Solución numérica de modelos epidemiológicos deterministas

1. Introducción

Muchas de las teorías centrales sobre la propagación de epidemias se expresan en un sistema de ecuaciones diferenciales ordinarias conocido como modelo compartimental. Esta sesión presenta técnicas para resolver numéricamente sistemas de ecuaciones diferenciales no lineales con un solucionador de tamaño de paso adaptativo.

2 El modelo SIR

Como presentamos en la conferencia, el modelo compartimental SIR clásico rastrea la fracción de la población en cada una de las tres clases (susceptible, infectada, recuperada). En un sistema demográficamente cerrado, el flujo que sale de una clase debe entrar en otra clase, dando lugar a una propiedad de conservación. Las variables de estado cambian según el siguiente sistema de ecuaciones diferenciales:

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

donde S, I y R son la proporción de individuos susceptibles, infectados y recuperados y γ es la tasa de recuperación.

En un sistema demográficamente abierto, el número de individuos de la población puede cambiar debido a los nacimientos y muertes que ocurren a tasas per cápita α y μ. el tenemos

$$\frac{dS}{dt} = \alpha - \beta SI - \mu S$$
$$\frac{dI}{dt} = \beta SI - \gamma I - \mu I$$
$$\frac{dR}{dt} = \gamma I - \mu R$$

Si establecemos α = μ entonces los nacimientos equilibran exactamente las muertes y la población permanece en un tamaño constante, lo que produce

$$\frac{dS}{dt} = \mu - \beta SI - \mu S$$
$$\frac{dI}{dt} = \beta SI - \gamma I - \mu I$$
$$\frac{dR}{dt} = \gamma I - \mu R$$
.

Como muchos modelos epidemiológicos, no se pueden resolver analíticamente las ecuaciones SIR. Más bien, para encontrar el trayectoria de un modelo de tiempo continuo como el SIR, integramos estas ecuaciones numéricamente. Qué Lo que queremos decir con esto es que utilizamos un algoritmo informático para aproximar la solución. En general, esto puede ser un asunto complicado. Afortunadamente, este es un problema bien estudiado en análisis numérico y (cuando el (las ecuaciones son funciones suaves y de buen comportamiento de un número relativamente pequeño de variables), se encuentran disponibles esquemas de integración numérica estándar para aproximar la integral con precisión arbitraria. Particularmente, R tiene un solucionador de ODE muy sofisticado, que (para muchos problemas) brindará soluciones muy precisas. Para utilizar el paquete de integración numérica, debemos cargar el paquete

```
> requerir(deResolver)                  Biblioteca #deSolve necesaria para esta sesión informática
```

El solucionador de ODE necesita conocer los lados derechos de la ODE. Le damos esta información como una función. (subrutina). Tenga en cuenta que la forma de los argumentos y la salida de esta función debe coincidir exactamente con lo que es lo que se espera de la rutina de la oda. Así, por ejemplo, la variable de tiempo t debe ser el primer argumento incluso si la función es autónoma o invariante en el tiempo, de modo que t se desprecie en el cálculo. aquí tenemos razón una función para devolver las derivadas del modelo SIR cerrado.

```
> señor.modelo.cerrado <- función (t, x, params) { + S <- x[1]    #aquí comenzamos una función con tres argumentos
                                                                  #crea la variable local S, el primer elemento de x
+ yo <- x[2]                                                      #crear variable local I
+ R <- x[3] +                                                     #crear variable local R
                                                                  #podemos simplificar el código usando "con"
+
con( as.list(parámetros), {                                       #este argumento para "con" nos permite usar los nombres de las variables
+                                                                 #el sistema de ecuaciones de tasas
+           dS <- -beta*S*I
+           dI <- beta*S*I-gamma*I
+           dR <- gamma*I
+           dx <- c(dS,dI,dR) lista(dx)                           #combinar resultados en un solo vector dx
+                                                                 #retornar resultado como una lista
+       }
+       )
+ }
```

Observe que aquí hemos asumido que β es constante.

[Nota: En caso de que la función with no sea familiar, sirve aquí para que los parámetros estén disponibles a las expresiones entre paréntesis, como si fueran variables. Se podría lograr el mismo efecto, por ejemplo por ejemplo, dS <- params["mu"]*(1-S)-params["beta"]*S*I y así sucesivamente.]

Ahora indicamos los momentos en los que queremos soluciones, asignamos algunos valores a los parámetros y especificamos las condiciones iniciales, es decir, los valores de las variables de estado S, I y R al comienzo de la simulación:

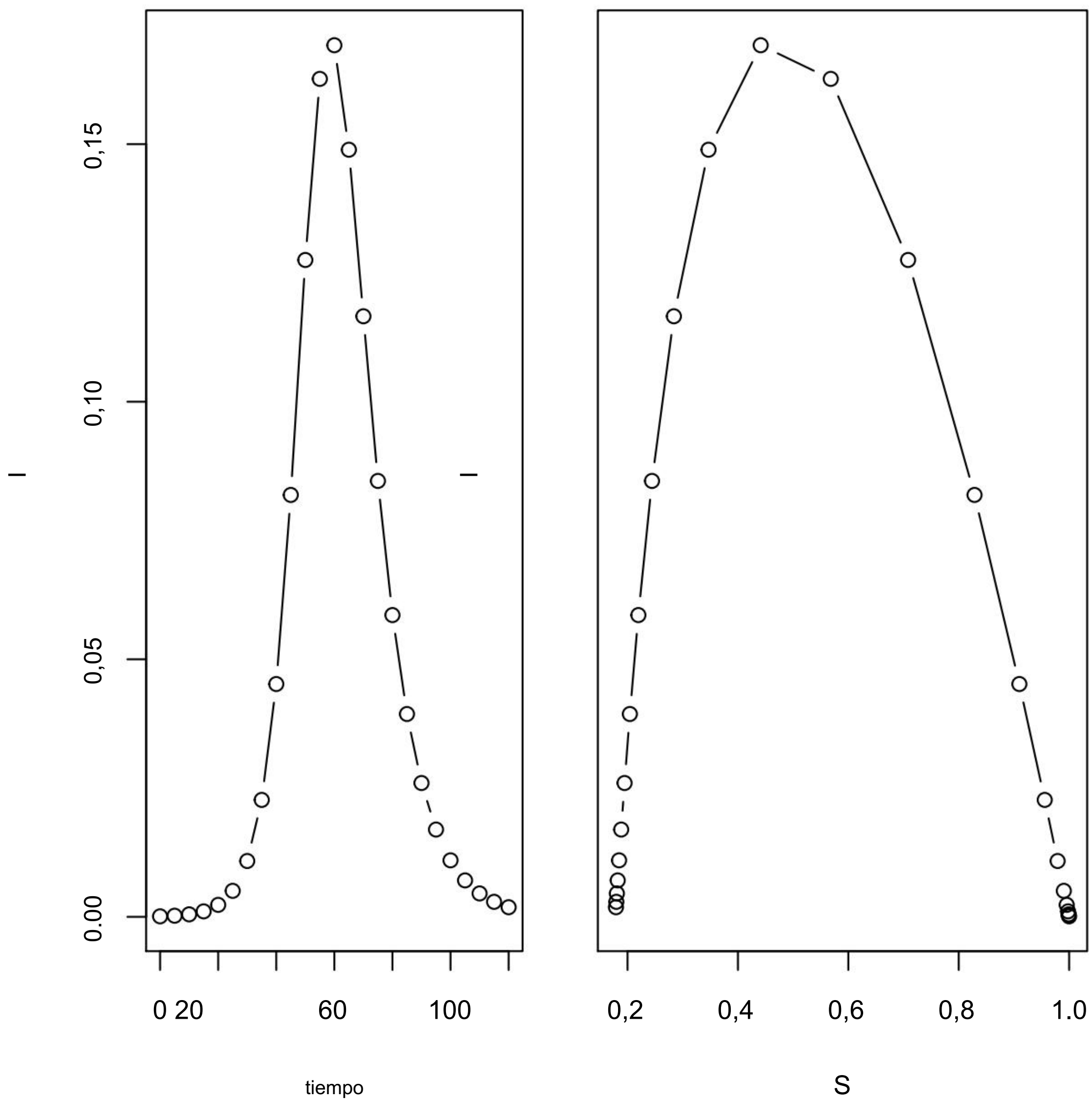
```
> veces <- seq(0,120,by=5) > params         #función seq devuelve una secuencia
<- c(beta=0.3,gamma=1/7) > xstart <-        #función c "c"combina valores en un vector
c(S=9999/10000,I=1/10000,R=0)               #condiciones iniciales
```

A continuación, simulamos la trayectoria de un modelo con el comando ode:

```
> out <- as.data.frame(ode(xstart,times,sir.model.closed,params)) #resultado almacenado en el dataframe
```

y trazar los resultados

```
> op <- par(fig=c(0,0.5,0,1),mar=c(4,4,1,1)) >          #establecer parámetros gráficos
plot(I~time,data=out,type='b') >                        #trazar la variable I contra el tiempo
par(fig=c(0.5,1,0,1),mar=c(4,1,1,1),new=T) >            #reestablecer parámetros gráficos
plot(I~S,data=out,type='b',yaxt ='n',xlab='S') > par(op) # retrato de la fase de la trama
                                                        #reestablecer parámetros gráficos
```



Ejercicio 1. Explore la dinámica del sistema para diferentes valores de los parámetros β y γ mediante simular y trazar trayectorias como series de tiempo y en espacio de fase (por ejemplo, I vs. S).

Ejercicio 2. Explora la dinámica del sistema para un conjunto de β y γ en diferentes condiciones iniciales.
¿Qué pasa si existe inmunidad preexistente en la población?

Ejercicio 3. Modificar los códigos dados para estudiar la dinámica de un modelo SIR demográficamente abierto.

*Ejercicio 4. Modificar los códigos dados para estudiar la dinámica de un modelo SEIR.