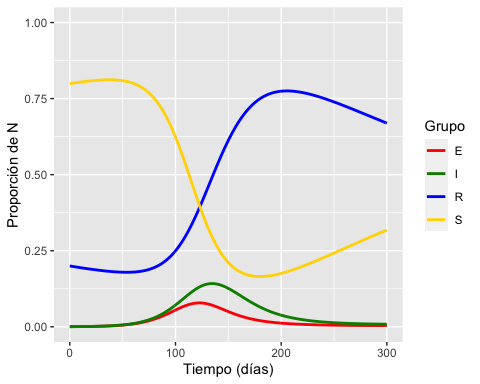
tail(round(out, 3))

## time S E I R  
## 296 295 0.311 0.004 0.009 0.676  
## 297 296 0.312 0.004 0.009 0.675  
## 298 297 0.314 0.004 0.009 0.674  
## 299 298 0.315 0.004 0.009 0.672  
## 300 299 0.317 0.004 0.009 0.671  
## 301 300 0.318 0.004 0.009 0.669

#rearreglo  
new.out <- as.data.frame(out) %>% gather(key, value, -time)   
head(new.out)

## time key value  
## 1 0 S 0.7990000  
## 2 1 S 0.7995889  
## 3 2 S 0.8001566  
## 4 3 S 0.8007063  
## 5 4 S 0.8012408  
## 6 5 S 0.8017619

#Grafica  
ggplot(data = new.out,   
 aes(x = time,  
 y = value,  
 group = key,  
 col = key  
 )) +   
 ylab("Proporción de N") + xlab("Tiempo (días)") +  
 geom\_line(size = 1) +   
 scale\_colour\_manual(values = c("red", "green4", "blue", "gold"), name = "Grupo") +  
 scale\_y\_continuous(labels = waiver(), limits = c(0, 1))



# valor máximo de I y tiempo usando un pipe  
new.out %>%   
 filter(key=="I") %>%   
 filter(value==max(value)) %>%   
 mutate(maxI = round(value, 2)) %>%  
 select(time, maxI)

## time maxI  
## 1 135 0.14

# valor máximo de E y tiempo usando un pipe  
new.out %>%   
 filter(key=="E") %>%   
 filter(value==max(value)) %>%   
 mutate(maxE = round(value, 2)) %>%  
 select(time, maxE)

## time maxE  
## 1 122 0.08

# valor de S\_umbral  
out$S[which.max(out$I)]

## [1] 0.292912

* aumentar beta

#parametros   
times = seq(0, 300, by=1)  
 parms = c(N = 1, beta = 0.30, gamma = 0.07, mu = 0.000003, ome = 0.003,   
 alpha = 0, sigma = 0.14)  
 start = c(S = 0.999, E = 0.001, I = 0, R = 0)  
   
#ode  
 out = ode(y = start, times = times, func = seirsmod,   
 parms = parms)  
out=as.data.frame(out)  
head(round(out, 3))

## time S E I R  
## 1 0 0.999 0.001 0 0  
## 2 1 0.999 0.001 0 0  
## 3 2 0.999 0.001 0 0  
## 4 3 0.999 0.001 0 0  
## 5 4 0.999 0.001 0 0  
## 6 5 0.999 0.001 0 0

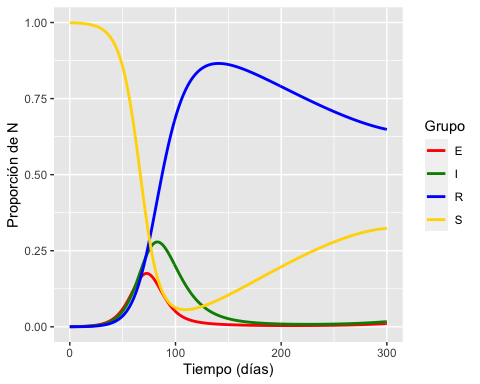
tail(round(out, 3))

## time S E I R  
## 296 295 0.322 0.010 0.016 0.653  
## 297 296 0.322 0.010 0.016 0.652  
## 298 297 0.323 0.010 0.016 0.651  
## 299 298 0.323 0.010 0.017 0.650  
## 300 299 0.323 0.010 0.017 0.649  
## 301 300 0.324 0.011 0.017 0.649

#rearreglo  
new.out <- as.data.frame(out) %>% gather(key, value, -time)   
head(new.out)

## time key value  
## 1 0 S 0.9990000  
## 2 1 S 0.9989806  
## 3 2 S 0.9989262  
## 4 3 S 0.9988418  
## 5 4 S 0.9987311  
## 6 5 S 0.9985958

#Grafica  
ggplot(data = new.out,   
 aes(x = time,  
 y = value,  
 group = key,  
 col = key  
 )) +   
 ylab("Proporción de N") + xlab("Tiempo (días)") +  
 geom\_line(size = 1) +   
 scale\_colour\_manual(values = c("red", "green4", "blue", "gold"), name = "Grupo") +  
 scale\_y\_continuous(labels = waiver(), limits = c(0, 1))



# valor máximo de I y tiempo usando un pipe  
new.out %>%   
 filter(key=="I") %>%   
 filter(value==max(value)) %>%   
 mutate(maxI = round(value, 2)) %>%  
 select(time, maxI)

## time maxI  
## 1 83 0.28

# valor máximo de E y tiempo usando un pipe  
new.out%>%   
 filter(key=="E") %>%   
 filter(value==max(value)) %>%   
 mutate(maxE = round(value, 2)) %>%  
 select(time, maxE)

## time maxE  
## 1 73 0.18

# valor de S\_umbral  
out$S[which.max(out$I)]

## [1] 0.1603834

* disminuir beta

#parametros   
times = seq(0, 300, by=1)  
 parms = c(N = 1, beta = 0.11, gamma = 0.07, mu = 0.000003, ome = 0.003,   
 alpha = 0, sigma = 0.14)  
 start = c(S = 0.999, E = 0.001, I = 0, R = 0)  
   
#ode  
 out = ode(y = start, times = times, func = seirsmod,   
 parms = parms)  
out=as.data.frame(out)  
head(round(out, 3))

## time S E I R  
## 1 0 0.999 0.001 0 0  
## 2 1 0.999 0.001 0 0  
## 3 2 0.999 0.001 0 0  
## 4 3 0.999 0.001 0 0  
## 5 4 0.999 0.001 0 0  
## 6 5 0.999 0.001 0 0

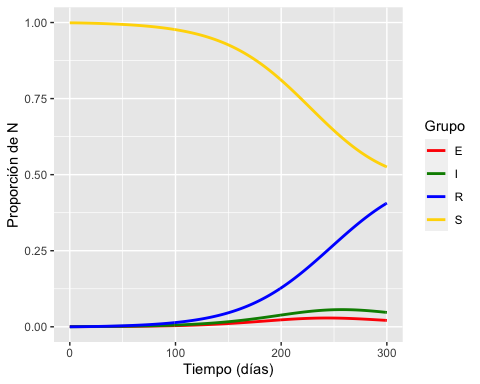
tail(round(out, 3))

## time S E I R  
## 296 295 0.533 0.022 0.049 0.396  
## 297 296 0.532 0.022 0.049 0.398  
## 298 297 0.530 0.022 0.048 0.400  
## 299 298 0.528 0.021 0.048 0.402  
## 300 299 0.527 0.021 0.048 0.404  
## 301 300 0.525 0.021 0.047 0.406

#rearreglo  
new.out <- as.data.frame(out) %>% gather(key, value, -time)   
head(new.out)

## time key value  
## 1 0 S 0.9990000  
## 2 1 S 0.9989929  
## 3 2 S 0.9989733  
## 4 3 S 0.9989433  
## 5 4 S 0.9989049  
## 6 5 S 0.9988595

#Grafica  
ggplot(data = new.out,   
 aes(x = time,  
 y = value,  
 group = key,  
 col = key  
 )) +   
 ylab("Proporción de N") + xlab("Tiempo (días)") +  
 geom\_line(size = 1) +   
 scale\_colour\_manual(values = c("red", "green4", "blue", "gold"), name = "Grupo") +  
 scale\_y\_continuous(labels = waiver(), limits = c(0, 1))



# valor máximo de I y tiempo usando un pipe  
new.out %>%   
 filter(key=="I") %>%   
 filter(value==max(value)) %>%   
 mutate(maxI = round(value, 2)) %>%  
 select(time, maxI)

## time maxI  
## 1 257 0.06

# valor máximo de E y tiempo usando un pipe  
new.out%>%   
 filter(key=="E") %>%   
 filter(value==max(value)) %>%   
 mutate(maxE = round(value, 2)) %>%  
 select(time, maxE)

## time maxE  
## 1 244 0.03

# valor de S\_umbral  
out$S[which.max(out$I)]

## [1] 0.6232986

* aumentar gamma

#parametros   
times = seq(0, 300, by=1)  
 parms = c(N = 1, beta = 0.21, gamma = 0.10, mu = 0.000003, ome = 0.003,   
 alpha = 0, sigma = 0.14)  
 start = c(S = 0.999, E = 0.001, I = 0, R = 0)  
   
#ode  
 out = ode(y = start, times = times, func = seirsmod,   
 parms = parms)  
out=as.data.frame(out)  
head(round(out, 3))

## time S E I R  
## 1 0 0.999 0.001 0 0  
## 2 1 0.999 0.001 0 0  
## 3 2 0.999 0.001 0 0  
## 4 3 0.999 0.001 0 0  
## 5 4 0.999 0.001 0 0  
## 6 5 0.999 0.001 0 0

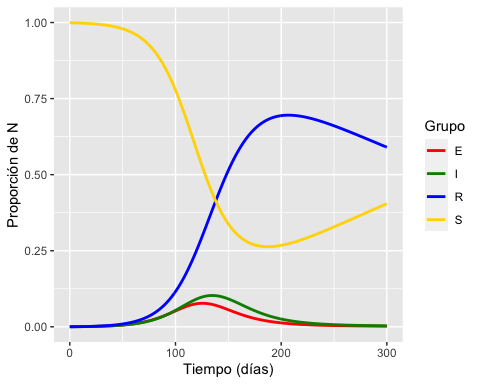
tail(round(out, 3))

## time S E I R  
## 296 295 0.397 0.002 0.003 0.597  
## 297 296 0.399 0.002 0.003 0.596  
## 298 297 0.400 0.002 0.003 0.594  
## 299 298 0.402 0.002 0.003 0.593  
## 300 299 0.403 0.002 0.003 0.592  
## 301 300 0.405 0.002 0.003 0.590

#rearreglo  
new.out <- as.data.frame(out) %>% gather(key, value, -time)   
head(new.out)

## time key value  
## 1 0 S 0.9990000  
## 2 1 S 0.9989866  
## 3 2 S 0.9989497  
## 4 3 S 0.9988936  
## 5 4 S 0.9988222  
## 6 5 S 0.9987376

#Grafica  
ggplot(data = new.out,   
 aes(x = time,  
 y = value,  
 group = key,  
 col = key  
 )) +   
 ylab("Proporción de N") + xlab("Tiempo (días)") +  
 geom\_line(size = 1) +   
 scale\_colour\_manual(values = c("red", "green4", "blue", "gold"), name = "Grupo") +  
 scale\_y\_continuous(labels = waiver(), limits = c(0, 1))



# valor máximo de I y tiempo usando un pipe  
new.out %>%   
 filter(key=="I") %>%   
 filter(value==max(value)) %>%   
 mutate(maxI = round(value, 2)) %>%  
 select(time, maxI)

## time maxI  
## 1 135 0.1

# valor máximo de E y tiempo usando un pipe  
new.out%>%   
 filter(key=="E") %>%   
 filter(value==max(value)) %>%   
 mutate(maxE = round(value, 2)) %>%  
 select(time, maxE)

## time maxE  
## 1 126 0.08

# valor de S\_umbral  
out$S[which.max(out$I)]

## [1] 0.4378933

* disminuir gamma

#parametros   
times = seq(0, 300, by=1)  
 parms = c(N = 1, beta = 0.21, gamma = 0.01, mu = 0.000003, ome = 0.003,   
 alpha = 0, sigma = 0.14)  
 start = c(S = 0.999, E = 0.001, I = 0, R = 0)  
   
#ode  
 out = ode(y = start, times = times, func = seirsmod,   
 parms = parms)  
out=as.data.frame(out)  
head(round(out, 3))

## time S E I R  
## 1 0 0.999 0.001 0.000 0  
## 2 1 0.999 0.001 0.000 0  
## 3 2 0.999 0.001 0.000 0  
## 4 3 0.999 0.001 0.000 0  
## 5 4 0.999 0.001 0.000 0  
## 6 5 0.999 0.001 0.001 0

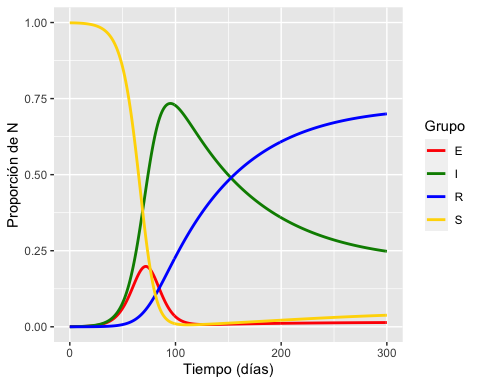
tail(round(out, 3))

## time S E I R  
## 296 295 0.037 0.014 0.251 0.698  
## 297 296 0.038 0.014 0.250 0.698  
## 298 297 0.038 0.014 0.250 0.699  
## 299 298 0.038 0.014 0.249 0.699  
## 300 299 0.038 0.014 0.249 0.699  
## 301 300 0.038 0.014 0.248 0.700

#rearreglo  
new.out <- as.data.frame(out) %>% gather(key, value, -time)   
head(new.out)

## time key value  
## 1 0 S 0.9990000  
## 2 1 S 0.9989861  
## 3 2 S 0.9989464  
## 4 3 S 0.9988833  
## 5 4 S 0.9987987  
## 6 5 S 0.9986934

#Grafica  
ggplot(data = new.out,   
 aes(x = time,  
 y = value,  
 group = key,  
 col = key  
 )) +   
 ylab("Proporción de N") + xlab("Tiempo (días)") +  
 geom\_line(size = 1) +   
 scale\_colour\_manual(values = c("red", "green4", "blue", "gold"), name = "Grupo") +  
 scale\_y\_continuous(labels = waiver(), limits = c(0, 1))



# valor máximo de I y tiempo usando un pipe  
new.out %>%   
 filter(key=="I") %>%   
 filter(value==max(value)) %>%   
 mutate(maxI = round(value, 2)) %>%  
 select(time, maxI)

## time maxI  
## 1 95 0.73

# valor máximo de E y tiempo usando un pipe  
new.out%>%   
 filter(key=="E") %>%   
 filter(value==max(value)) %>%   
 mutate(maxE = round(value, 2)) %>%  
 select(time, maxE)

## time maxE  
## 1 72 0.2

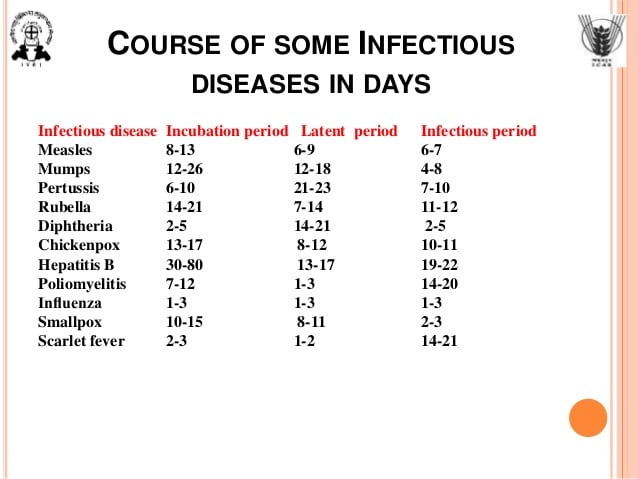
# valor de S\_umbral  
out$S[which.max(out$I)]

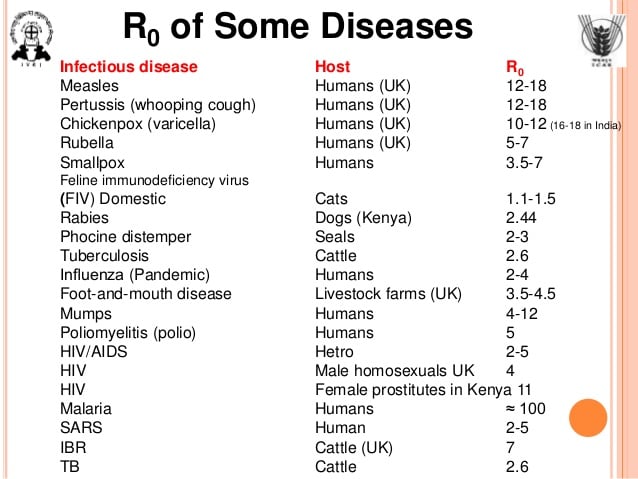
## [1] 0.01608351

## EJERCICIO 2

*Análisis de una enfermedad infecciosa a partir de datos de la literatura epidemiológica.*

A partir de los datos de las siguientes tablas (período infeccioso, período de latencia y ), realizar el análisis de una enfermedad infecciosa usando el modelo SEIRS.





#Influenza  
#parametros   
times = seq(0, 300, by=1)  
 parms = c(N = 1, beta = 1.5, gamma = 0.5, mu = 0.000003, ome = 0.003,   
 alpha = 0, sigma = 0.5)  
 start = c(S = 0.999, E = 0.001, I = 0, R = 0)  
   
#ode  
 out = ode(y = start, times = times, func = seirsmod,   
 parms = parms)  
out=as.data.frame(out)  
head(round(out, 3))

## time S E I R  
## 1 0 0.999 0.001 0.000 0.000  
## 2 1 0.999 0.001 0.000 0.000  
## 3 2 0.998 0.001 0.001 0.000  
## 4 3 0.997 0.002 0.001 0.001  
## 5 4 0.995 0.002 0.001 0.001  
## 6 5 0.993 0.003 0.002 0.002

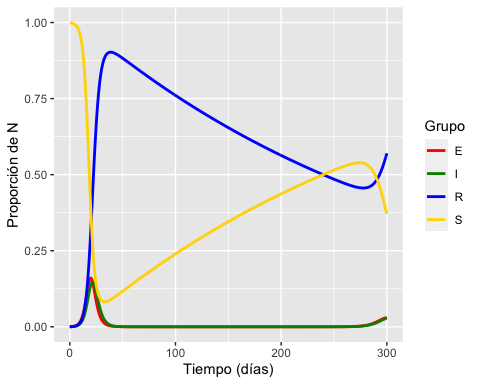
tail(round(out, 3))

## time S E I R  
## 296 295 0.439 0.025 0.021 0.516  
## 297 296 0.426 0.026 0.023 0.525  
## 298 297 0.413 0.028 0.024 0.535  
## 299 298 0.399 0.029 0.026 0.546  
## 300 299 0.385 0.030 0.027 0.558  
## 301 300 0.371 0.031 0.028 0.570

#rearreglo  
new.out <- as.data.frame(out) %>% gather(key, value, -time)   
head(new.out)

## time key value  
## 1 0 S 0.9990000  
## 2 1 S 0.9987135  
## 3 2 S 0.9980218  
## 4 3 S 0.9969550  
## 5 4 S 0.9954002  
## 6 5 S 0.9931627

#Grafica  
ggplot(data = new.out,   
 aes(x = time,  
 y = value,  
 group = key,  
 col = key  
 )) +   
 ylab("Proporción de N") + xlab("Tiempo (días)") +  
 geom\_line(size = 1) +   
 scale\_colour\_manual(values = c("red", "green4", "blue", "gold"), name = "Grupo") +  
 scale\_y\_continuous(labels = waiver(), limits = c(0, 1))



# valor máximo de I y tiempo usando un pipe  
new.out %>%   
 filter(key=="I") %>%   
 filter(value==max(value)) %>%   
 mutate(maxI = round(value, 2)) %>%  
 select(time, maxI)

## time maxI  
## 1 22 0.15

# valor máximo de E y tiempo usando un pipe  
new.out%>%   
 filter(key=="E") %>%   
 filter(value==max(value)) %>%   
 mutate(maxE = round(value, 2)) %>%  
 select(time, maxE)

## time maxE  
## 1 20 0.16

# valor de S\_umbral  
out$S[which.max(out$I)]

## [1] 0.2427958

#Viruela  
#parametros   
times = seq(0, 300, by=1)  
 parms = c(N = 1, beta = 2.1, gamma = 1/10, mu = 0.000003, ome = 0.003,   
 alpha = 0, sigma = 1/9.5)  
 start = c(S = 0.999, E = 0.001, I = 0, R = 0)  
   
#ode  
 out = ode(y = start, times = times, func = seirsmod,   
 parms = parms)  
out=as.data.frame(out)  
head(round(out, 3))

## time S E I R  
## 1 0 0.999 0.001 0.000 0  
## 2 1 0.999 0.001 0.000 0  
## 3 2 0.999 0.001 0.000 0  
## 4 3 0.998 0.002 0.000 0  
## 5 4 0.997 0.002 0.000 0  
## 6 5 0.996 0.003 0.001 0

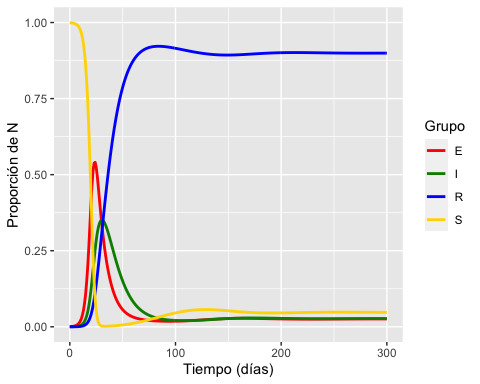
tail(round(out, 3))

## time S E I R  
## 296 295 0.048 0.026 0.027 0.899  
## 297 296 0.048 0.026 0.027 0.899  
## 298 297 0.048 0.026 0.027 0.899  
## 299 298 0.048 0.026 0.027 0.899  
## 300 299 0.048 0.026 0.027 0.899  
## 301 300 0.048 0.026 0.027 0.899

#rearreglo  
new.out <- as.data.frame(out) %>% gather(key, value, -time)   
head(new.out)

## time key value  
## 1 0 S 0.9990000  
## 2 1 S 0.9988949  
## 3 2 S 0.9985848  
## 4 3 S 0.9980485  
## 5 4 S 0.9972238  
## 6 5 S 0.9960057

#Grafica  
ggplot(data = new.out,   
 aes(x = time,  
 y = value,  
 group = key,  
 col = key  
 )) +   
 ylab("Proporción de N") + xlab("Tiempo (días)") +  
 geom\_line(size = 1) +   
 scale\_colour\_manual(values = c("red", "green4", "blue", "gold"), name = "Grupo") +  
 scale\_y\_continuous(labels = waiver(), limits = c(0, 1))



# valor máximo de I y tiempo usando un pipe  
new.out %>%   
 filter(key=="I") %>%   
 filter(value==max(value)) %>%   
 mutate(maxI = round(value, 2)) %>%  
 select(time, maxI)

## time maxI  
## 1 30 0.35

# valor máximo de E y tiempo usando un pipe  
new.out%>%   
 filter(key=="E") %>%   
 filter(value==max(value)) %>%   
 mutate(maxE = round(value, 2)) %>%  
 select(time, maxE)

## time maxE  
## 1 24 0.54

# valor de S\_umbral  
out$S[which.max(out$I)]

## [1] 0.00270895

#COVID-19  
seirbasic=function(t, y, parms){  
 #Pull state variables from y vector  
 S=y[1]  
 E=y[2]  
 I=y[3]  
 R=y[4]  
 #Pull parameter values from parms vector  
 beta=parms["beta"]  
 gamma=parms["gamma"]  
 N=parms["N"]  
 sigma= parms["sigma"]  
 #Define equations  
 dS = - beta \* S \* I / N   
 dE = beta \* S \* I / N - sigma \* E   
 dI = sigma \* E - gamma \* I  
 dR = gamma \* I  
 res=c(dS, dE, dI, dR)  
 #Return list of gradients  
 list(res)  
 }

#parametros   
times = seq(0, 120, by=1)  
 parms = c(N = 1, beta = 1, gamma = 1/5, sigma = 1/7)  
 start = c(S = 0.999, E = 0.001, I = 0, R = 0)  
   
#ode  
 out = ode(y = start, times = times, func = seirbasic,   
 parms = parms)  
out=as.data.frame(out)  
head(round(out, 3))

## time S E I R  
## 1 0 0.999 0.001 0.000 0  
## 2 1 0.999 0.001 0.000 0  
## 3 2 0.999 0.001 0.000 0  
## 4 3 0.998 0.001 0.000 0  
## 5 4 0.998 0.001 0.000 0  
## 6 5 0.998 0.002 0.001 0

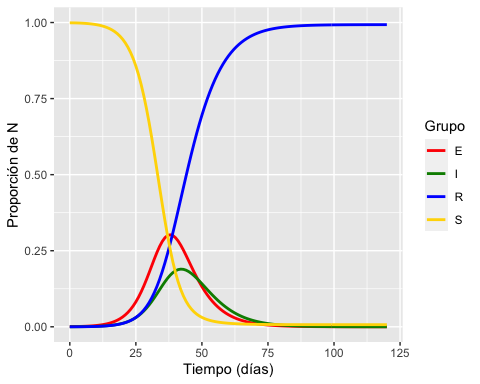
tail(round(out, 3))

## time S E I R  
## 116 115 0.007 0 0 0.993  
## 117 116 0.007 0 0 0.993  
## 118 117 0.007 0 0 0.993  
## 119 118 0.007 0 0 0.993  
## 120 119 0.007 0 0 0.993  
## 121 120 0.007 0 0 0.993

#rearreglo  
nuevo.fuera <- as.data.frame(out) %>% gather(key, value, -time)   
head(nuevo.fuera)

## time key value  
## 1 0 S 0.9990000  
## 2 1 S 0.9989358  
## 3 2 S 0.9987613  
## 4 3 S 0.9984927  
## 5 4 S 0.9981307  
## 6 5 S 0.9976665

#Grafica  
ggplot(data = nuevo.fuera,   
 aes(x = time,  
 y = value,  
 group = key,  
 col = key  
 )) +   
 ylab("Proporción de N") + xlab("Tiempo (días)") +  
 geom\_line(size = 1) +   
 scale\_colour\_manual(values = c("red", "green4", "blue", "gold"), name = "Grupo") +  
 scale\_y\_continuous(labels = waiver(), limits = c(0, 1))



# valor máximo de I y tiempo usando un pipe  
new.out %>%   
 filter(key=="I") %>%   
 filter(value==max(value)) %>%   
 mutate(maxI = round(value, 2)) %>%  
 select(time, maxI)

## time maxI  
## 1 30 0.35

# valor máximo de E y tiempo usando un pipe  
new.out%>%   
 filter(key=="E") %>%   
 filter(value==max(value)) %>%   
 mutate(maxE = round(value, 2)) %>%  
 select(time, maxE)

## time maxE  
## 1 24 0.54

# valor de S\_umbral  
out$S[which.max(out$I)]

## [1] 0.1223459

#COVID-19  
#parametros   
times = seq(0, 120, by=1)  
 parms = c(N = 1, beta = 1, gamma = 1/5, mu = 0.000003, ome = 0.003,   
 alpha = 0, sigma = 1/7)  
 start = c(S = 0.999, E = 0.001, I = 0, R = 0)  
   
#ode  
 out = ode(y = start, times = times, func = seirsmod,   
 parms = parms)  
out=as.data.frame(out)  
head(round(out, 3))

## time S E I R  
## 1 0 0.999 0.001 0.000 0  
## 2 1 0.999 0.001 0.000 0  
## 3 2 0.999 0.001 0.000 0  
## 4 3 0.998 0.001 0.000 0  
## 5 4 0.998 0.001 0.000 0  
## 6 5 0.998 0.002 0.001 0

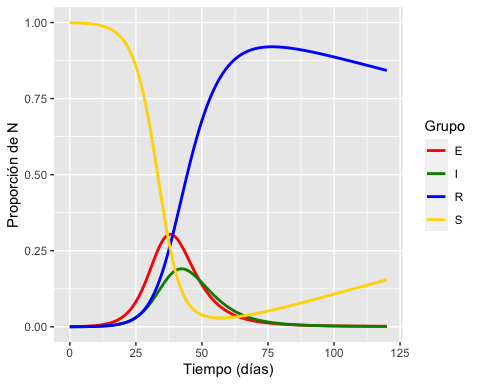
tail(round(out, 3))

## time S E I R  
## 116 115 0.143 0.002 0.002 0.854  
## 117 116 0.145 0.002 0.002 0.852  
## 118 117 0.148 0.002 0.001 0.849  
## 119 118 0.150 0.002 0.001 0.847  
## 120 119 0.152 0.002 0.001 0.845  
## 121 120 0.155 0.002 0.001 0.842

#rearreglo  
new.out <- as.data.frame(out) %>% gather(key, value, -time)   
head(new.out)

## time key value  
## 1 0 S 0.9990000  
## 2 1 S 0.9989358  
## 3 2 S 0.9987614  
## 4 3 S 0.9984931  
## 5 4 S 0.9981315  
## 6 5 S 0.9976680

#Grafica  
ggplot(data = new.out,   
 aes(x = time,  
 y = value,  
 group = key,  
 col = key  
 )) +   
 ylab("Proporción de N") + xlab("Tiempo (días)") +  
 geom\_line(size = 1) +   
 scale\_colour\_manual(values = c("red", "green4", "blue", "gold"), name = "Grupo") +  
 scale\_y\_continuous(labels = waiver(), limits = c(0, 1))



# valor máximo de I y tiempo usando un pipe  
new.out %>%   
 filter(key=="I") %>%   
 filter(value==max(value)) %>%   
 mutate(maxI = round(value, 2)) %>%  
 select(time, maxI)

## time maxI  
## 1 42 0.19

# valor máximo de E y tiempo usando un pipe  
new.out%>%   
 filter(key=="E") %>%   
 filter(value==max(value)) %>%   
 mutate(maxE = round(value, 2)) %>%  
 select(time, maxE)

## time maxE  
## 1 38 0.3

# valor de S\_umbral  
out$S[which.max(out$I)]

## [1] 0.1260304