

# Contents

---

title: "Matrix models for SIR processes"  
author: Neil McRoberts, Robin Choudhury  
output: html\_document

---

## Introduction

You might have seen the excellent video on YouTube by 3Blue1Brown which uses simulation to look at some general concepts about social isolation and disease transmission, motivated by the early evidence of how the COVID-19 epidemic was developing in the USA (<https://www.youtube.com/watch?v=gxAaO2rsdIs>). In the commentary, the narrator refers to disease transmission, not in terms of rates, but in terms of *the probability of infection* from contact between infected and susceptible individuals. In the first section of the material on *SIR* models we will follow that lead and look at how we can use projection matrices to build models of *SIR*-type models.

## SIR models as state diagrams

We already looked at a state diagram for the basic *SIR* model when we first looked at using DiagrammeR. The code is repeated below for convenience:

```
# SIR

grViz("
  digraph SIR {
    graph [rankdir = LR]

    #nodes
    node [shape=circle]
    S;I;R

    #edges
    S->I [label=pIS] [fontsize=10]
    I->R [label=pRI] [fontsize=10]
    S->S [label='1-pIS'] [fontsize=10]
    I->I [label='1-pRI'] [fontsize=10]
    R->R [label=1] [fontsize=10]
  }

")
```

The diagram represents the process of the epidemic as a series of transitions from one state to the next, with each transition occurring with a fixed probability in the interval of time corresponding to model time steps. By convention, (and counter-intuitively) the subscripts on the probabilities are in *to-from* order, so  $p_{SI}$ , for example, means "the probability of moving **to** Infectious **from** Susceptible". Remember that the figure includes pathways from each state back to itself. These allow for the possibility that individuals stay in the same state in the time period covered by one iteration of the model. We draw in these *self loops* for completeness, but they can be left off for clarity in complex state diagrams, as long as you remember that the probability of staying in a state might be greater than zero, and remember to include the values when you build the projection matrix

### The SIR projection matrix

$$\mathbf{A} = \begin{pmatrix} p_{SS} & 0 & 0 \\ p_{IS} & p_{II} & 0 \\ 0 & p_{RI} & p_{RR} \end{pmatrix}$$

Again, as a reminder, the probabilities on the self loops go on the main diagonal of the matrix and the probabilities for moving one state at a time through the epidemic states go on the first sub-diagonal. The rows and columns of the matrix are both in the same order as the states in the transition diagram (*SIR*) and the matrix is "*read*" so that the transitions are down the columns.

This basic *SIR* model represents what we would expect to happen under constant conditions if there was no intervention. You can think of this as the "*spine*" of all the subsequent models that we might build from this one to investigate the impact of different intervention policies, or for diseases with different numbers of states or dynamics.

Since we project the dynamics of the *SIR* matrix in exactly the same way as we would any other matrix model the structure of the code needed to do the numerical projection should be familiar. The basic *SIR* model represents a *closed* system; *i.e.* one in which no-one enters or leaves, so it is convenient to model the process using the population scaled to sum to 1 (so that the output represents proportions) or 100 (so that the output represents percentages). Some simple R code to do this might look like this:

```
# Simple matrix projection model for SIR process
# Assume 3 states as in standard SIR model: Susceptible, Infected, Removed
# There is no immigration or emigration and no mortality. We assume that
# it is possible to define a time interval such that meaningful
# probabilities can be assumed for the state transitions.

# Start by listing the probabilities
```

```

p_is <-0.25
p_ss <-1-p_is
p_ri <-0.3
p_ii <-1-p_ri
p_rr <-1

# Now define a matrix to take the transition probabilities

A <-matrix(nrow=3, ncol=3,
           data=c(p_ss,p_is,0,
                  0,p_ii,p_ri,
                  0,0,p_rr))
#print A to make sure it has been constructed correctly
A
#check the column entries of each column of A sum to 1
#There are prettier, more efficient ways to implement
#this for larger matrices.

sum(A[,1])
sum(A[,2])
sum(A[,3])

# Now we need a matrix to hold the output. Declaring it
# in advance is more efficient than trying to dynamically
# add columns during calculation. Each column of the output
# is a time step, so the width of this matrix defines the
# total duration of the projection.

projection <-matrix(nrow=3, ncol=21)

# Now we define the initial states and assign them to the first column of
# the output matrix

initial <-c(100,0,0) #We assume 100% of the population is susceptible at the start
projection[,1]<-initial

# The numerical projection of the SIR model is now performed using a
# "for" loop to iterate a matrix multiplication

for (i in seq(2,21,1)) {
  projection[,i]<-A*%projection[,i-1]
}

# Note that the iteration starts at time 2 and works by
# calculating the "current" value from the immediate past

```

```

# value. That is, compared with the way we would typically
# write the equation as a projection of the present value to
# the immediate future, we are calculating with the
# pair of time index values slid backwards by one step.

# Make a vector to hold the time steps and generate a plot of
# the numerical output
t<-seq(1,21,1)
graph1<-plot(t,projection[1,], ty="l", lwd=3, col="darkblue",
             xlab="time", ylab="individuals")
lines(t,projection[2,], lwd=3, lty=2, col="red")
lines(t,projection[3,], lwd=3, lty=3, col="black")
legend("right", inset=0.015, legend=c("S","I","R"),
      lty=c(1,2,3), col=c("darkblue","red","black"), lwd=c(3,3,3))

```

The dynamics displayed by numerical projection of the matrix show a pattern that has become familiar to many of us over the last couple of years, thanks to COVID-19. The susceptible (healthy) fraction of the populations declines by a fixed proportion (the transition probability  $p_{IS}$ ) each time step. This generates a trajectory that looks like an exponential decay curve. The increase in the removed state is basically a slightly delayed mirror image of the decay curve for the susceptible state, with the delay caused by the time spent in the infectious state.

### Making an SIRS model

To make the *SIR* model into a *SIRS* model relatively few changes are needed. In the graphical model we need to add an edge that starts in *R* and goes to *S*. In the Graphviz code this is as simple as adding an additional edge statement to the set of edge declarations. Associated with the new edge in the graph, we need to change one of the zero entries in the projection matrix to hold the value of the probability on the edge,  $p_{SR}$ . Remembering that we read the projection matrix “from-to” down the columns, we see that the new probability value must go in the top, right cell of the matrix in the row corresponding to *S* in the column corresponding to *R*. All that remains to do is to adjust the value of  $p_{RR}$ , in the bottom right cell because we had previously set it to 1, because *R* was an absorbing state. Now, the requirement is that the column under *R* in the matrix should sum to 1; *i.e.*  $p_{RR} + p_{SR} = 1$ .

These changes are included in the R script file supplied for the lecture. You will be able to see that the net result is that the population stabilizes with some individuals in each of the three states, instead of ending up with all individuals in *R*. This is not surprising. The *SIRS* model is appropriate in public health for seasonal diseases or recurring diseases with a transient period of immunity that degrades over time. We could also apply it plant health, where treatment with pesticides would put plants into *R*, but then plant growth and degradation

of the active ingredient of the pesticide would cause the plants to move in the  $S$  category.

Written with StackEdit.