

PROYECTO ENFERMEDADES CONTAGIOSAS

MÉTODOS DE SIMULACIÓN POR ORDENADOR

Como objetivo de este proyecto se plantea el desarrollo de un modelo dinámico basado en el modelo SIR, un modelo de infección simple que representa la evaluación del desarrollo de una enfermedad contagiosa sobre una determinada población.

IDENTIFICACIÓN DEL PROBLEMA

Sabiendo la importancia de las enfermedades contagiosas y su avance sobre una determinada población se pretende estudiar el avance de una determinada enfermedad sobre una población respecto al avance del tiempo.

FORMULACIÓN Y DESARROLLO DEL MODELO

Utilizando el modelo epidemiológico SIR se definen varios estados posibles para la población estudiada:

VARIABLES DE ESTADO

- 1) **Susceptible (S):** Corresponde con los individuos sin ningún tipo de inmunidad al agente infeccioso y que por tanto pueden ser infectados en caso de estar expuestos al agente infeccioso.
- 2) **Infectado (I):** Corresponde con los individuos que ya están infectados por el agente infeccioso en un momento dado y además son capaces de extender la infección al resto de la población susceptible (S).
- 3) **Recuperado (R):** Corresponde con los individuos que son inmunes a la infección y consecuentemente no afectan a la transmisión del agente infeccioso al estar en contacto con el resto de población.



FUNCIONES DE CAMBIO DE ESTADO

Existen dos funciones de cambio de estado para la población:

- 1) **Susceptible a Infectado:** Los individuos susceptibles pasan a formar parte de los infectados (estado I) cuando tienen contacto con dichos individuos.
Los contactos entre susceptibles e infectados están estimados mediante el parámetro β para cada uno de los individuos.
- 2) **Infectado a Recuperado:** Los individuos infectados pasan a formar parte de los recuperados (estado R) cuando la infección desaparece. La probabilidad de recuperación se representa mediante el parámetro γ .

Mediante el parámetro γ podemos estimar el período de tiempo que un infectado tardará en recuperarse.

En el modelo SIR se establecen una serie de interacciones entre los elementos de población, pero en este proyecto se desarrollará la más simple de todas en la cuál se estima que la población no tiene entradas ni salidas, es decir, el número total de población es constante y se establecen en los posibles estados mencionados anteriormente: S, I, R.

La población a analizar corresponde con la suma de los tres estados, es decir:

$$N = S + R + I$$

N se corresponde con la población total

ECUACIONES DEL MODELO

En términos de los diferentes estados y funciones de cambio de estado mencionados anteriormente se presentan las ecuaciones diferenciales del modelo:

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

PARÁMETROS DEL MODELO

Como se puede observar en las ecuaciones planteadas así como en los parámetros mencionados anteriormente, existen varios parámetros que representan las diferentes probabilidades de producir un cambio de estado para cada miembro individual de la población:

$$\begin{aligned}\text{Tasa de transmisión} &= \beta \\ \text{Tasa de recuperación} &= \gamma \\ \text{Período medio de recuperación} &= \frac{1}{\gamma}\end{aligned}$$

Los resultados de la simulación dependen directamente de los parámetros utilizados por lo que lo más lógico sería obtener los parámetros de datos de experimentación reales. Es importante destacar que dichos parámetros tienen un gran significado ya que deciden la forma de transmisión y recuperación de los individuos de población.

Dichos parámetros deberían ser obtenidos mediante datos observación ya que si no se podría dar el caso de que los parámetros escogidos no representen las condiciones ideales para el modelo y por consiguiente los resultados obtenidos no sean fiables o ajustados a la realidad.

SIMULACIÓN

Para realizar la simulación del proyecto se ha utilizado como lenguaje Python ya que incorpora herramientas gráficas potentes además de un gran número de librerías de integración.

Las dependencias requeridas para ejecutar el proyecto son:

```
import numpy as np
from pylab import *
import scipy.integrate as spi
```

NUMPY

Se utiliza para generar rangos de intervalos (definidos mediante 3 parámetros: inicio, fin y steps). La función de numpy que genera los intervalos necesarios se conoce como:

```
arange()
```

Numpy es utilizado también a la hora de crear matrices ya que nos permite generar arrays con valores iniciales, para ello se utiliza la función:

```
np.zeros((3))
```

SPI

Se utiliza para integrar las ecuaciones diferenciales del modelo en base al tiempo. Para ello, se genera mediante la función:

```
modelo_SIR = spi.odeint(
    modelo.get_ecuaciones_diferenciales,
    poblacion_inicial,
    intervalos
)
```

Esta función permite integrar las ecuaciones generadas mediante el propio modelo SIR utilizando la población inicial generada y los intervalos marcados.

CLASES Y EJECUCIÓN

El código de la simulación se ha desarrollado utilizando dos clases auxiliares propias:

- 1) Graficar: Esta clase permite generar gráficas sencillas con los datos obtenidos mediante la simulación (integración de las ecuaciones diferenciales).
- 2) ModeloSIR: Siguiendo un diseño de POO se ha desarrollado esta clase que permite establecer las condiciones iniciales (población), parámetros (beta y gamma), obtener el array de intervalos y generar las ecuaciones diferenciales utilizadas por el método de integración de SPI.

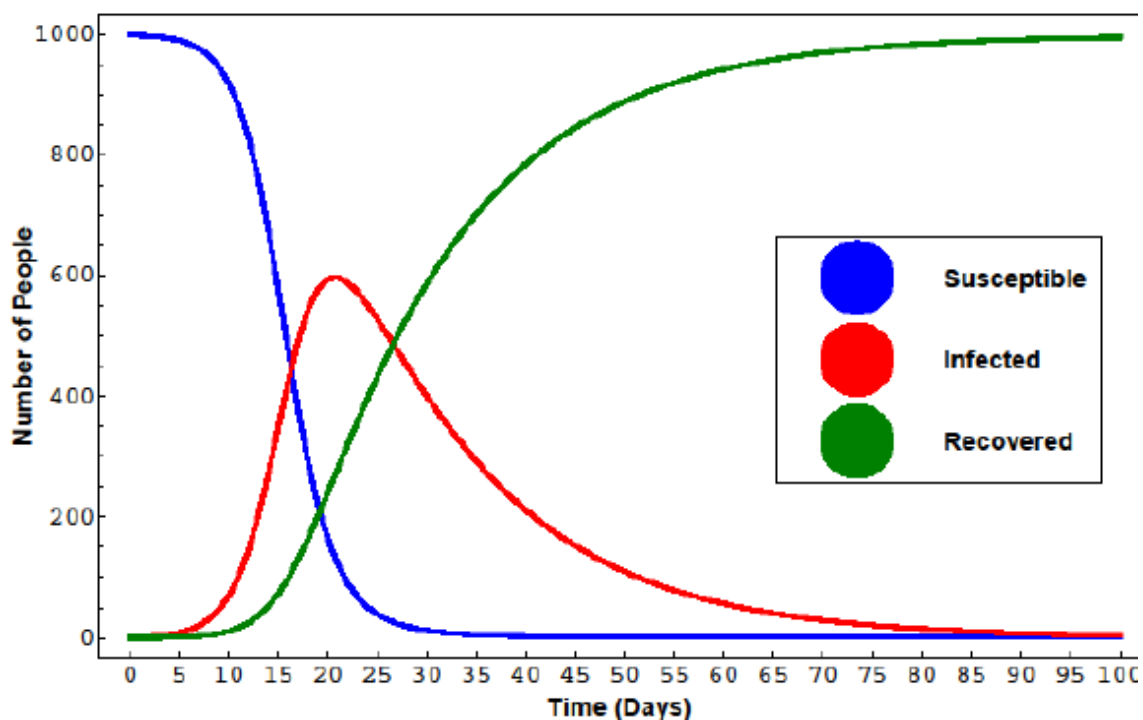
Para ejecutar la simulación se debe ejecutar el script Simulacion.py que se encarga de llamar generar los valores iniciales y graficar los datos mediante las dos clases creadas. *** Es necesario tener instaladas las librerías externas de Python mencionadas previamente: Numpy y SPI ***

Ejecución

```
python Simulacion.py
```

RESULTADOS OBTENIDOS

Mediante la siguiente gráfica (*Gráfica 1*) podemos ver que los resultados se ajustan a las curvas teóricas del modelo SIR (*Gráfica 2*), mediante el eje x se representan los distintos intervalos (100 días usados en la simulación de ejemplo), en el eje y podemos observar los datos de población (100.000 personas).



Gráfica 2: Curva teórica

Procedemos a la simulación estableciendo los siguientes valores de población iniciales:

- 1) S: 99900
- 2) I: 100
- 3) R: 0

Parámetros iniciales:

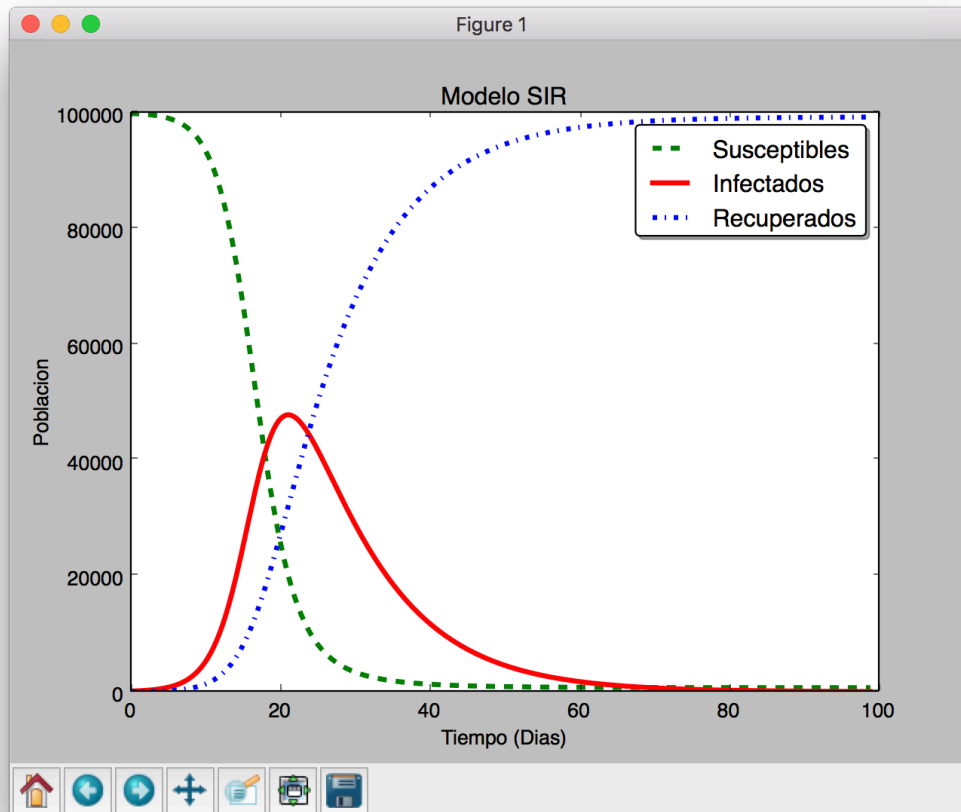
- 1) Beta: 0,50
- 2) Gamma: 0,10

Intervalos de integración (en días):

- 1) Inicio: 1
- 2) Fin: 100

3) Steps: 0,02

En color verde observamos la curva descendente del número de susceptibles, en color rojo podemos ver el crecimiento de la curva de infectados y su posterior descenso hasta normalizar los valores en 0. Por último, en color azul podemos observar la curva ascendente de individuos recuperados de la infección.



Gráfica 1: Resultados de simulación

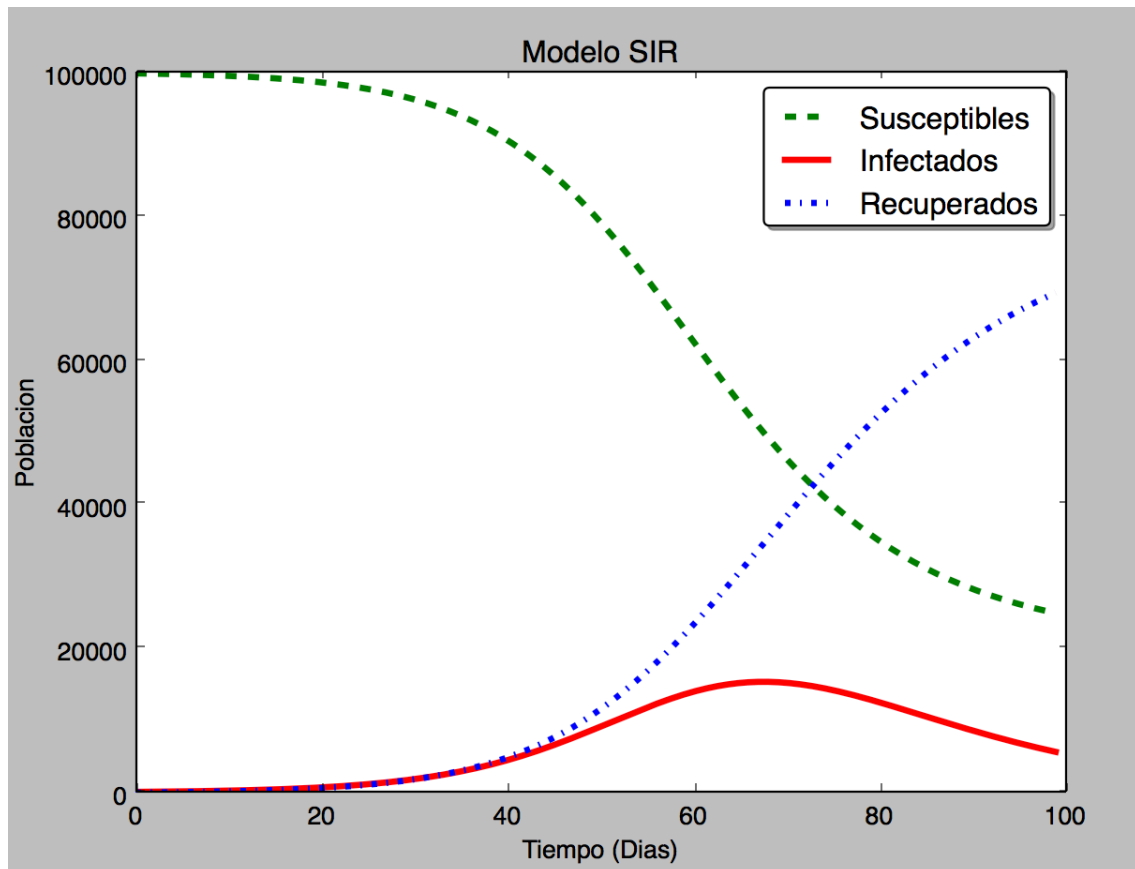
Observamos mediante esta gráfica que los resultados tienen sentido y se validan con el modelo original planteado por SIR, el número de susceptibles decrece a medida que avanza el tiempo pues el número de infectados aumenta a lo largo de dichos intervalos de tiempo.

También es posible ver (en color azul) que el número de recuperados aumenta a lo largo del tiempo pues los distintos individuos de la población analizada ya han pasado por los estados previos (S e I).

VARIACIÓN DE PARÁMETROS

Conociendo la alta dependencia de los parámetros en los resultados obtenidos mediante el modelo, repetimos la simulación anterior modificando el parámetro de transmisión de 0,50 a 0,20 y obtenemos unos resultados totalmente contrarios a la curva teórica.

Podemos observar los resultados de la nueva simulación mediante la siguiente gráfica (Gráfica 3):



Gráfica 3: Resultados de simulación con $\beta = 0,20$

Mediante esta gráfica podemos ver que realmente los resultados se siguen validando mediante el modelo, aunque ya no se adaptan a la curva teórica pues la elección de una tasa de contagio (β) de esta simulación está fijada en un valor muy bajo (0,20) y afecta al ritmo de contagio de forma mucho menor, por lo que el número de infectados apenas aumenta.