Actividad 1: Modelo SIR

A. Govea, D. Vertiz, F. Gutierrez, J. Dong

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 *suceptibles* 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 suceptibles 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 $S \to I$) o que una vez enfermos se recuperen (cambio $I \to R$). Además, 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:

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

donde:

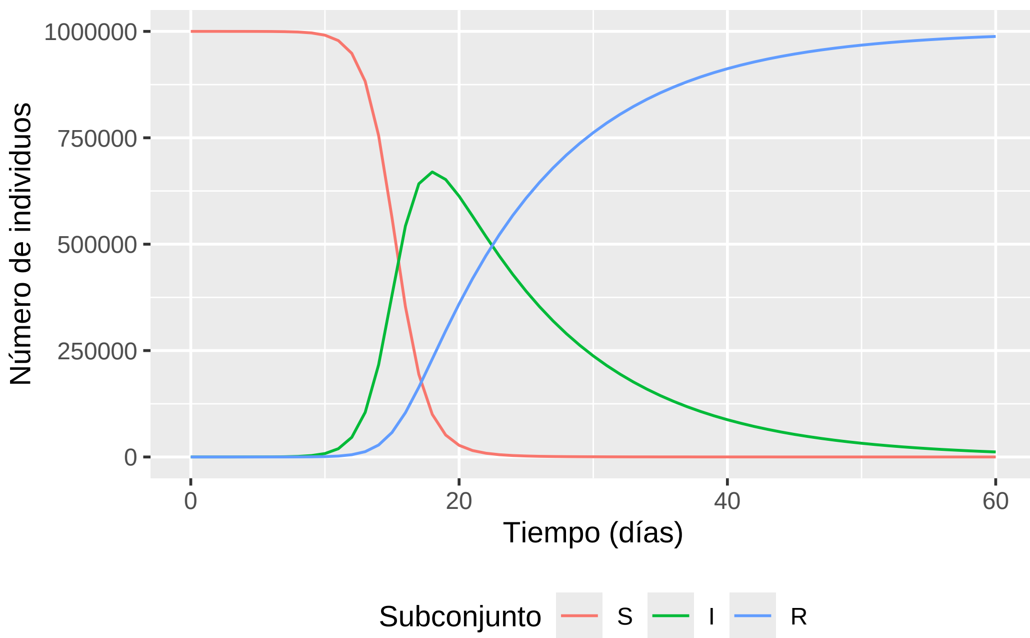
- $N = S + R + I$
- la cantidad $\beta \frac{I}{N}$ representa la razón con que las personas salen del compartimento S (se infectan);
- en la primera ecuación dS representa el cambio debido a las personas que salen del compartimento S (el signo negativo se debe a que las personas salen)
- en la segunda 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
```

```r
        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

```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")
```




```r
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.

```r
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

Solución Analítica

Para encontrar el día en que el número de contagios es máximo tenemos que igualar la derivada de Infectados Respecto del tiempo a cero, de la siguiente manera:

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

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

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

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

Finalmente podemos obtener la cantidad de personas susceptibles dentro del modelo cuando I "número de personas Infectados" es máximo.

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

Para encontrar cuánto vale "I" podemos dividir la derivada de "I" entre "S".

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

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

De esta manera obtendremos la Ecuación Diferencial de Infectados Respecto de Población Susceptible; y podremos encontrar el valor de "I".

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

$$\int dI = \int \left(-1 + \frac{\gamma N}{\beta S} \right) dS$$

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

$$C = I_0 + S_0 - N \frac{\gamma}{\beta} \ln |S_0|$$

$$I = -S(t) + I_0 + S_0 + N \frac{\gamma}{\beta} \ln \left| \frac{S(t)}{S_0} \right|$$

Se sabe que

$$N_0 = I_0 + S_0$$

Entonces podemos deducir que

$$I = -S(t) + N + N \frac{\gamma}{\beta} \ln \left| \frac{S(t)}{S_0} \right|$$

$$I = N + \frac{N\gamma}{\beta} \left[\ln \left(\frac{N\gamma}{\beta S_0} \right) - 1 \right]$$

$$I = 1000000 + \frac{1000000(0.1)}{1} \left[\ln \left(\frac{1000000(0.1)}{999999(1)} \right) \right]$$

$$I = 669741$$

Finalmente obtenemos el número de Infectados. Con este valor concluimos que el día donde se registra el número máximo de Infectados es el día 18.

Análisis del dataframe

Solución Numérica

Como podemos ver el punto más alto en la curva verde, es decir, los infectados, con los valores iniciales, fue en el día 18, con

```
# 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,
```

```r
                              times = times,
                              func = sir_model,
                              parms = parameters))
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 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.

El primer paso necesario es encontrar el valor de $\frac{ds}{dr}$, el cual se obtiene dividiendo las ecuaciones diferenciales de suceptibles y recuperados

$$\frac{ds}{dr} = \frac{ds}{dt} \cdot \frac{dt}{dr} = (-\beta \frac{I}{N} S)(\frac{1}{\gamma I})$$

obtiendo que $\frac{ds}{dr}$ es:

$$\frac{ds}{dr} = -\frac{\beta}{N\gamma} S$$

Integramos ambas partes de la igualdad:

$$\int \frac{ds}{S} = \int -\frac{\beta}{N\gamma} dR$$

obteniendo la siguiente igualdad:

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

Tomamos los valores inciales de S y de R para obtener el valor de C

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

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

$$R_0 = 0 \Rightarrow C = ln\,|S_0|$$

#por que dejo de ser s y r_o para ser s y r respecto de t

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

Aplicamos base e en ambos lados de la iguadad:

$$S(t) = S_0 \cdot e^{-\frac{\beta}{N\gamma} R(t)}$$

Estamos busacando una fórmula que exprese el tiempo t necesario para que el número de susceptibles sea la mitad del valor inicial en función de β, por lo que sustituimos $S(t)$ por $\frac{1}{2} S_0$

$$\frac{1}{2} S_0 = S_0 \cdot e^{-\frac{\beta}{N\gamma} R(t)} \Rightarrow \frac{1}{2} = e^{-\frac{\beta}{N\gamma} R(t)}$$

Aplicamos ln en ambos lados de la igualdad:

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

Despejamos la igualdad para buscar el valor de R(t)

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

Sustituimos los valores de β, N y γ para encontrar $R(t)$

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

y obtenemos que el valor tiempo t necesario paraque el número de susceptibles sea la mitad del valor inicial en función de β es:

$$R(t) = 69,314.7181$$

El tiempo t en el que el número de susceptibles es la mitad del valor inicial en función de β en la grafica se encuentra en algun punto entre los dias 18 y 19

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):

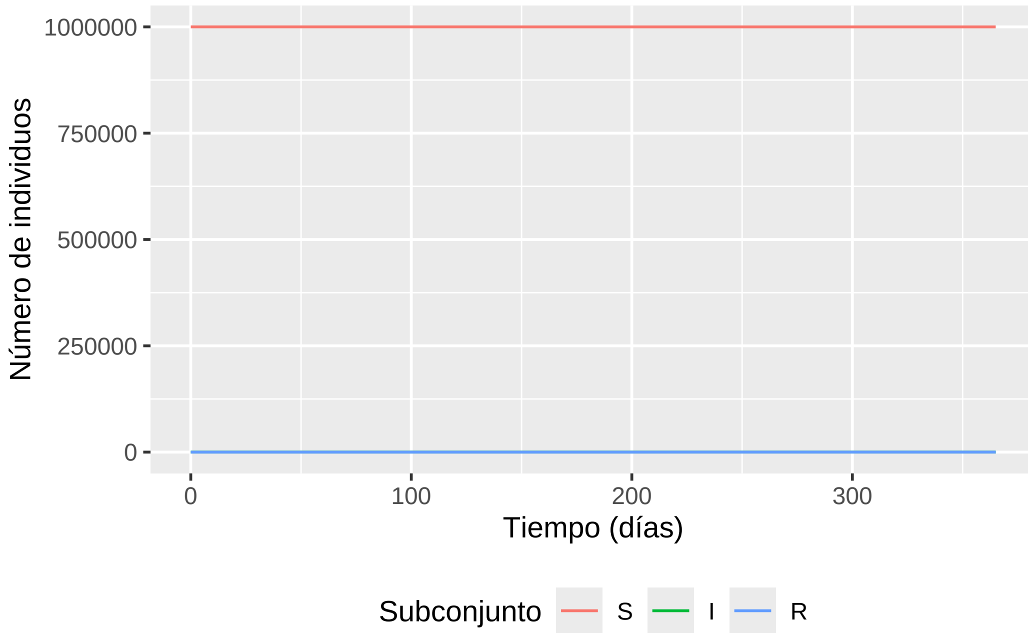
- $\beta = 0.1$, 365 días
- $\beta = 0.3$, 365 días
- $\beta = 0.7$, 60 días
- $\beta = 0.9$, 60 días
- $\beta = 1.2$, 60 días

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.

β= 0.1, 365 días

```
# 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 = 0.1,      # 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 = 18.25)

# 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))

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     0.00 999999.0 1.0000000  0.000000
2    18.25 999997.2 0.9999965  1.824997
3    36.50 999995.4 0.9999897  3.649985
4    54.75 999993.5 0.9999795  5.474957
5    73.00 999991.7 0.9999661  7.299908
6    91.25 999989.9 0.9999492  9.124831
7   109.50 999988.1 0.9999291 10.949720
8   127.75 999986.2 0.9999056 12.774570
9   146.00 999984.4 0.9998788 14.599374
10  164.25 999982.6 0.9998487 16.424126
11  182.50 999980.8 0.9998152 18.248820
12  200.75 999978.9 0.9997785 20.073449
13  219.00 999977.1 0.9997383 21.898009
14  237.25 999975.3 0.9996949 23.722492
15  255.50 999973.5 0.9996481 25.546893
16  273.75 999971.6 0.9995980 27.371206
17  292.00 999969.8 0.9995446 29.195424
18  310.25 999968.0 0.9994879 31.019542
19  328.50 999966.2 0.9994278 32.843553
```

```
20 346.75 999964.3 0.9993644 34.667451
21 365.00 999962.5 0.9992977 36.491231
```

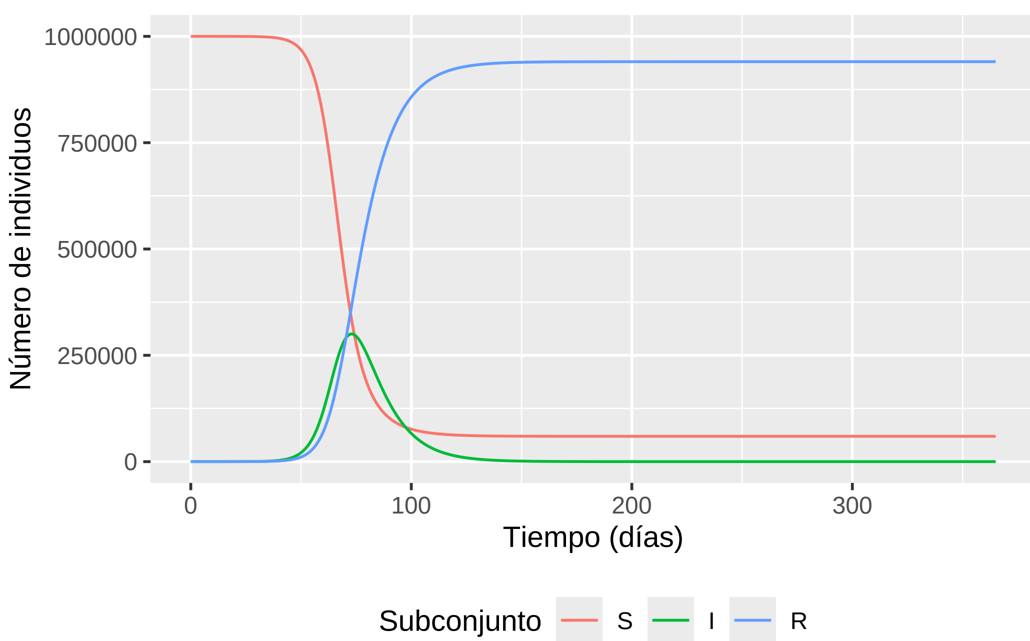
Como podemos ver, β es demasiado bajo para el tiempo tan largo como es un año. Incluso en el modelo, se aprecia en los datos duros que solo llegó a infectar a casi 35 personas de 1 millón iniciales.

(En 360 días, los Sanos llegaron a 999963 y los infectados a casi 36, una cifra nada importante e incluso se podría considerar una enfermedad que no va a durar).

β= 0.3, 365 días

```
# 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 = 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))

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")
```

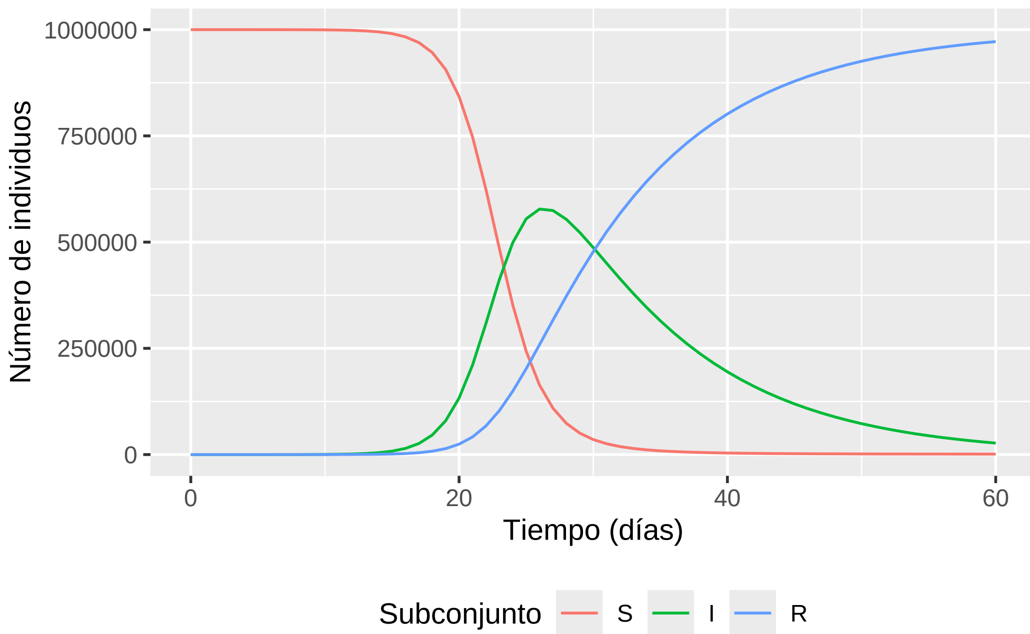


Con $\beta = 0.3$, la tasa de infección es moderada pero suficiente para iniciar una epidemia, haciendo que el número de infectados crezca de manera constante a lo largo del tiempo. Aunque el pico de infectados ocurre más tarde que con valores más altos de β, la propagación es controlada y la epidemia se extiende durante un período más prolongado, con muchos susceptibles aún presentes hacia el final del año.

β= 0.7, 60 días

```
# 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 = 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))

output_long <- melt(as.data.frame(output), id = "time")
```

```r
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")
```

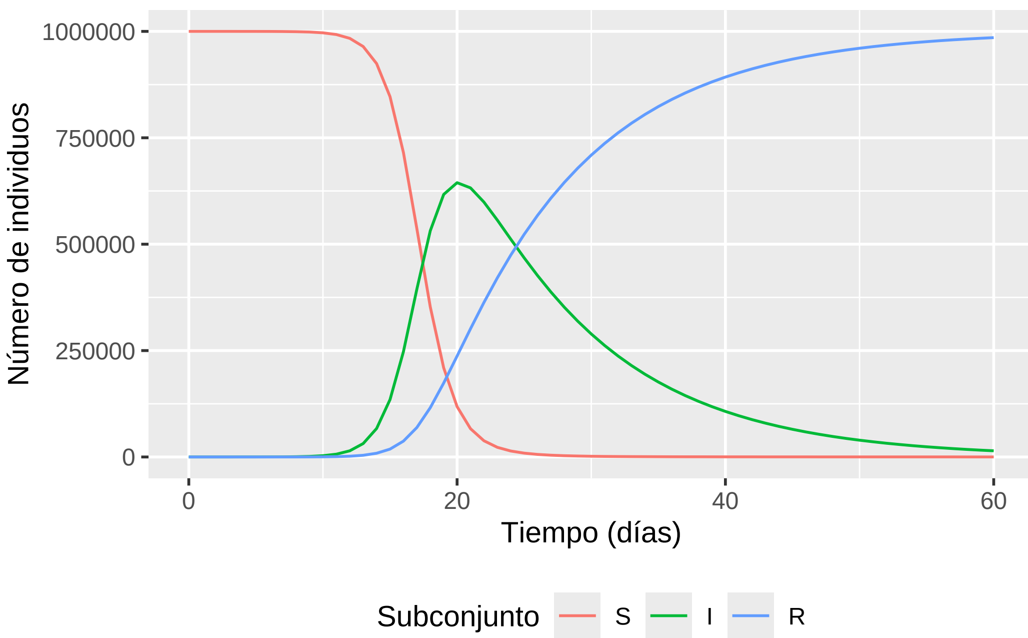


Con β= 0.7 , la tasa de infección es significativamente mayor, lo que provoca una propagación más rápida de la enfermedad. Aunque el tiempo total es más corto (60 días), un mayor número de personas se infecta antes de que los casos empiecen a disminuir. Esto resulta en un pico de infectados mucho más temprano, con una reducción más acelerada de los susceptibles y una mayor población recuperada al final, comparado con $\beta = 0.3$, lo que indica una epidemia más severa en menos tiempo.

β= 0.9, 60 días

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

# Initial states
```

```r
initial_state_values <- c(S = 999999,  # Número de susceptibles inicial
                                       #
                          I = 1,       # Se inicia con una persona infectada
                          R = 0)       #


# razones en unidades de díasˆ-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))

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")
```

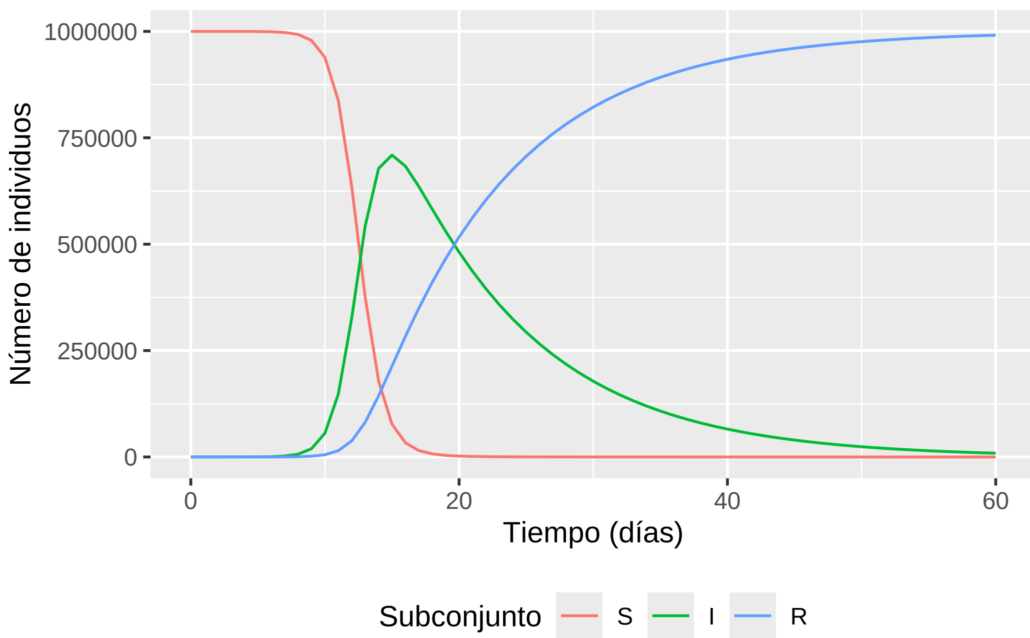


Con $\beta = 0.9$ durante 60 días, la propagación del virus es rápida, con un aumento significativo en el número de infectados en poco tiempo. A diferencia de $\beta = 0.3$ o $\beta = 0.7$, aquí una gran parte de la población susceptible es infectada en tan solo dos meses. Esto resulta en una disminución acelerada de los susceptibles, mientras los infectados alcanzan un pico temprano y más alto. El equilibrio entre susceptibles y recuperados ocurre de manera abrupta, lo que indica que la enfermedad agota a los susceptibles a gran velocidad.

β= 1.2, 60 días

```
# 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 = 1.2,      # 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))

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 $\beta = 1.2$ durante 60 días, la tasa de infección es extremadamente alta, lo que genera un brote explosivo en pocos días. La mayoría de los susceptibles son infectados rápidamente y el número de recuperados crece casi de inmediato. A diferencia de valores más bajos como $\beta = 0.3$ o $\beta = 0.9$, la transición de susceptibles a infectados y luego a recuperados ocurre en un tiempo muy corto. Esto sugiere que, sin medidas preventivas, la enfermedad podría saturar los sistemas de salud rápidamente.

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 R0 de la infección, que como James Holland lo describió, se puede obtener así:

$$R_0 \propto \left(\frac{\text{infection}}{\text{contact}}\right) \cdot \left(\frac{\text{contact}}{\text{time}}\right) \cdot \left(\frac{\text{time}}{\text{infection}}\right)$$

y después de analizar la susceptibilidad de una población, describe un coeficiente:

$$\frac{\beta}{\nu} = R_0 \geq 1$$

Que se refiere en este caso a la proporción de β y γ, por lo que con las diferentes β, serían

$$R_0 = 1, 3, 4, 9, 12$$

respectivamente.

Pregunta 4

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

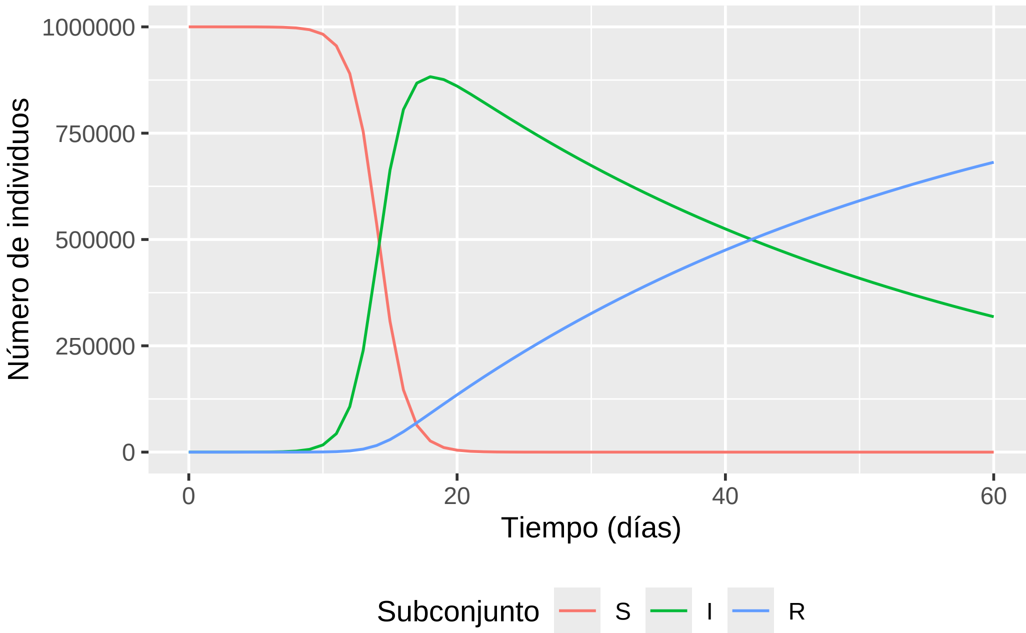
- $\gamma = 0.025$, 60 días
- $\gamma = 0.2$, 60 días
- $\gamma = 0.5$, 60 días
- $\gamma = 1$, 365 días

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.

γ= 0.025, 60 días

```
# 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 = 0.025)   # 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
```

```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))

#graficos de la evolucion 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")
```




```r
print(output)
```

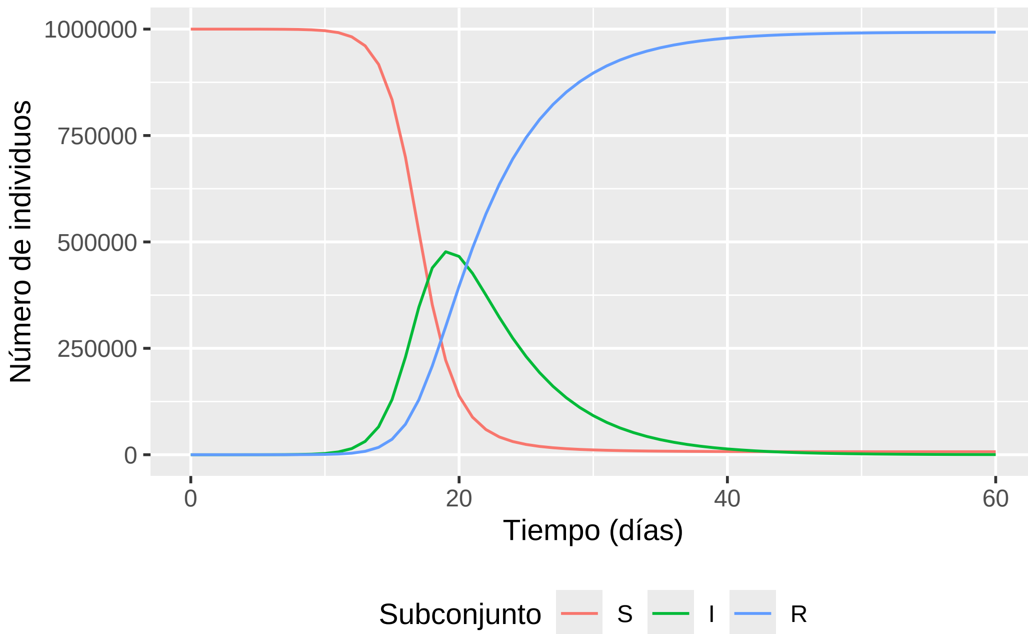
```
   time            S            I            R
1     0 9.999990e+05 1.000000e+00 0.000000e+00
2     1 9.999973e+05 2.651171e+00 4.233779e-02
3     2 9.999928e+05 7.028673e+00 1.545820e-01
4     3 9.999809e+05 1.863396e+01 4.521575e-01
5     4 9.999494e+05 4.940020e+01 1.241064e+00
6     5 9.998657e+05 1.309571e+02 3.332463e+00
7     6 9.996440e+05 3.471104e+02 8.876251e+00
8     7 9.990567e+05 9.196941e+02 2.356767e+01
9     8 9.975032e+05 2.434376e+03 6.247431e+01
10    9 9.934079e+05 6.426764e+03 1.653228e+02
11   10 9.827138e+05 1.685030e+04 4.359092e+02
12   11 9.554588e+05 4.340213e+04 1.139066e+03
13   12 8.900975e+05 1.069920e+05 2.910584e+03
14   13 7.538350e+05 2.391005e+05 7.064517e+03
15   14 5.374329e+05 4.470433e+05 1.552376e+04
16   15 3.073986e+05 6.631112e+05 2.949023e+04
17   16 1.463834e+05 8.055785e+05 4.803811e+04
18   17 6.306218e+04 8.678470e+05 6.909084e+04
19   18 2.621107e+04 8.827496e+05 9.103931e+04
```

20	19	1.086622e+04	8.760814e+05	1.130524e+05
21	20	4.557867e+03	8.606697e+05	1.347725e+05
22	21	1.945215e+03	8.419953e+05	1.560595e+05
23	22	8.463628e+02	8.222897e+05	1.768640e+05
24	23	3.756229e+02	8.024514e+05	1.971730e+05
25	24	1.700284e+02	7.828415e+05	2.169885e+05
26	25	7.847470e+01	7.636033e+05	2.363182e+05
27	26	3.691489e+01	7.447909e+05	2.551722e+05
28	27	1.769082e+01	7.264209e+05	2.735614e+05
29	28	8.633279e+00	7.084944e+05	2.914970e+05
30	29	4.288387e+00	6.910059e+05	3.089898e+05
31	30	2.167273e+00	6.739470e+05	3.260508e+05
32	31	1.113911e+00	6.573082e+05	3.426907e+05
33	32	5.820010e-01	6.410798e+05	3.589197e+05
34	33	3.089994e-01	6.252517e+05	3.747480e+05
35	34	1.666405e-01	6.098143e+05	3.901855e+05
36	35	9.124819e-02	5.947580e+05	4.052419e+05
37	36	5.071374e-02	5.800734e+05	4.199265e+05
38	37	2.859727e-02	5.657514e+05	4.342486e+05
39	38	1.635562e-02	5.517830e+05	4.482170e+05
40	39	9.484189e-03	5.381594e+05	4.618406e+05
41	40	5.574190e-03	5.248722e+05	4.751278e+05
42	41	3.319434e-03	5.119131e+05	4.880869e+05
43	42	2.002233e-03	4.992739e+05	5.007261e+05
44	43	1.222884e-03	4.869468e+05	5.130532e+05
45	44	7.559934e-04	4.749240e+05	5.250760e+05
46	45	4.729315e-04	4.631981e+05	5.368019e+05
47	46	2.993390e-04	4.517617e+05	5.482383e+05
48	47	1.915362e-04	4.406077e+05	5.593923e+05
49	48	1.239085e-04	4.297290e+05	5.702710e+05
50	49	8.102576e-05	4.191190e+05	5.808810e+05
51	50	5.360390e-05	4.087709e+05	5.912291e+05
52	51	3.584293e-05	3.986783e+05	6.013217e+05
53	52	2.420756e-05	3.888349e+05	6.111651e+05
54	53	1.647926e-05	3.792345e+05	6.207655e+05
55	54	1.132625e-05	3.698712e+05	6.301288e+05
56	55	7.860772e-06	3.607391e+05	6.392609e+05
57	56	5.504585e-06	3.518324e+05	6.481676e+05
58	57	3.888704e-06	3.431456e+05	6.568544e+05
59	58	2.771168e-06	3.346733e+05	6.653267e+05
60	59	1.991469e-06	3.264102e+05	6.735898e+05
61	60	1.442901e-06	3.183511e+05	6.816489e+05

γ= 0.2, 60 días

```
# 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 = 0.2)   # 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))

#graficos de la evolucion del sistema
output_long <- melt(as.data.frame(output), id = "time")
```

```r
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")
```




```r
print(output)
```

```
   time        S            I            R
1     0 999999.000 1.000000e+00 0.000000e+00
2     1 999997.468 2.225542e+00 3.063862e-01
3     2 999994.059 4.953019e+00 9.882589e-01
4     3 999986.471 1.102305e+01 2.505786e+00
5     4 999969.585 2.453176e+01 5.883056e+00
6     5 999932.007 5.459392e+01 1.339906e+01
7     6 999848.387 1.214884e+02 3.012498e+01
8     7 999662.342 2.703148e+02 6.734296e+01
9     8 999248.572 6.012860e+02 1.501421e+02
10    9 998329.090 1.336649e+03 3.342615e+02
```

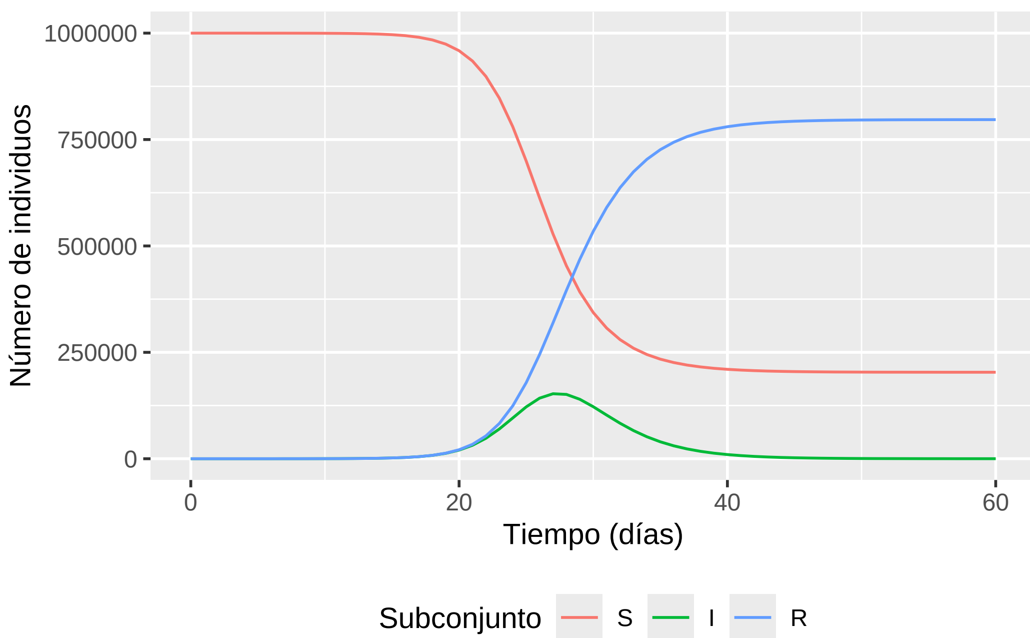
11	10	996289.561	2.967171e+03	7.432680e+02
12	11	991783.980	6.566228e+03	1.649792e+03
13	12	981919.330	1.443164e+04	3.649026e+03
14	13	960738.133	3.125138e+04	8.010484e+03
15	14	917097.912	6.559408e+04	1.730801e+04
16	15	834338.368	1.294386e+05	3.622305e+04
17	16	699301.510	2.291640e+05	7.153445e+04
18	17	524302.186	3.465605e+05	1.291373e+05
19	18	352660.117	4.388900e+05	2.084498e+05
20	19	222031.282	4.769817e+05	3.009870e+05
21	20	138113.082	4.659506e+05	3.959363e+05
22	21	88271.115	4.262606e+05	4.854683e+05
23	22	59101.454	3.751989e+05	5.656997e+05
24	23	41695.436	3.228322e+05	6.354724e+05
25	24	30950.499	2.739766e+05	6.950729e+05
26	25	24061.785	2.305126e+05	7.454256e+05
27	26	19480.615	1.928531e+05	7.876663e+05
28	27	16331.037	1.607323e+05	8.229366e+05
29	28	14101.597	1.336060e+05	8.522924e+05
30	29	12483.390	1.108465e+05	8.766701e+05
31	30	11283.545	9.183569e+04	8.968808e+05
32	31	10377.785	7.600591e+04	9.136163e+05
33	32	9683.671	6.285477e+04	9.274616e+05
34	33	9145.029	5.194732e+04	9.389076e+05
35	34	8722.647	4.291218e+04	9.483652e+05
36	35	8388.544	3.543517e+04	9.561763e+05
37	36	8122.363	2.925222e+04	9.626254e+05
38	37	7909.029	2.414234e+04	9.679486e+05
39	38	7737.204	1.992125e+04	9.723415e+05
40	39	7598.247	1.643565e+04	9.759661e+05
41	40	7485.490	1.355821e+04	9.789563e+05
42	41	7393.739	1.118340e+04	9.814229e+05
43	42	7318.909	9.223784e+03	9.834573e+05
44	43	7257.763	7.607028e+03	9.851352e+05
45	44	7207.722	6.273312e+03	9.865190e+05
46	45	7166.714	5.173197e+03	9.876601e+05
47	46	7133.074	4.265844e+03	9.886011e+05
48	47	7105.454	3.517529e+03	9.893770e+05
49	48	7082.759	2.900411e+03	9.900168e+05
50	49	7064.101	2.391511e+03	9.905444e+05
51	50	7048.753	1.971868e+03	9.909794e+05
52	51	7036.124	1.625838e+03	9.913380e+05
53	52	7025.728	1.340515e+03	9.916338e+05

```
54   53   7017.168 1.105254e+03 9.918776e+05
55   54   7010.119 9.112744e+02 9.920786e+05
56   55   7004.312 7.513346e+02 9.922444e+05
57   56   6999.527 6.194629e+02 9.923810e+05
58   57   6995.585 5.107347e+02 9.924937e+05
59   58   6992.337 4.210889e+02 9.925866e+05
60   59   6989.660 3.471770e+02 9.926632e+05
61   60   6987.453 2.862379e+02 9.927263e+05
```

$\gamma = 0.5$, 60 días

```
# 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 = 0.5)   # 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))

#graficos de la evolucion 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	999999.0	1.000000e+00	0.000000e+00
2	1	999997.7	1.648720e+00	6.487217e-01
3	2	999995.6	2.718271e+00	1.718281e+00
4	3	999992.0	4.481646e+00	3.481677e+00
5	4	999986.2	7.388911e+00	6.389006e+00
6	5	999976.6	1.218204e+01	1.118231e+01
7	6	999960.8	2.008418e+01	1.908495e+01
8	7	999934.8	3.311153e+01	3.211366e+01
9	8	999891.8	5.458708e+01	5.359293e+01
10	9	999821.0	8.998625e+01	8.900227e+01
11	10	999704.3	1.483278e+02	1.473716e+02
12	11	999512.0	2.444575e+02	2.435766e+02
13	12	999195.1	4.027872e+02	4.021113e+02
14	13	998673.3	6.633910e+02	6.632718e+02
15	14	997814.9	1.091867e+03	1.093258e+03
16	15	996404.3	1.795093e+03	1.800573e+03
17	16	994091.8	2.945857e+03	2.962380e+03
18	17	990314.0	4.819911e+03	4.866127e+03
19	18	984178.7	7.847904e+03	7.973400e+03
20	19	974309.1	1.267808e+04	1.301286e+04
21	20	958671.4	2.022562e+04	2.110294e+04
22	21	934479.0	3.163843e+04	3.388257e+04
23	22	898388.9	4.803551e+04	5.357564e+04
24	23	847336.1	6.983549e+04	8.282843e+04
25	24	780197.6	9.569889e+04	1.241035e+05
26	25	699672.3	1.217566e+05	1.785711e+05
27	26	612690.8	1.423622e+05	2.449470e+05
28	27	528140.8	1.526635e+05	3.191957e+05
29	28	453289.5	1.510989e+05	3.956115e+05
30	29	391691.2	1.396686e+05	4.686402e+05
31	30	343504.0	1.222181e+05	5.342779e+05
32	31	306976.3	1.025316e+05	5.904921e+05
33	32	279767.5	8.333457e+04	6.368979e+05
34	33	259671.4	6.615979e+04	6.741688e+05
35	34	244876.4	5.162311e+04	7.035005e+05
36	35	233988.7	3.977047e+04	7.262408e+05
37	36	225969.4	3.035315e+04	7.436774e+05
38	37	220054.7	2.300618e+04	7.569391e+05
39	38	215686.2	1.734880e+04	7.669650e+05
40	39	212455.4	1.303337e+04	7.745112e+05
41	40	210063.4	9.764101e+03	7.801725e+05
42	41	208291.0	7.299789e+03	7.844092e+05
43	42	206976.6	5.449078e+03	7.875743e+05

```
44   43 206001.5 4.062958e+03 7.899355e+05
45   44 205277.7 3.026881e+03 7.916954e+05
46   45 204740.3 2.253599e+03 7.930061e+05
47   46 204341.2 1.677089e+03 7.939817e+05
48   47 204044.7 1.247629e+03 7.947076e+05
49   48 203824.5 9.279051e+02 7.952476e+05
50   49 203660.9 6.899838e+02 7.956491e+05
51   50 203539.3 5.129945e+02 7.959477e+05
52   51 203449.0 3.813651e+02 7.961697e+05
53   52 203381.8 2.834883e+02 7.963347e+05
54   53 203332.0 2.107192e+02 7.964573e+05
55   54 203294.9 1.566226e+02 7.965485e+05
56   55 203267.3 1.164101e+02 7.966163e+05
57   56 203246.9 8.652005e+01 7.966666e+05
58   57 203231.6 6.430357e+01 7.967041e+05
59   58 203220.3 4.779117e+01 7.967319e+05
60   59 203211.9 3.551861e+01 7.967526e+05
61   60 203205.7 2.639740e+01 7.967679e+05
```

γ= 1, 365 días

```
# 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 = 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
```

```r
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))

#graficos de la evolucion 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 1.0000000  0.0000000
2     1 999998 0.9999985  0.9999993
3     2 999997 0.9999960  1.9999965
4     3 999996 0.9999925  2.9999908
5     4 999995 0.9999880  3.9999811
6     5 999994 0.9999825  4.9999665
7     6 999993 0.9999760  5.9999458
8     7 999992 0.9999685  6.9999181
9     8 999991 0.9999600  7.9998824
10    9 999990 0.9999505  8.9998376
11   10 999989 0.9999400  9.9997829
12   11 999988 0.9999285 10.9997173
13   12 999987 0.9999160 11.9996396
14   13 999986 0.9999025 12.9995489
15   14 999985 0.9998880 13.9994443
16   15 999984 0.9998725 14.9993246
17   16 999983 0.9998560 15.9991890
18   17 999982 0.9998385 16.9990363
19   18 999981 0.9998200 17.9988657
```

20	19	999980	0.9998005	18.9986760
21	20	999979	0.9997800	19.9984664
22	21	999978	0.9997585	20.9982358
23	22	999977	0.9997360	21.9979831
24	23	999976	0.9997126	22.9977075
25	24	999975	0.9996881	23.9974079
26	25	999974	0.9996626	24.9970833
27	26	999973	0.9996361	25.9967327
28	27	999972	0.9996086	26.9963552
29	28	999971	0.9995801	27.9959496
30	29	999970	0.9995506	28.9955151
31	30	999969	0.9995202	29.9950506
32	31	999968	0.9994887	30.9945551
33	32	999967	0.9994562	31.9940276
34	33	999966	0.9994227	32.9934671
35	34	999965	0.9993883	33.9928727
36	35	999964	0.9993528	34.9922433
37	36	999963	0.9993163	35.9915780
38	37	999962	0.9992789	36.9908756
39	38	999961	0.9992404	37.9901353
40	39	999960	0.9992009	38.9893561
41	40	999959	0.9991605	39.9885369
42	41	999958	0.9991190	40.9876767
43	42	999957	0.9990766	41.9867746
44	43	999956	0.9990331	42.9858296
45	44	999955	0.9989887	43.9848406
46	45	999954	0.9989433	44.9838066
47	46	999953	0.9988968	45.9827267
48	47	999952	0.9988494	46.9815999
49	48	999951	0.9988010	47.9804252
50	49	999950	0.9987516	48.9792016
51	50	999949	0.9987011	49.9779280
52	51	999948	0.9986497	50.9766035
53	52	999947	0.9985973	51.9752272
54	53	999946	0.9985439	52.9737979
55	54	999945	0.9984895	53.9723147
56	55	999944	0.9984342	54.9707766
57	56	999943	0.9983778	55.9691827
58	57	999942	0.9983204	56.9675319
59	58	999941	0.9982621	57.9658232
60	59	999940	0.9982027	58.9640557
61	60	999939	0.9981423	59.9622283

Realizando las modificaciones en γ (tasa de recuperacion), dejando el valor de β constante (tasa de infeccion) encontramos 4 casos distintos en la relacion de γ y β.

- En el primer caso, tenemos que el valor de $\gamma < \beta$ pero alejado de beta. En este caso la epidemia ocurrre y mucho mas fuerte que en otros casos, ejemplo es la grafica 1. En donde $\gamma = 0.025$ y $\gamma < \beta$. En el caso de los susceptibles la curva desciende a mayor velocidad, en los infectados el punto máximo de la curva es más grande y en los recuperados la curva aumenta más lento.
- En el segundo caso, tenemos que el valor de $\gamma < \beta$ pero cercano de beta. En este caso la epidemia ocurrre pero con menos fuerza que en otros casos, ejemplo de esto es la grafica 3. En donde $\gamma = 0.5$ y $\gamma < \beta$. En el caso de los susceptibles la curva desciende más lento, el punto máximo de infectados es menor y la curva de recuperados aumenta a mayor velocidad.
- En el tercer caso, tenemos que el valor de $\gamma = \beta$. En este caso la infeccion se encuentra en equilibrio, ya que el ritmo de infeccion es exactamente el mismo que el de recuperacion. En este caso no existe curva de infectados y los susceptibles y recuperados son una linea recta.
- El cuarto caso no esta presente en ninguna de las graficas que analizamos, a pesar de esto en este caso tenemos que $\gamma > \beta$. Cuando esto sucede la infeccion se encuentra controlada y podriamos apreciar un comportamiento similar a cuando $\gamma = \beta$.

El coeficiente de reproducción R_0 nos indica el número de casos secundarios que generará los casos primarios. R_0 es definido por $\frac{\beta}{\gamma}$, y para que una epidemia ocurra este coeficiente tiene que se mayor a 1. Esto lo podemos ver en los primeros 3 ejemplos analizados en los que el coeficiente $R_0 > 1$, también vemos que mientras más grande sea R_0 más grande es la curva de infectados y mas fuerte es la epidemia. Por último, en el último caso vemos como R_0 es igual a 1, por lo que la epidemia no ocurre