Spread of Communicable Diseases

Conor Selwood

December 12, 2014

1 Introduction

Looking at the rate at which a disease spreads can be an important task for helping manage outbreaks of deadly diseases in developed and developing countries. The first attempts of finding a mathematical model to help explain a diseases behaviour were in the 20th century by William Hamer and Ronald Ross. Since then it has been developed and improved to look at those at risk, those already infected and those vaccinated. In this report I will attempt to show a way to model the relationship between these groups and make a way to input your own data to look at the relationship and find exact values. More information about the history of modelling the spread of disease can be found at [1]

2 Spread of Diseases using sage

2.1 Differential equations showing spread of disease

The change in the amount of people susceptible to a disease, the amount infected and those who have recovered can be modelled using three differential equations;

$$\frac{dS}{dt} = -\alpha SI \qquad (1) \qquad\qquad \frac{dI}{dt} = \alpha SI - \beta I \qquad (2) \qquad\qquad \frac{dI}{dt} = \alpha SI - \beta I \qquad (3)$$

Equation (1) shows us the rate of change of those susceptible to infection, as you can see it depends on the current number of people susceptible, the current number of people infected and an infection constant α. There is a negative sign as the number of susceptible is always decreasing. The rate of change of people infected is shown by equation (2) this depends on the number susceptible at any given time, the number already infected at any time, the infection constant α and a new recovery constant β. Finally the rate at which people recover from the disease is shown in equation(3) only depends on the number of people infected and the recovery constant β. I found these equations and much more information explaining them at [4]

2.2 Class defining Diseases

There are many diseases in the world with different infection rates and recovery rates, also depending when we start recording the information the number of initial people belonging to each group may vary therefore I have created a class that allows you to input all these variables yourself

```
class Disease():
"""
A class that shows the spread of disease over time

Attributes:
```

```
alpha: The infection rate of a disease
beta: The recovery rate of a disease
initialsusceptible: The initial number of people susceptible to infection
initialinfected: The initial number of people infected
initialrecovered: The initial number of people who have recovered from infection
endpoints: The amount of time the data is recorded over

Methods:
graph: shows a graph of groups over time
Information: gives the number of people in the groups at any given time
"""

 def __init__(self, alpha, beta, initialsusceptible, initialinfected, initialrecovered, endpoints):
      self.alpha = alpha
      self.beta = beta
      self.initialsusceptible = initialsusceptible
      self.initialinfected = initialinfected
      self.initialrecovered = initialrecovered
      self.endpoints = endpoints
```

2.3 Relationship between groups

The first method in this class shows us a graph of all three groups against time this can easily show us the standard relationship between variables.

```
 def graph(self):
   """
   Returns a graph that shows how the three groups change over time

   Arguments:N/A

   Output: Graph of Susceptible, infected and recovered over time
   """
      t,S,I,R = var('t,S,I,R')  # setting the variables for the equation
      alpha = self.alpha
      beta = self.beta
      de1 = -alpha * S * I
      de2 = alpha * S * I - beta * I
      de3 = beta * I
      P = desolve_system_rk4([de1, de2, de3], [S,I,R], ivar=t, ics=[0, self.initialsusceptible,
self.initialinfected, self.initialrecovered], step=0.1, end_points= self.endpoints)  # solves the
three differential equations with initial conditions set
      p = list_plot([[t,S] for t,S,I,R in P], plotjoined=True, color='blue',legend_label=
'Susceptible')  # plot all the points of time against those susceptible
      p += list_plot([[t,I] for t,S,I,R in P], plotjoined=True, color='red',legend_label=
'Infected')  # plot all the points of time against those infected
      p += list_plot([[t,R] for t, S,I,R in P], axes_labels=['time(days)','number of people'],
    axes=True, plotjoined=True,color='green',legend_label='Recovered')  # plot all the points of
time against those recovered and set axes
      show(p)  # show graph
```

Parts of this code and much more information about the modelling of spread of diseases can be found here [3] . As an example if we have a disease class where $\alpha = 0.03$, $\beta = 0.07$, initialsusceptible $= 20$,
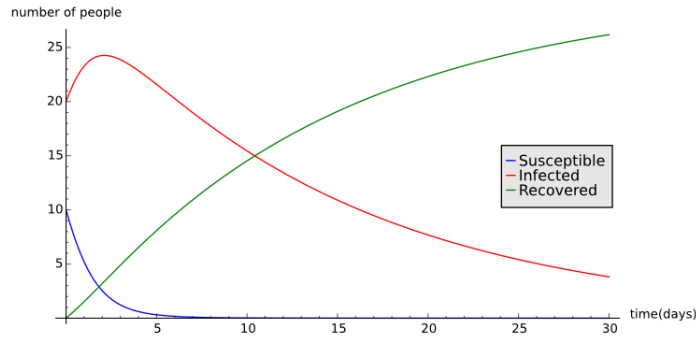
Figure 1: Graph with our particular variables

initialinfected $= 10$ and initialrecovered $= 0$. Looking at the information spread over 30 days we have the graph:

2.4 Exact values at any time

Sometimes it is more useful to have estimated results (at any time) of the number of people in each group for more calculations, we can obtain that data with the following method:

```
def Information(self, time):
        """

        Returns the number of people in each group at any given time

        Argument: Time in days

        Output: number of susceptible, infected and recovered
        """
        time = int(time)
        t,S,I,R = var('t,S,I,R')
        alpha = self.alpha
        beta = self.beta
        de1 = -alpha * S * I
        de2 = alpha * S * I - beta * I
        de3 = beta * I
        P = desolve_system_rk4([de1, de2, de3], [S,I,R], ivar=t, ics=[0, self.initialsusceptible,
self.initialinfected, self.initialrecovered], step=1, end_points= self.endpoints)
        s = [[t,S] for t,S,I,R in P]
        i = [[t,I] for t,S,I,R in P]
        r = [[t,R] for t,S,I,R in P]
        if time > self.endpoints or time < 0:  # makes sure reasonable time is entered
            print 'Time must be between 0 and %i' % (self.endpoints)
        else:
            x = [row for row in P[time]]
            return 'After %i days the number of people who are susceptible is %i, The number of people
of people who have recovered is %i' % (time, round(x[1]), round(x[2]), round(x[3]))
 # outputs information
```

3

Using the same example as last time we can calculate an approximation of the number of people in each group at any different time interval.

Time	Susceptible	Infected	Recovered
0	20	10	0
5	1	22	7
14	0	12	18
30	0	4	26

3   Conclusion

So in conclusion in this report I have looked at what factors contribute to the speed and severity at which a disease and the rate at which people can recover from different diseases. We can see from the code that the higher the infection rate, α the faster the number of susceptible falls and the number of infected rises, and the higher the recovery rate, β the slower the infection rate rises and the faster the recovery rate rises. In future I believe this report can be improved my taking into account Birth and Death rates and also the mortality rates of a disease can include a new group of those individuals who have died from the disease. My workings can be found on a sage work sheet at [2]

References

[1] Wikipedia: Mathematical modelling of infectious disease. `http://en.wikipedia.org/wiki/Mathematical_modelling_of_infectious_disease`. Accessed: 11-12-2014.

[2] Selwood C. Sage: Workings. `https://cloud.sagemath.com/projects/57d08055-34de-4d80-9895-505c3fd1ce52/files/Workings.sagews`. Accessed: 11-12-2014.

[3] Verónica Orjuela Contreras. Analytical and numerical simulation of epidemic models using maple and sage.

[4] Smith D. and Moore L. Mma: The sir model for spread of disease - the differential equation model. `http://www.maa.org/publications/periodicals/loci/joma/the-sir-model-for-spread-of-disease-the-differential-equation-model`. Accessed: 11-12-2014.