

Actividad 1: Modelo SIR

E. Crescio, J. Montalvo, E. Uresti

2024-10-09

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 SIR para una población de N individuos en la que se considera la interacción entre un conjunto de S individuos *susceptibles* de contraer la enfermedad, un conjunto I de individuos *infectados* y uno conjunto R de individuos *recuperados* de la enfermedad.

Este modelo tiene los siguientes supuestos:

- las 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 todos los susceptibles y que los tiempos para recuperarse son iguales para todos los infectados; y
- el tamaño N de la población es constante.

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

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

$$\begin{aligned}\frac{dS}{dt} &= -\beta \frac{I}{N} S \\ \frac{dI}{dt} &= \beta \frac{I}{N} S - \gamma I \\ \frac{dR}{dt} &= \gamma I\end{aligned}$$

donde:

- $N = S + R + I$
- la cantidad $\beta \frac{I}{N}$ representa la razón con que las personas salen del compartimento S (se infectan);
- en la primera ecuación dS representa el cambio debido a las personas que salen del compartimento S (el signo negativo se debe a que las personas salen)
- en la segunda ecuación dI representa el cambio debido a las personas que salen del compartimento I (una parte se debe a las personas que del compartimento S pasan al compartimento I , y otra parte se debe a las personas que salen del compartimento I porque se recuperan);
- la cantidad γ representa la razón con que las personas se recuperan.

```
# PACKAGES:
library(deSolve)
library(reshape2)
library(ggplot2)

# Initial states

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 = 1,           # razón de infección
                 gamma = 0.1)       # razón de recuperación

# valores de tiempo para resolver la ecuación, de 0 a 60 días
times <- seq(from = 0, to = 60, by = 1)

# Solución del modelo
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
    dI <- lambda * S - gamma * I
```

```

        dR <- gamma * I
        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))

```

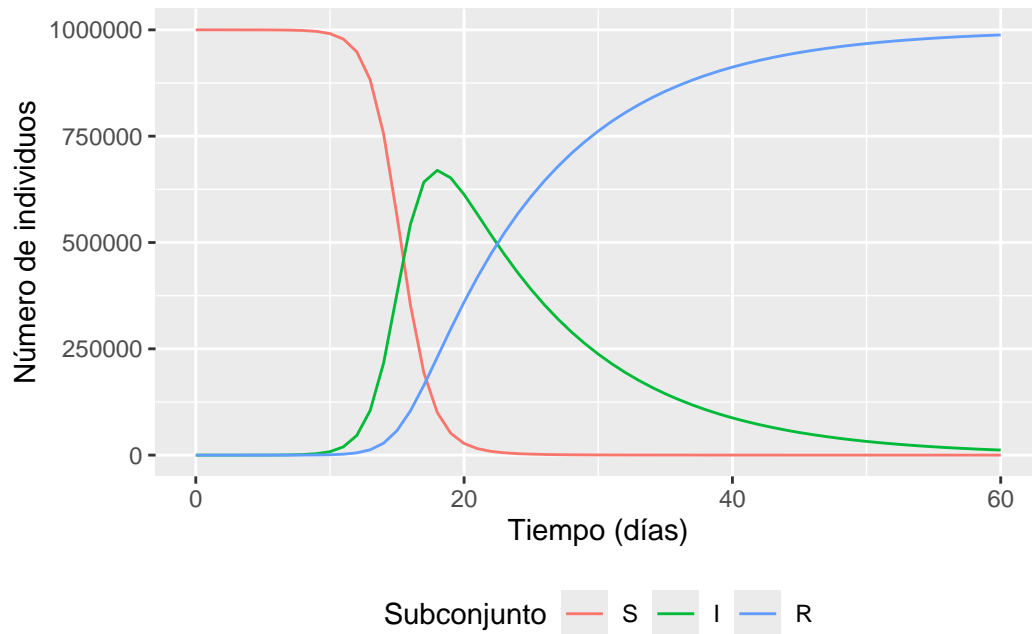
Gráficos de la evolución del sistema

```

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")

```



```
print(output)
```

	time	S	I	R
1	0	999999.00000	1.000000e+00	0.000000e+00
2	1	999997.37822	2.459606e+00	1.621788e-01
3	2	999993.38930	6.049632e+00	5.610726e-01
4	3	999983.57827	1.487954e+01	1.542186e+00
5	4	999959.44789	3.659682e+01	3.955293e+00
6	5	999900.10189	9.000780e+01	9.890310e+00
7	6	999754.16580	2.213478e+02	2.448644e+01
8	7	999395.41186	5.442110e+02	6.037710e+01
9	8	998514.17406	1.337233e+03	1.485931e+02
10	9	996353.64083	3.281157e+03	3.652024e+02
11	10	991081.33821	8.022894e+03	8.957673e+02
12	11	978360.74138	1.945168e+04	2.187583e+03
13	12	948492.02019	4.621989e+04	5.288093e+03
14	13	882611.79053	1.049013e+05	1.248689e+04
15	14	755358.20503	2.165856e+05	2.805622e+04
16	15	562081.90779	3.803074e+05	5.761069e+04
17	16	353134.71697	5.427748e+05	1.040905e+05
18	17	194018.29408	6.420016e+05	1.639801e+05
19	18	100188.40465	6.697414e+05	2.300702e+05

20	19	51602.44697	6.519790e+05	2.964186e+05
21	20	27387.59027	6.128460e+05	3.597664e+05
22	21	15185.54519	5.660735e+05	4.187409e+05
23	22	8830.96091	5.182201e+05	4.729489e+05
24	23	5383.20133	4.721700e+05	5.224468e+05
25	24	3431.25004	4.290862e+05	5.674826e+05
26	25	2279.57994	3.893446e+05	6.083759e+05
27	26	1573.20098	3.529631e+05	6.454637e+05
28	27	1124.10875	3.198001e+05	6.790758e+05
29	28	829.03581	2.896469e+05	7.095240e+05
30	29	629.24759	2.622729e+05	7.370978e+05
31	30	490.22664	2.374463e+05	7.620635e+05
32	31	391.05841	2.149444e+05	7.846646e+05
33	32	318.70736	1.945584e+05	8.051229e+05
34	33	264.83290	1.760949e+05	8.236403e+05
35	34	223.96770	1.593760e+05	8.404000e+05
36	35	192.44672	1.442393e+05	8.555682e+05
37	36	167.76176	1.305366e+05	8.692957e+05
38	37	148.16299	1.181330e+05	8.817188e+05
39	38	132.40779	1.069061e+05	8.929615e+05
40	39	119.59909	9.674482e+04	9.031356e+05
41	40	109.07928	8.754833e+04	9.123426e+05
42	41	100.35935	7.922529e+04	9.206743e+05
43	42	93.07082	7.169293e+04	9.282140e+05
44	43	86.93256	6.487628e+04	9.350368e+05
45	44	81.72761	5.870743e+04	9.412108e+05
46	45	77.28671	5.312490e+04	9.467978e+05
47	46	73.47646	4.807302e+04	9.518535e+05
48	47	70.19070	4.350139e+04	9.564284e+05
49	48	67.34424	3.936439e+04	9.605683e+05
50	49	64.86809	3.562073e+04	9.643144e+05
51	50	62.70600	3.223302e+04	9.677043e+05
52	51	60.81171	2.916745e+04	9.707717e+05
53	52	59.14695	2.639338e+04	9.735475e+05
54	53	57.67984	2.388311e+04	9.760592e+05
55	54	56.38365	2.161157e+04	9.783321e+05
56	55	55.23587	1.955604e+04	9.803887e+05
57	56	54.21740	1.769601e+04	9.822498e+05
58	57	53.31200	1.601287e+04	9.839338e+05
59	58	52.50576	1.448981e+04	9.854577e+05
60	59	51.78671	1.311161e+04	9.868366e+05
61	60	51.14455	1.186448e+04	9.880844e+05

Pregunta 1

Analizando el dataframe `output` encuentre el día en que el número de contagios es máximo (el pico de la curva verde). ¿Después de cuántos días del inicio ocurre el máximo? Usando las ecuaciones diferenciales del modelo, encuentre una relación entre los parámetros del modelo válida para el valor de t correspondiente al máximo de la curva de infección.

Analizando el dataframe, podemos deducir que el pico de contagios se dio aproximadamente 18 días después de la aparición del primer infectado; este pico fue de 669741.4 personas

Observando el sistemas ecuaciones, podemos notar que éstas no son lineales, por lo que para encontrar la solución analítica de la cantidad del pico de infectados, será necesario deshacernos de dt . Esto lo hacemos diviendo $\frac{I}{S}$.

$$\frac{dI}{dS} = -1 + \frac{\gamma N}{\beta S}$$

Al resolver la ecuación diferencial por método de separación de variables, se obtiene lo siguiente:

$$I(t) = -S(t) + N \frac{\gamma}{\beta} \ln(|S(t)|) + C$$

Por nuestras condiciones iniciales, conocemos que

$$S_o = 999999 \quad I_o = 1 \quad R_o = 0$$

Evaluando I en t_o , es posible obtener la solución particular para nuestro PVI.

$$\begin{aligned} C &= I_o + S_o - N \frac{\gamma}{\beta} \ln(S_o) \\ I(t) &= -S(t) + N \frac{\gamma}{\beta} \ln(|S(t)|) + (I_o + S_o - N \frac{\gamma}{\beta} \ln(S_o)) \\ I(t) &= -S(t) + N + N \frac{\gamma}{\beta} \left(\ln \left(\frac{|S(t)|}{S_o} \right) \right) \end{aligned}$$

Geométricamente, cuando queremos encontrar el máximo de una función, decimos que en ese punto la pendiente de su recta tangente es 0; es decir, que su derivada es 0. No obstante, una pendiente de 0 también nos puede indicar un mínimo, así que al igualar el diferencial a 0, es necesario determinar la concavidad de la función.

$$\frac{dI}{dt} = \beta \frac{I}{N} S - \gamma I$$

$$\beta \frac{I}{N} S - \gamma I = 0$$

$$I \left(\frac{\beta S}{N} - \gamma \right) = 0$$

Que I fuera 0 implicaría que es un mínimo, así que igualamos el otro factor a 0, y tomamos a I por un punto máximo. Encontrando así la siguiente igualdad para S .

$$S = N \frac{\gamma}{\beta}$$

Una vez que conocemos el valor de S para cuando los infectados están en su pico, lo sustituimos en nuestra función $I(t)$ encontrada anteriormente.

$$I_{max} = N + N \frac{\gamma}{\beta} \left(\ln \left(\frac{N \gamma}{S_o \beta} \right) - 1 \right)$$

Para comprobar que dicha ecuación es una solución a su diferencial, sustituimos con los valores ya conocidos $N = 1,000,000$ $S_o = 999,999$ $\gamma = 0.1$ $\beta = 1$

$$I_{max} = 1000000 + 1000000 \frac{0.1}{1} \left(\ln \left(\frac{1000000 \cdot 0.1}{999999 \cdot 1} \right) - 1 \right)$$

$$I_{max} = 669741.5907$$

Dicha cifra coincide con la encontrada con el dataframe, así que concluimos que la solución es correcta.

Pregunta 2

Analizando el dataframe `output` encuentre después de cuántos días el número de “susceptibles” se reduce a la mitad. Usando la ecuación diferencial que expresa la variación del número de susceptibles, encuentre de manera analítica una fórmula que exprese el tiempo t necesario para que el número de susceptibles sea la mitad del valor inicial en función de β .

Dentro el dataframe se encuentra que después de 15 días, la cantidad de personas susceptibles se reduce casi por la mitad.

$$\frac{dS}{dR} = \frac{dS}{dt} * \frac{dt}{dR} = \left(-\frac{\beta IS}{N} \right) \left(\frac{1}{\gamma I} \right) \frac{dS}{dR} = -\frac{\beta}{N\gamma} S$$

$$\int \frac{dS}{S} = - \int \frac{\beta}{N\gamma} dR \ln|S| = -\frac{\beta}{N\gamma} R + C$$

Tomando en cuenta las condiciones iniciales del PVI, podemos sustituir y despejar para C .

$$\ln|S_0| = -\frac{\beta}{N\gamma} R_0 + C \quad C = \frac{\beta}{N\gamma} R_0 + \ln|S_0|$$

$$R_0 = 0 \rightarrow C = \ln|S_0| //$$

Y reemplazamos en la ecuación antes encontrada.

$$\ln|S(t)| = -\frac{\beta}{N\gamma} R(t) + \ln|S_0| //$$

Se aplica la base euler, para obtener lo siguiente:

$$S(t) = S_o * e^{-\frac{\beta}{N\gamma}R(t)}$$

Como buscamos la mitad de la población, dividimos $S_o/2$

$$\frac{1}{2}S_o = S_o e^{-\frac{\beta}{N\gamma}R(t)}$$

$$\ln\left(\frac{1}{2}\right) = -\frac{\beta}{N\gamma}R(t)$$

Despejando $R(t)$, obtenemos que

$$R(t) = \ln\left(\frac{1}{2}\right) \left(-\frac{\beta}{N\gamma}\right)^{-1}$$

Por último, para verificar que la solución al diferencial de \dot{R} es correcta, sustituimos con los valores ya conocidos

$$R(t) = \ln\left(\frac{1}{2}\right) \left(-\frac{1}{1000000(0.1)}\right)^{-1}$$

$$R(t) = 69314.7181$$

Pregunta 3

Estudie la dinámica del contagio variando los parámetros β y γ . Empiece con $\gamma = 0.1$ constante cambiando β (que representa la ‘fuerza’ de la infección):

- $\beta = 0.1$, 365 días

```
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 = 0.1,          # razón de infección
                gamma = 0.1)         # razón de recuperación

# valores de tiempo para resolver la ecuación, de 0 a 60 días
times <- seq(from = 0, to = 365, by = 1)

# Solución del modelo
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
dI <- lambda * S - gamma * I
dR <- gamma * I
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))

```

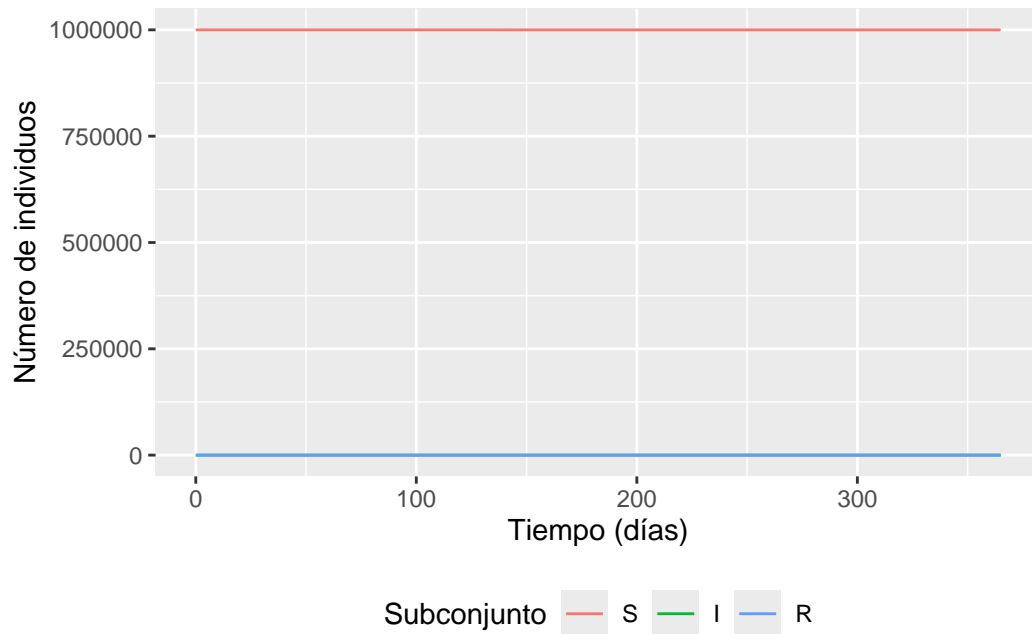
Gráficos de la evolución del sistema

```

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")

```



- $\beta = 0.3$, 365 días

```
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 = 0.3,          # razón de infección
                 gamma = 0.1)        # razón de recuperación

# valores de tiempo para resolver la ecuación, de 0 a 365 días
times <- seq(from = 0, to = 365, by = 1)

# Solución del modelo
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
    dI <- lambda * S - gamma * I
```

```

        dR <- gamma * I
        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))

```

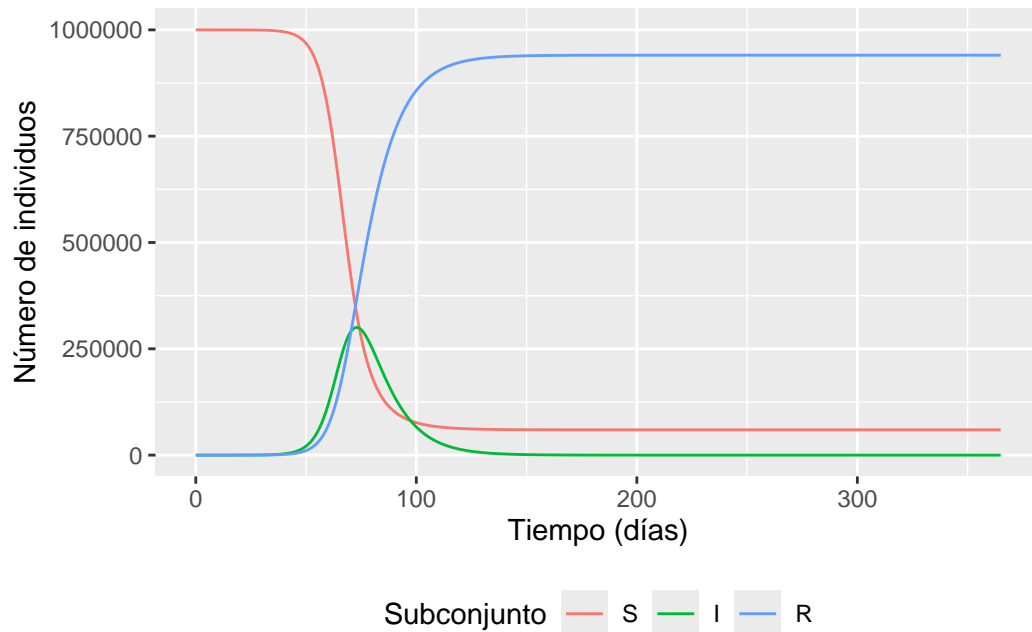
Gráficos de la evolución del sistema

```

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")

```



- $\beta = 0.7$, 60 días

```
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 = 0.7,          # razón de infección
                gamma = 0.1)         # razón de recuperación

# valores de tiempo para resolver la ecuación, de 0 a 60 días
times <- seq(from = 0, to = 60, by = 1)

# Solución del modelo
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
    dI <- lambda * S - gamma * I
```

```

        dR <- gamma * I
        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))

```

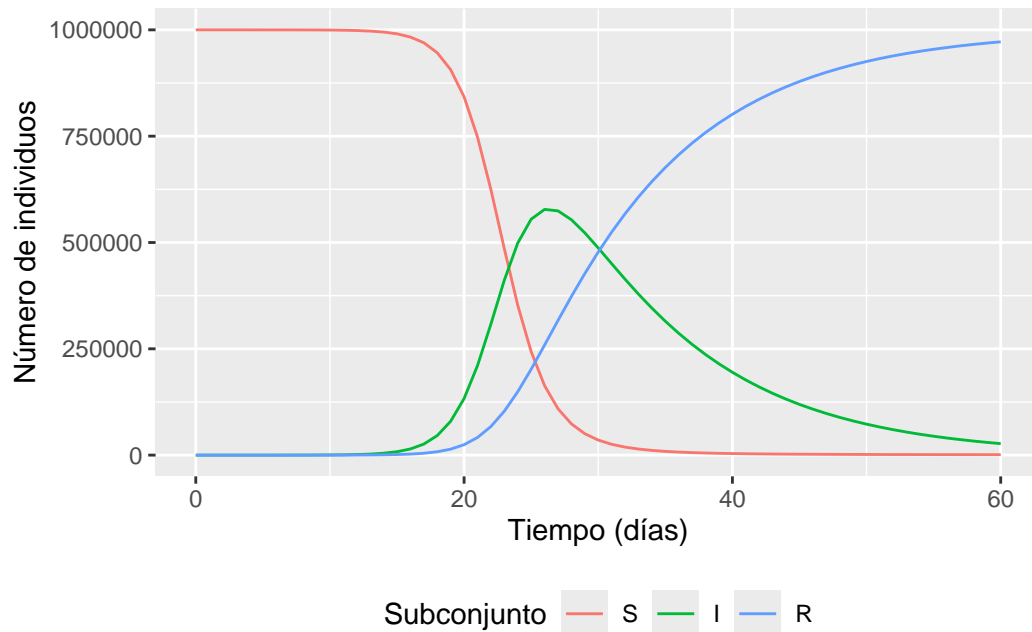
Gráficos de la evolución del sistema

```

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")

```



- $\beta = 0.9$, 60 días

```
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 = 0.9,          # razón de infección
                gamma = 0.1)        # razón de recuperación

# valores de tiempo para resolver la ecuación, de 0 a 60 días
times <- seq(from = 0, to = 60, by = 1)

# Solución del modelo
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
    dI <- lambda * S - gamma * I
```

```

        dR <- gamma * I
        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))

```

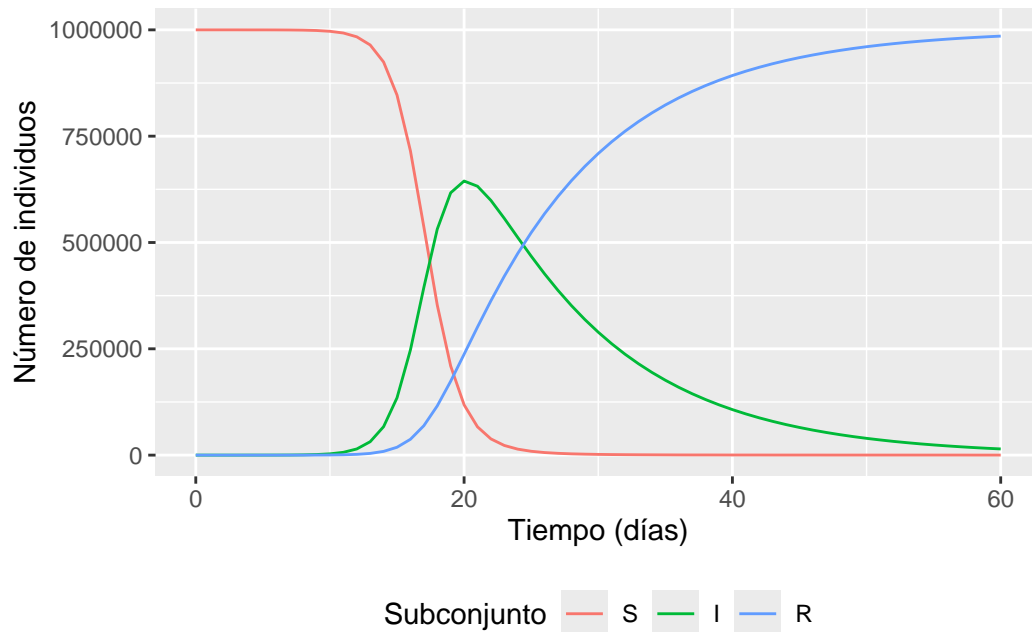
Gráficos de la evolución del sistema

```

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")

```



- $\beta = 1.2$, 60 días

```
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 = 1.2,          # razón de infección
                gamma = 0.1)         # razón de recuperación

# valores de tiempo para resolver la ecuación, de 0 a 60 días
times <- seq(from = 0, to = 60, by = 1)

# Solución del modelo
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
    dI <- lambda * S - gamma * I
```



```

        dR <- gamma * I
        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))

```

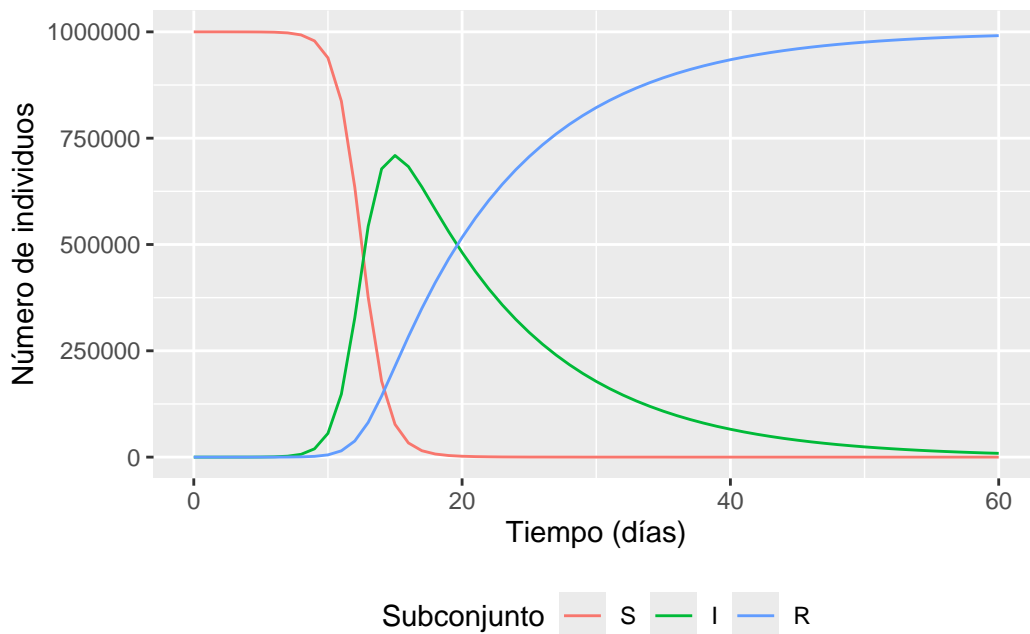
Gráficos de la evolución del sistema

```

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")

```



Comente acerca de los cambios que se observan en las curvas. Encuentre una relación entre β y γ necesaria para que ocurra la epidemia. Para que haya una epidemia la fuerza de infección (β) debe ser suficientemente alta por un tiempo suficientemente largo (γ suficientemente bajo) de manera que se pueda transmitir el agente patógeno. A partir de este estudio se puede definir el coeficiente R_0 de la infección.

Al contar siempre con $\gamma = 0.1$ pero con distintos valores en β (.1, .3, .7, .9, 1.2), se pueden observar distintos cambios en la relación entre la propagación de la infección y la recuperación. Graficando las curvas se analiza que mientras mayor es β con respecto a γ , el número de gente sana (la curva S) baja de manera más precipitada, mientras que el número de infectados (curva I) alcanza un pico máximo mayor y en menor tiempo, pero abarca menos tiempo. Debido al aumento más rápido en la curva I , el número de recuperados (curva R) también aumenta más rápido a mayor β .

En todos los casos γ es igual a 0.1, los casos específicos: - En el caso donde β y γ ambos son iguales a 0.1, las curvas permanecen casi constantes con respecto al tiempo. - Cuando $\beta = 0.3$, la epidemia avanza lento y alcanza un pico de infectados un poco sobre un cuarto del total de ciudadanos, en alrededor de 75 días. - Cuando $\beta = 0.7$, la epidemia avanza mucho más rápido que el caso anterior, y alcanza un pico por encima de la mitad de ciudadanos, en alrededor de 25 días. - Cuando $\beta = 0.9$, la epidemia avanza más rápido que los casos anteriores, alcanzando un pico de alrededor de 700000 en 20 días. - Cuando $\beta = 1.2$, la epidemia muestra el avance más rápido de todos los casos, y alcanza un pico de casi 750000 en alrededor de 15 días.

Parece que para que se desate la infección, β debe ser mayor que γ , y mientras mayor sea que γ , la cantidad de infectados aumenta más rápido con respecto al tiempo, aunque el pico

también dura menos tiempo. Cuando ambos coeficientes son iguales, parece tener un impacto mínimo en las curvas S y R , pues se recuperan e infectan al mismo ritmo.

Pregunta 4

Después, con $\beta = 1$ varíe el valor de γ :

- $\gamma = 0.025$, 60 días

```
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 = 1,          # razón de infección
                 gamma = 0.025)    # razón de recuperación

# valores de tiempo para resolver la ecuación, de 0 a 60 días
times <- seq(from = 0, to = 60, by = 1)

# Solución del modelo
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

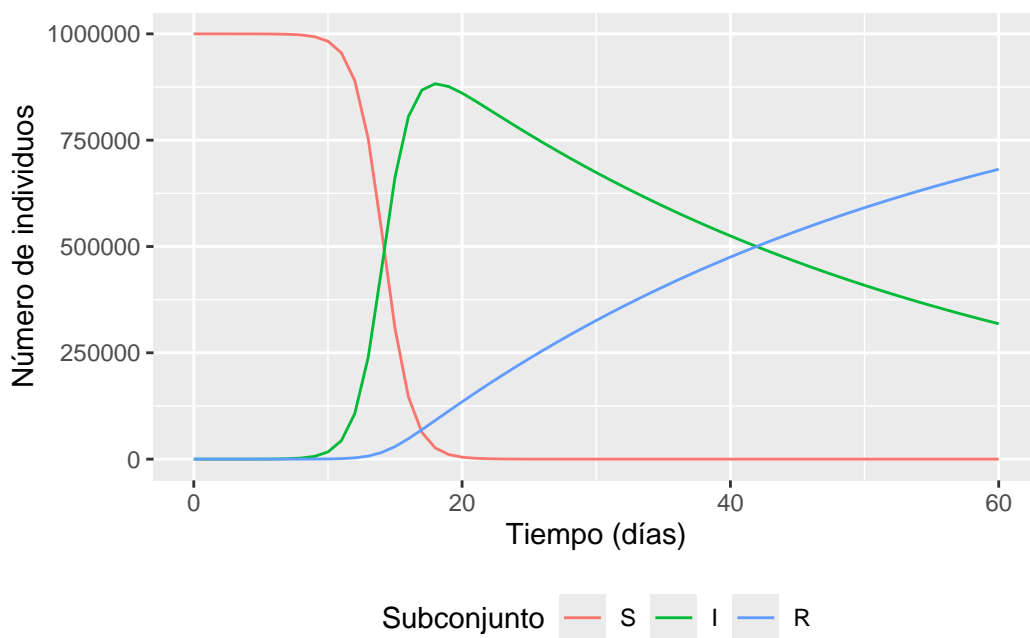
    N <- S+I+R
    lambda <- beta * I/N
    dS <- -lambda * S
    dI <- lambda * S - gamma * I
    dR <- gamma * I
    return(list(c(dS, dI, dR)))
  })
}

# 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))
```

Gráficos de la evolución del sistema

```
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")
```



- $\gamma = 0.2$, 60 días

```
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 = 1,           # razón de infección
```

```

        gamma = 0.2)    # razón de recuperación

# valores de tiempo para resolver la ecuación, de 0 a 60 días
times <- seq(from = 0, to = 60, by = 1)

# Solución del modelo
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
    dI <- lambda * S - gamma * I
    dR <- gamma * I
    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))

```

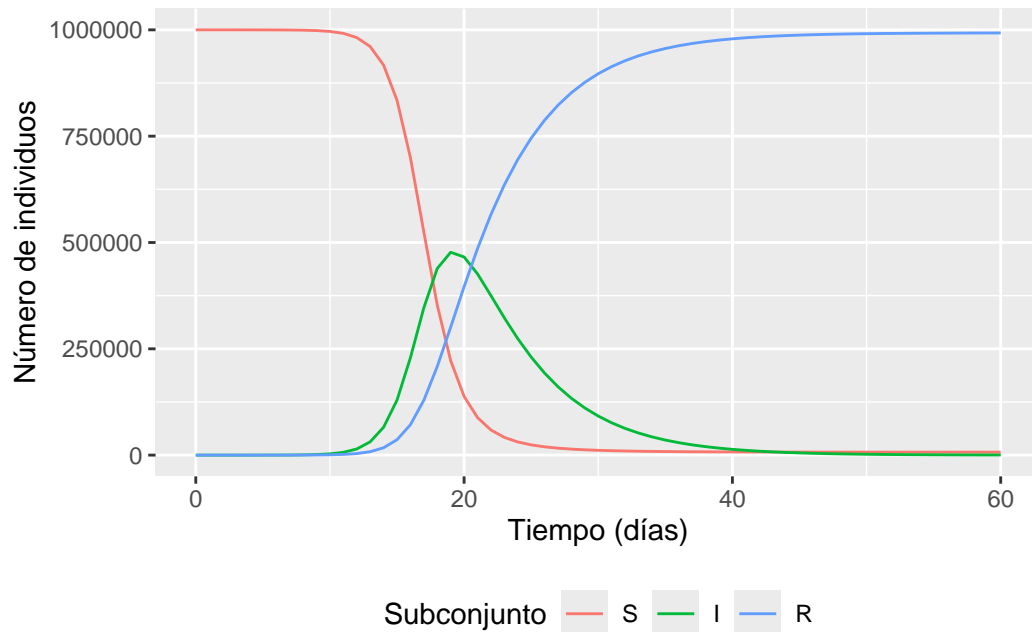
Gráficos de la evolución del sistema

```

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")

```



- $\gamma = 0.5$, 60 días

```
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 = 1,           # razón de infección
                gamma = 0.5)        # razón de recuperación

# valores de tiempo para resolver la ecuación, de 0 a 60 días
times <- seq(from = 0, to = 60, by = 1)

# Solución del modelo
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
    dI <- lambda * S - gamma * I
  })
}
```

```

        dR <- gamma * I
        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))

```

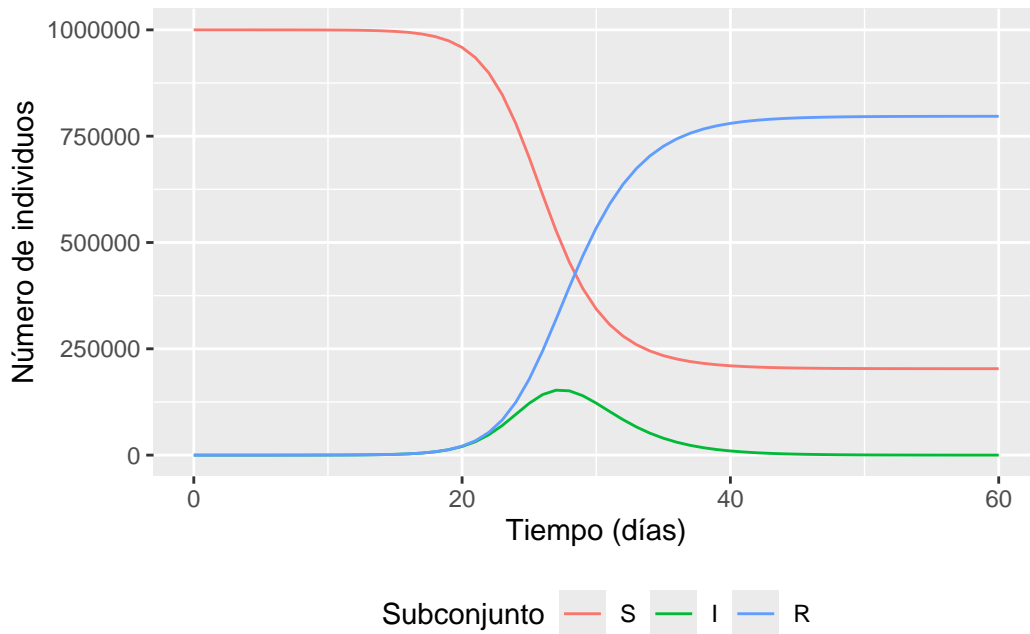
Gráficos de la evolución del sistema

```

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")

```



- $\gamma = 1$, 365 días

```
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 = 1,           # razón de infección
                gamma = 1)          # razón de recuperación

# valores de tiempo para resolver la ecuación, de 0 a 60 días
times <- seq(from = 0, to = 365, by = 1)

# Solución del modelo
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
    dI <- lambda * S - gamma * I
```



```

        dR <- gamma * I
        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))

```

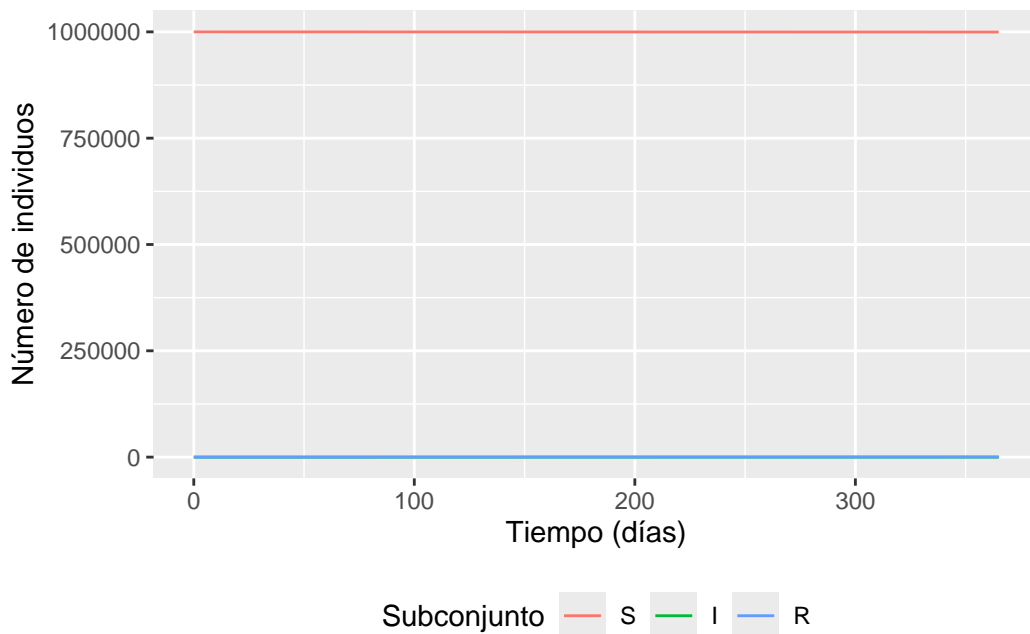
Gráficos de la evolución del sistema

```

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")

```



Comente acerca de los cambios que se observan en las curvas. Encuentre una relación entre β y γ necesaria para que ocurra la epidemia. Para que haya una epidemia la fuerza de infección (β) debe ser suficientemente alta por un tiempo suficientemente largo (γ suficientemente bajo) de manera que se pueda transmitir el agente patógeno. A partir de este estudio se puede definir el coeficiente R_0 de la infección.

Se cuentan con tres curvas: Susceptibles (S), Infectados (I), y Recuperados (R). También, con razones de cambio, β es la tasa de infección y γ la tasa de recuperación. Al contar con β siempre en 1 pero con distintos valores en γ (.025, .2, .5, 1), se pueden observar distintos cambios en la relación entre la propagación de la infección y la recuperación. Graficando las curvas se analiza que mientras γ sea menor a β , conforme el valor de γ este más lejos del valor de β , o sea que sea una cantidad más pequeña, se presentan los siguientes casos: La curva S desciende a mayor velocidad, el punto máximo de la curva I va incrementando su valor, y en la curva R aumenta más lentamente. En el caso específico que se presenta cuando $\beta = \gamma = 1$, no existe una curva I , ya que realmente es la misma tasa de infección y recuperación, por lo que las curvas S y R son lineales.

En todos los casos se sabe que $\beta = 1$, cada uno en específico:

- Cuando $\gamma = 0.025$ el agente patógeno se transmite extremadamente rápido en comparación a como los que ya están infectados se van recuperando, por lo que esto sería un brote altamente fuerte y difícil de detener su propagación.
- Cuando $\gamma = 0.2$ esto también indica una epidemia significativa (propagación alta), pero con una mayor posibilidad a poder controlarla que en el caso anterior gracias a que la recuperación ocurre más rápidamente.

- Cuando $\gamma = 0.5$ la epidemia es menos explosiva, la tasa de recuperación es rápida, por lo que habrá más control y menos impacto.
- Cuando $\gamma = 1$ esto es un punto de equilibrio en una epidemia, por cada persona que se recupera, otra se infecta. La epidemia no crece ni decrece de manera significativa.

Analizado esto, se llega a la relación general: entre más pequeño γ , implica una propagación más rápida del agente patógeno y una mayor dificultad para controlarla. Por otro lado, entre más grande sea γ , la recuperación es más rápida, por lo tanto la epidemia tiene menor impacto.

A partir de este estudio se define el coeficiente R_0 de la infección como el número básico de reproducción, este es un indicador clave en las epidemias. Esta dado por: $R_0 = \frac{\gamma}{\beta}$ (tasa de recuperación entre tasa de infección). Gracias a esto se puede saber la cantidad de casos que se dan por los casos primarios. Por ejemplo, en los casos vistos previamente, R_0 va en decremento, por lo tanto, mientras que R_0 sea $>$, γ será $<$. Esto nos dice que cuando R_0 tiene un valor grande, la epidemia es más fuerte y que, por ejemplo, en el caso de $\beta = \gamma = 1$, al utilizar: $R_0 = \frac{\gamma}{\beta} R_0 = \frac{1}{1} = 1$ significa que se mantiene en 1, por lo que no habrá un aumento explosivo de casos en la epidemia. Para concluir, si se quiere eliminar la epidemia, es necesaria la reducción de R_0 (debajo de 1), esto se lograría aumentando la tasa de recuperación o disminuyendo la tasa de infección.