

# The Power of Math

Let us make a simple table with Age, Weight and Gender for some students.

Name	Age	Weight	Gender
Happy	12	1010	M
Sad	34	1000	F
Elated	123	9	F

Clearly, tables are a fantastic way of expressing information in a neat and tidy way.

Well, in Maths, you can also express a table into an easier format.

We can do this:

$$Information = \begin{pmatrix} 12 & 1010 & M \\ 34 & 1000 & F \\ 123 & 9 & F \end{pmatrix}$$

This is exactly the same as a table, albeit written in Mathematical notation.

Also, since Maths is mostly numbers, we need to change the male and female to a numerical system:

$$M = 1, \quad F = 2$$

So,

$$Information = \begin{pmatrix} 12 & 1010 & 1 \\ 34 & 1000 & 2 \\ 123 & 9 & 2 \end{pmatrix}$$

Now, we call this structure in Maths a “matrix”. Matrices for plural. In normal day life, a table.

So what can we do with this “matrix”??

## 1. Addition

We all know that  $1+1 = 2$

So what is a matrix + another matrix? A new matrix!

But how do we add “tables”??

Can we do this?

$$\begin{pmatrix} 12 & 1010 & 1 \\ 34 & 1000 & 2 \\ 123 & 9 & 2 \end{pmatrix} + 3?$$

No of course! Why? What are you adding the 3 to ?

Weight, age, gender? And for who?

Pretend we wanted to adjust Happy’s weight by 10090kg. To do this, we need to add like this:

$$\begin{pmatrix} 12 & 1010 & 1 \\ 34 & 1000 & 2 \\ 123 & 9 & 2 \end{pmatrix} + \begin{pmatrix} 0 & 10090 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Which gives our desired matrix.

Subtraction is the same.

## 2. Multiplication and Division

Now, we know  $4 \times 8 = 32$ .

But what is a matrix times another matrix?

How do we “times” a table???

Is

$$\begin{pmatrix} 12 & 1010 & 1 \\ 34 & 1000 & 2 \\ 123 & 9 & 2 \end{pmatrix} \times 3?$$

Allowed?

YES in fact?

$$\begin{pmatrix} 12 & 1010 & 1 \\ 34 & 1000 & 2 \\ 123 & 9 & 2 \end{pmatrix} \times 3 = \begin{pmatrix} 36 & 3030 & 3 \\ 102 & 3000 & 6 \\ 369 & 27 & 6 \end{pmatrix}$$

How about:

$$\begin{pmatrix} 12 & 1010 & 1 \\ 34 & 1000 & 2 \\ 123 & 9 & 2 \end{pmatrix} \times \begin{pmatrix} 0 & 10090 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}?$$

We can do this strangely enough, but the times for matrices has multiple meanings:

1. Cross multiplication
2. Dot multiplication
3. Element multiplication

Now, in our model, we focus on 1. And 3.

1. Cross multiplication

$$\begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix} \times \begin{pmatrix} 5 & 7 \\ 6 & 8 \end{pmatrix} = \begin{pmatrix} 1(5+7) & 3(6+8) \\ 2(5+7) & 4(6+8) \end{pmatrix} = \begin{pmatrix} 12 & 42 \\ 24 & 56 \end{pmatrix}$$

## 2. Element multiplication (NOTICE .x or DOT X)

$$\begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix} \cdot \begin{pmatrix} 5 & 7 \\ 6 & 8 \end{pmatrix} = \begin{pmatrix} 1 \times 5 & 3 \times 7 \\ 2 \times 6 & 4 \times 8 \end{pmatrix} = \begin{pmatrix} 5 & 21 \\ 12 & 32 \end{pmatrix}$$

Also, we need to know the “identity” matrix or  $I$

This is just:

$$I = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

All identity matrices MUST be “even” or looking like a box. 3 by 3. NOT 3 by 4.

We can also have:

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

In fact, we can have ANY size!

Some astounding mathematical facts:

What is (CROSS MULTIPLY)

$$\begin{pmatrix} 12 & 1010 & 1 \\ 34 & 1000 & 2 \\ 123 & 9 & 2 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

We get

$$\begin{pmatrix} 12 & 1010 & 1 \\ 34 & 1000 & 2 \\ 123 & 9 & 2 \end{pmatrix}$$

So, a rule:

$$\text{Matrix} \times I = \text{Matrix}$$

Also, what is

$$\begin{pmatrix} 12 & 1010 & 1 \\ 34 & 1000 & 2 \\ 123 & 9 & 2 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

(DOT TIMES - ELEMENT)

$$\begin{pmatrix} 12 & 0 & 0 \\ 0 & 1000 & 0 \\ 0 & 0 & 2 \end{pmatrix}$$

We deleted the unnecessary ones!

---

## INTRO TO MSIR MODEL

---

Markov Chains is very useful. It is used in finance, statistics, engineering, epidemiology and lots of other places.

HOWEVER, the knowledge of Markov Chains for epidemiologists is VERY limited unfortunately. Therein lies the problem of incorrect models.

In healthcare, researchers use the basic SIR or SEIR Models.

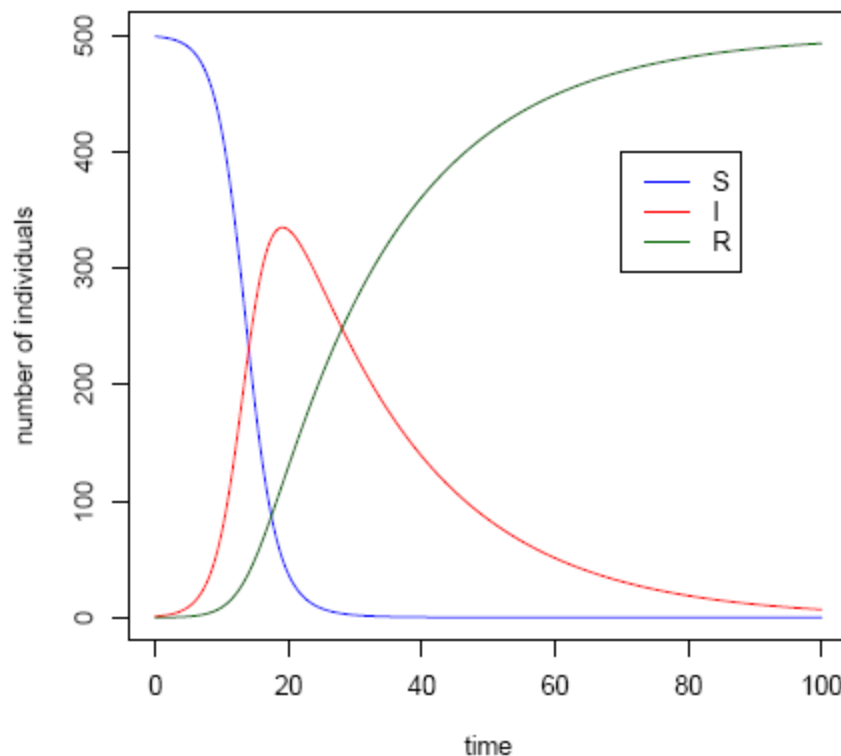
S = Susceptible

I = Infected

R = Recovered (Dead/Immune)

E = Infected BUT no symptoms (just transmitter - HIV)

These models are useful, as they monitor change.



However, there is a problem with this model.

It DOES NOT account for “real world” scenarios. This model is for an ENTIRE population. Meaning the infected ratios are in fact always the SAME everywhere.

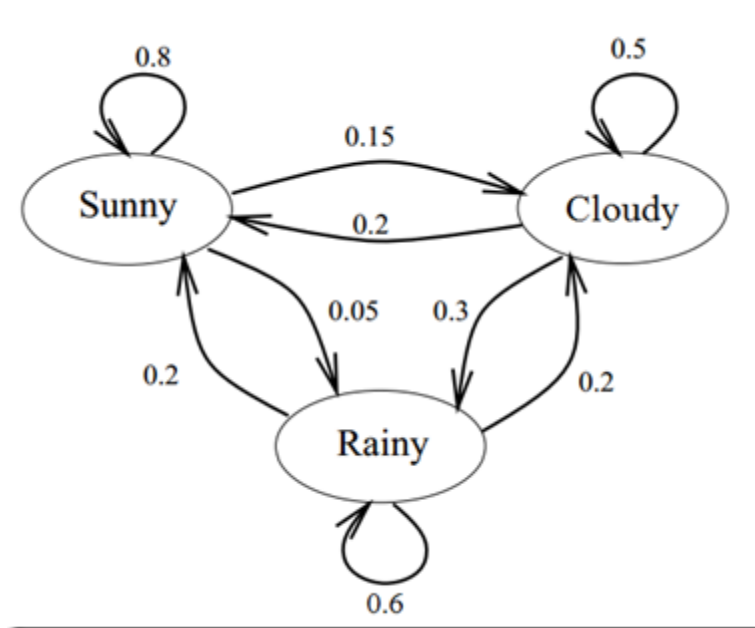
Now, I pose a question. If one place has NO infected, and so NO disease is ever there, how can it have infected people? The differential equations cannot account for this, as the SIR model is for an ENTIRE group of areas.

So, how do we account for locational spreading? (This is the largest factor in the Zika Hack)

Introducing Markov Chains! (MSIR = Markov SIR)

Now, I did try to search online about Markov Chains for SIR models, but I only found a “Greenwood Model”. It is not complete, albeit fails to even capture the power of maths. I am ashamed the Greenwood Model uses Markov Chains, but is super limited.

What is a Markov Chain?

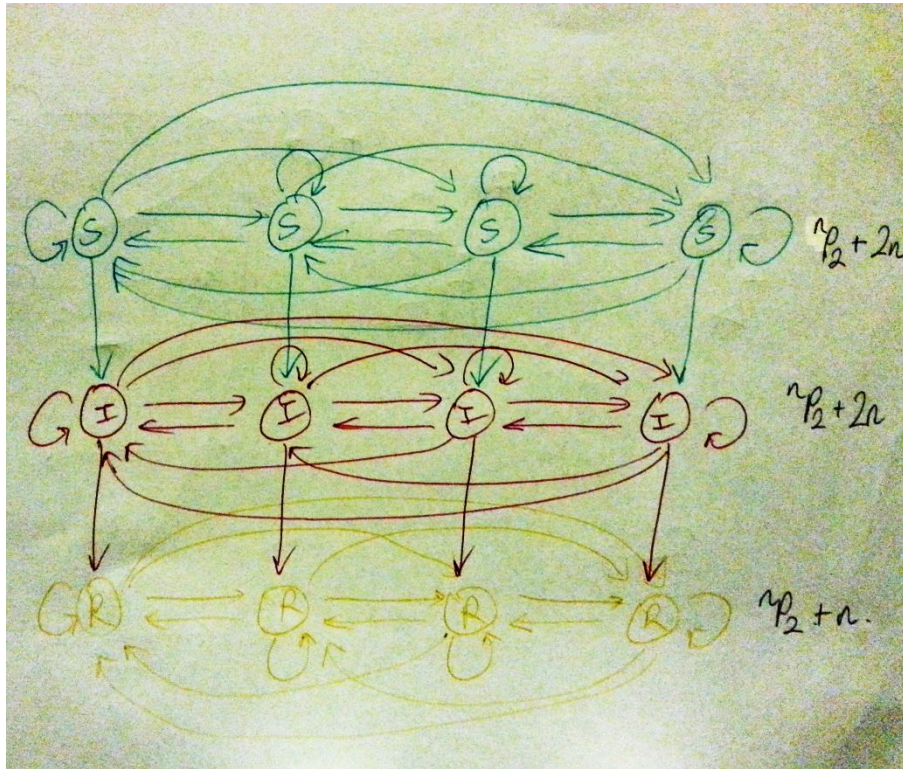


The diagram above is exactly that. Probabilities of whether something will occur. Eg: if it is sunny, how much probability will it be cloudy? 15%.

So, why not apply it to SIR models?

First, each area, let's call them A, B, C, D will have each 3 sets of nodes (S, I, R)

As you can see below:



Now, each node has probabilities of travelling to another node.

Every arrow depicts the pathways one can take.

Now, we can express this into a table.

From \ To		S				I				R			
		a	b	c	d	a	b	c	d	a	b	c	d
S	a	$S \rightarrow S$				$S \rightarrow I$				$S \rightarrow R$			
	b												
	c												
	d												
I	a	$I \rightarrow S$				$I \rightarrow I$				$I \rightarrow R$			
	b												
	c												
	d												
R	a	$R \rightarrow S$				$R \rightarrow I$				$R \rightarrow R$			
	b												
	c												
	d												

Or a Matrix!



Now, in a Markov Matrix, there is just one rule

1. ALL ROWS add to 1

Why?

These are probabilities. Pretend I can be infected, not infected. Say I can be infected 0.05%. So, the not infected chance is 99.95%.

That's why it must add to 1.

---

## MOSQUITO DISTRIBUTION

---

Firstly, we need to know how and why mosquitoes spread the Zika Virus.

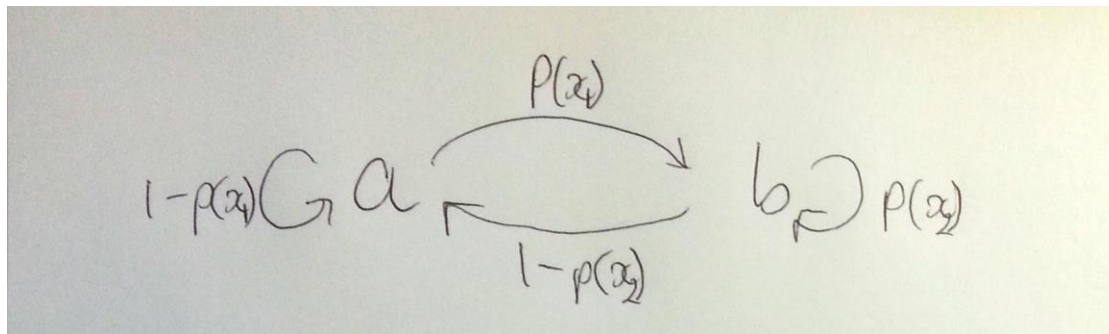
For example: People/mosquitoes move into spaces where it is suitable for survival (ie precipitation and temperature ranges)

We will use both temperature and precipitation as a scaling factor.

Remember, the matrix before is a steady-state humanistic movement one (humans moving).

Now, let us consider mosquito movement ONLY.

So, if we just take areas A and B into account, we can produce another Markov Diagram for Mosquito movement:



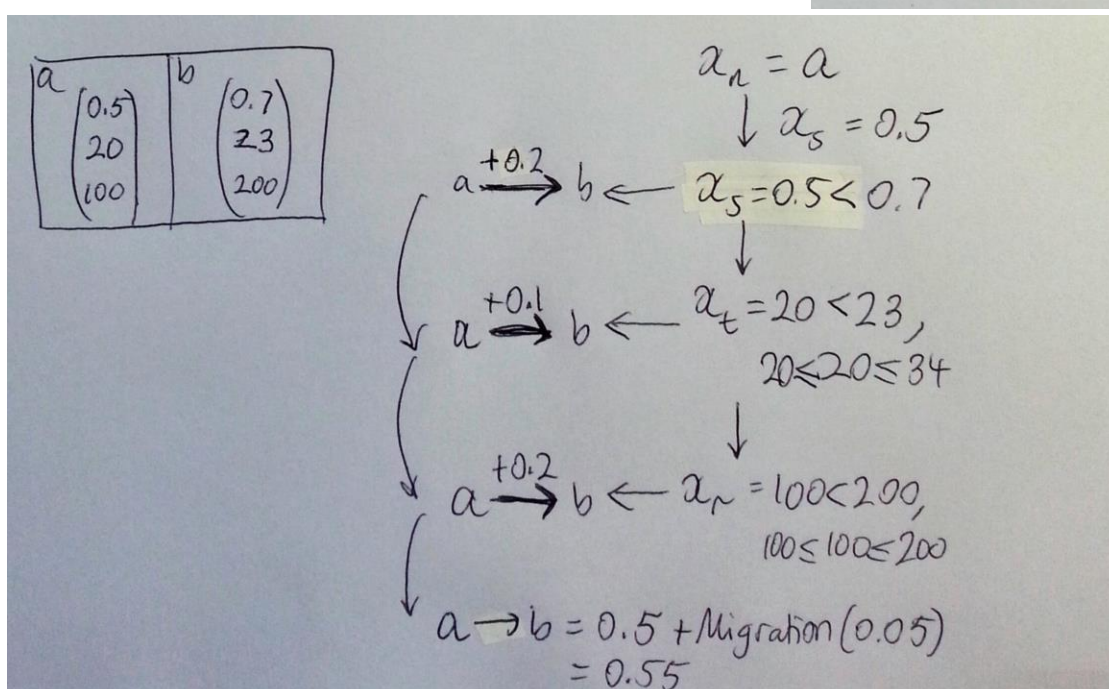
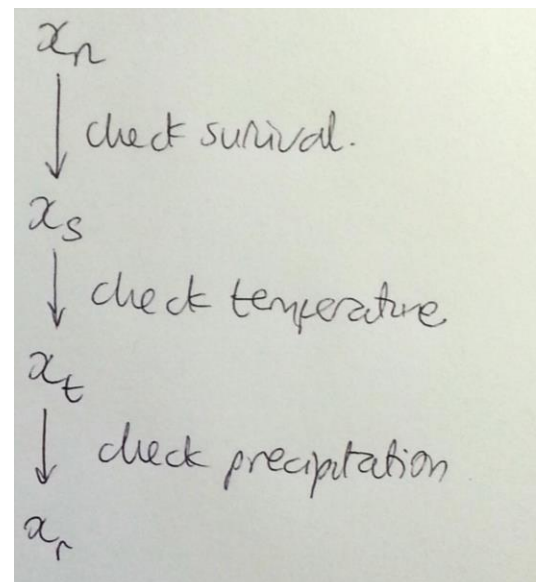
Let me guess that a mosquito's best survival is at:

Temperature : 20 - 34

Precipitation: 200 - 300

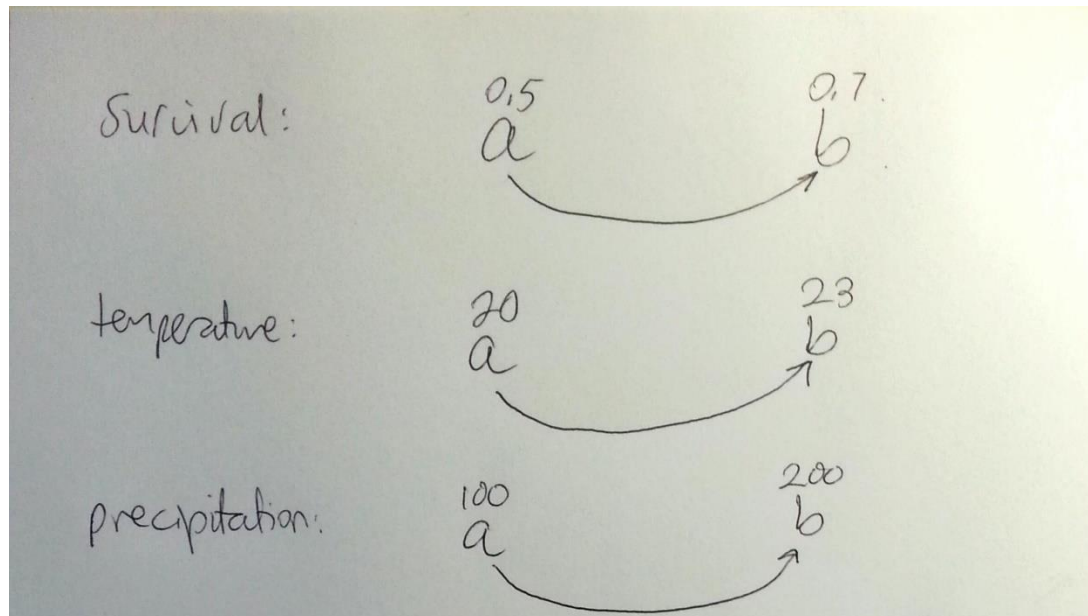
So, I can inspect each connectable node for mosquito movement, and produce a looping process to assign automatic probabilities of movement.

For example:



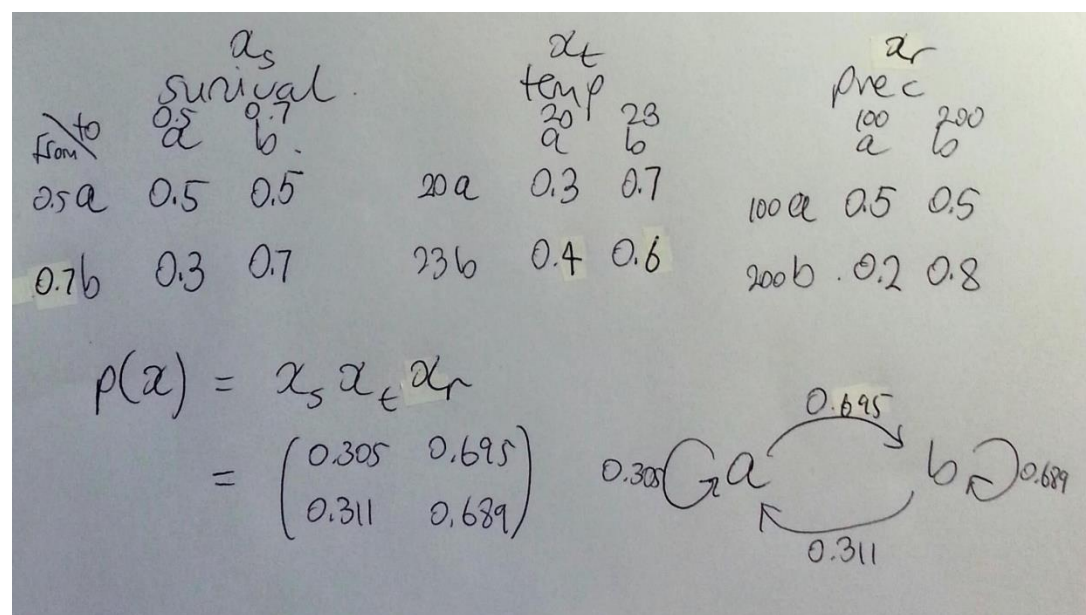
Now, on paper this is fine. We then utilise Markov Chains again (...)

(Seems like matrices are very useful ... I use to hate matrices)



Note - these are most probable transitions. The arrows represent the most likely scenario.

We then use an algorithm to compute each separate probability matrix, and then times them in order to produce  $p(x)$



So:

$$p(x) = x_s x_t x_r$$

Now, the issue arises when we want to compute each of these separate matrices.

We firstly need to know some rules:

1. ROWS need to add to 1
2. (Assumed): Mosquitos love 20-34C and 200-300mm rain.

So, how do we translate this into programming? We can utilise ratio analyses ☺

But first, we have 2 scenarios:

1. We can access probability data for mosquito abundance from researchers
2. We CAN'T access the data.

The first one is fantastic, but the second one requires ourselves to modify or create our own working probabilistic model.

Let's say we DON'T get access. Then, we can still apply ratios:

But first, we need to derive a method for finding a mosquito's best survival temperature and precipitation.

We assume 20-34 is the best. Let's guess 27 is the best. So let's apply the old "20-80 principle" by Pareto.

We will assume within the 20-34 range (which we will account for 20% of the temperature distribution), there exists 80% of all mosquitoes. (Purely just guessing, but with mathematical sense.)

So, if before 20 and after 34 is evenly spread, then:

$$missing + missing = 2missing = 0.8scale$$

$$34 - 20 = 14 = 0.2scale$$

$$So\ scale = 70$$

$$So\ missing = \frac{56}{2} = 28$$

So the temperature scale is from approx. -8 to 62 degrees.

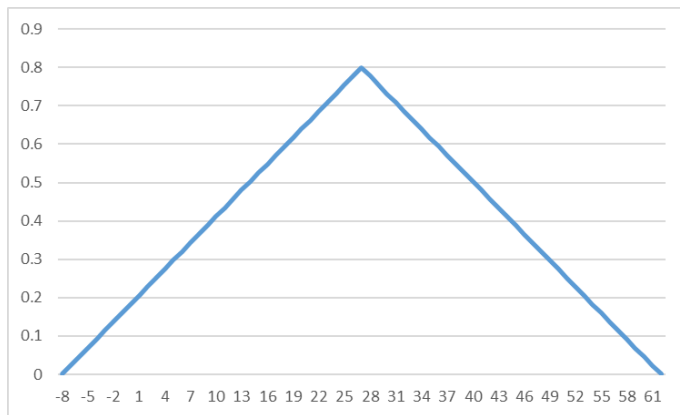
Now, I'm just going to assign a random guess:

- If temperature = 27, then  $t = 0.8$
- If  $20 < T < 34$ , then  $t = 0.8 - 0.2 \frac{|27-T|}{7}$
- Interestingly enough, we don't need to work it out, we can just use:

$$t = 0.8 - 0.16 \frac{|27 - T|}{7}$$

Now, the graph made is an upside down absolute value graph.

**We can implement a normal distribution curve later.**



As with precipitation, let's assume the same (mosquitoes love 200-300mm water)

So, with similar analyses:

$$missing = 0.8scale$$

$$300 - 200 = 100 = 0.2scale$$

$$So\ scale = 500$$

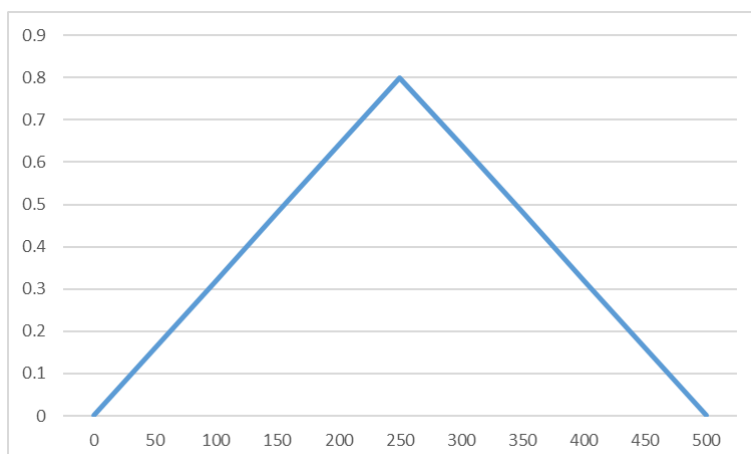
$$So\ missing = 0.8 * 500 = 400$$

So the precipitation rate is from approx. 0 to 500.

Now, I'm just going to assign a random guess:

➤ If precipitation = 250, then  $p = 0.8$

$$p = 0.8 - 0.16 \frac{|250 - T|}{50}$$



Yay!

We can also produce a normal distribution.

Take  $\sigma = 7, \mu = 27$ . So (\*13.5 for scaling)

$$t = 13.5 \left( \frac{1}{7\sqrt{2\pi}} e^{-\frac{(T-27)^2}{2 \times 7^2}} \right)$$

And

Take  $\sigma = 50, \mu = 250$ . So (\*100 for scaling)

$$r = 100 \left( \frac{1}{50\sqrt{2\pi}} e^{-\frac{(P-250)^2}{2 \times 50^2}} \right)$$

$$T = \begin{pmatrix} a = 34 \\ b = 23 \\ c = 25 \\ d = 28 \end{pmatrix} \quad \text{Then applying: } t = \begin{pmatrix} a = 0.466 \\ b = 0.6534 \\ c = 0.7386 \\ d = 0.7615 \end{pmatrix}$$

$$P = \begin{pmatrix} a = 340 \\ b = 230 \\ c = 250 \\ d = 280 \end{pmatrix} \quad \text{Then applying: } r = \begin{pmatrix} a = 0.1579 \\ b = 0.7356 \\ c = 0.7978 \\ d = 0.6664 \end{pmatrix}$$

So, now we have both temperature and precipitation matrices. We will then apply Markov chains in order to understand if a mosquito wants to stay in that place or not.

**ASSUMPTION: Mosquitoes like to stay where they are, and don't like moving.**

So, we first produce a  $n$  by  $n$  matrix that follows the size of all the nodes.

So for 4 boxes (4 by 4 matrix)

1. Fill diagonals with probabilities already found.

$$\begin{pmatrix} 0.466 & & & \\ & 0.6534 & & \\ & & 0.7386 & \\ & & & 0.7615 \end{pmatrix}$$

2. PROPERTY: all rows MUST add to 1.
3. So, let's use a ratio analysis and common sense.

We know that anyone wants to move to more favourable conditions (unless if they stay put which is more likely). So let's say a mosquito is in A (0.466).

They want to obviously move to B or C or D (higher chance of survival).

So, we use ratios:

$$T_{n,m} = (1 - t_{n,n}) \times \frac{t_{n,m}}{-t_{n,n} + \sum_{k=1}^{k=m} t_{n,k}}$$

```

14
15 ss = size(T)[2]
16 mat = eye(ss)
17
18 for uu in 1:ss
19     Q = copy(vec(T))
20     Q[uu] = 0
21     jj = 0
22
23     for x in Q
24         if x == 0
25             d = T[uu]
26         else
27             y = total-T[uu]
28             b = x/y
29             c = 1-T[uu]
30             d = c*b
31         end
32         jj += 1
33
34     mat[uu,jj] = d
35 end

```

> mat

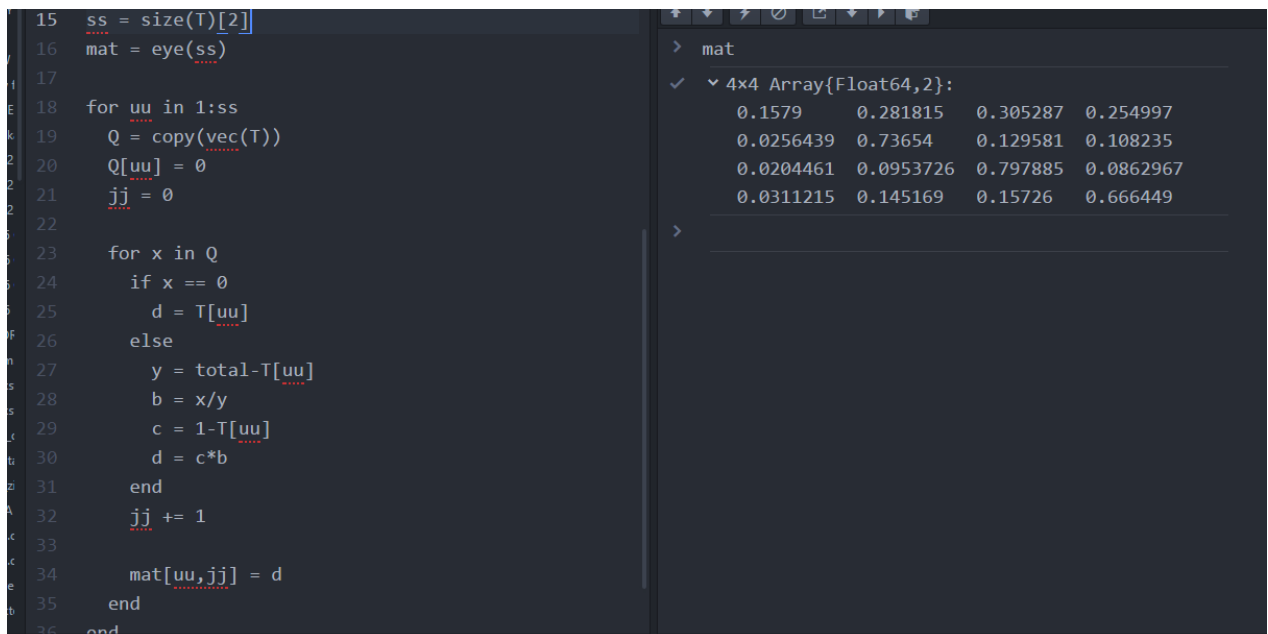
✓ 4x4 Array{Float64,2}:

0.466658	0.161832	0.182912	0.188598
0.0822128	0.653492	0.130125	0.13417
0.0648214	0.0907738	0.738617	0.105787
0.0598577	0.0838228	0.0947417	0.761578

>



After temperature data, we can do the same for precipitation data.



The image shows a MATLAB script on the left and the variable viewer on the right. The script processes precipitation data by iterating over rows of a matrix T, calculating a value d based on a total and a ratio, and storing it in a matrix mat. The variable viewer shows the resulting 4x4 matrix mat.

```

15 ss = size(T)[2]
16 mat = eye(ss)
17
18 for uu in 1:ss
19     Q = copy(vec(T))
20     Q[uu] = 0
21     jj = 0
22
23     for x in Q
24         if x == 0
25             d = T[uu]
26         else
27             y = total-T[uu]
28             b = x/y
29             c = 1-T[uu]
30             d = c*b
31         end
32         jj += 1
33     end
34     mat[uu,jj] = d
35 end
36 end
  
```

Variable viewer: mat

4x4 Array{Float64,2}:			
0.1579	0.281815	0.305287	0.254997
0.0256439	0.73654	0.129581	0.108235
0.0204461	0.0953726	0.797885	0.0862967
0.0311215	0.145169	0.15726	0.666449

YAY!

And now, we can apply our formula we derived:

$$p(x) = x_s x_t x_r$$

Or in our case (without probability data from external source):

$$p(x) = x_t x_r$$

So, in Julia, we find that:

```
> k*mat
✓ 4x4 Array{Float64,2}:
 0.0874447  0.29553  0.339037  0.277988
 0.0365756  0.53638  0.234703  0.192342
 0.0309572  0.170927  0.637519  0.160596
 0.0372397  0.198201  0.224494  0.540065
> |
```

YES!

We have just derived “hypothetical” yet methodologically accurate probability data for each box. This means a mosquito wants to migrate from A to A (or stay) 0.087 times, whilst they want to move to B 0.2955 times.

This makes sense, as A’s temp = 34, whilst the precipitation is 340. Both are not that optimal. But, B’s temp is 23 and the precipitation is 230. Way better.

CHECK to see if ALL rows add to 1: CORRECT!

---

But that’s not all!

We didn’t consider the growth rate of mosquitoes. Or do we??

We DON’T! Why? Because the probability model shows where there should be. HOWEVER, we must scale each result by a new factor of living standards.

- Higher **living standards** = more pesticides, bug spray eg eg and contraception, hygiene (meaning less mosquitoes and less infection)

So, we find that:

$$p(x) = x_s x_t x_r x_l$$

To find  $x_l$ , we apply ratios again.

If  $L = \begin{pmatrix} a = 0.46 \\ b = 0.55 \\ c = 0.33 \\ d = 0.65 \end{pmatrix}$ , then let us ASSUME mosquitoes LIKE to go to LESS developed areas.

Using:

$$T_{n,m} = (1 - t_{n,n}) \times \frac{t_{n,m}}{-t_{n,n} + \sum_{k=1}^m t_{n,k}}$$

WHICH IS WRONGG!!!!

WHY? Because this formula highlights if the mosquito wants to go to another place IF HAS HIGHER NUMBER. We want LOWER NUMBER.

So how do we fix this issue? Easy. Take the inverse.

$$L = \begin{pmatrix} a = 0.46 \\ b = 0.55 \\ c = 0.33 \\ d = 0.65 \end{pmatrix} \quad \text{becomes:} \quad L_{inv} = 1 - \begin{pmatrix} a = 0.46 \\ b = 0.55 \\ c = 0.33 \\ d = 0.65 \end{pmatrix} = \begin{pmatrix} 0.54 \\ 0.45 \\ 0.67 \\ 0.35 \end{pmatrix}$$

BUT, to fill our new  $x_t$  matrix, we use the new one as well! (As inverse)

```

1 p = [0.46 0.55 0.33 0.65]
2
3 T=1-p
4
5 total = sum(T)
6
7 ss = size(T)[2]
8 mat = eye(ss)
9
10 for uu in 1:ss
11     Q = copy(vec(T))
12     Q[uu] = 0
13     jj = 0
14
15     for x in Q
16         if x == 0
17             d = T[uu]
18         else
19             y = total-T[uu]
20             b = x/y
21             c = 1-T[uu]
22             d = c*b

```

mat

4x4 Array{Float64,2}:			
0.54	0.140816	0.20966	0.109524
0.190385	0.45	0.236218	0.123397
0.132985	0.110821	0.67	0.086194
0.211446	0.176205	0.262349	0.35

So

$$L_{inv} = 1 - L$$

Now, we understand that if an area is:

1. Temp between 20-34
2. Prec between 200 and 300
3. Quality of life is low

Mosquitoes are going to be seen more there.

---

## MARKOV MODELLING

---

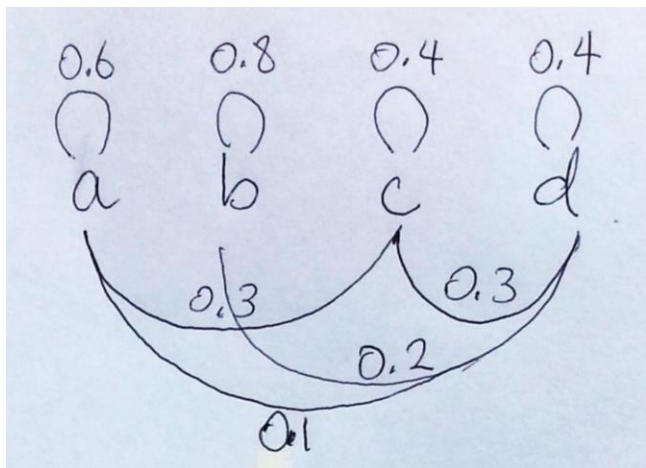
We first need to basic Markov Model to start this model up.

At first, I visualised a 2 by 2 area, where each area is connected by rail, road, plane, water.

So, A is connected to C and D, and B is connected to D only and so on.

Each connection is weighted according to how many connections or how important the connections are. Eg: Plane is less likely than road so plane is around 0.1 whilst road is 0.3.

I converted the probabilities into a Markov Chain diagram:



So, infected would stay in A 60% of the time, whilst they will move to C 30% of the time, and move into D 10% of the time.

The others are also produced like A. Note where all the probabilities for one node (eg A) adds to 1 (a property of probability).

By converting the diagram into a matrix, we get the matrix  $M_1$

$$\begin{array}{c|cccc}
 \text{From} \backslash \text{to} & a & b & c & d \\
 \hline
 a & 0.6 & 0 & 0.3 & 0.1 \\
 b & 0 & 0.8 & 0 & 0.2 \\
 c & 0.3 & 0 & 0.4 & 0.3 \\
 d & 0.1 & 0.2 & 0.3 & 0.4
 \end{array} = M_1$$

Note, all columns add to 1, and all rows add to 1 (symmetric)

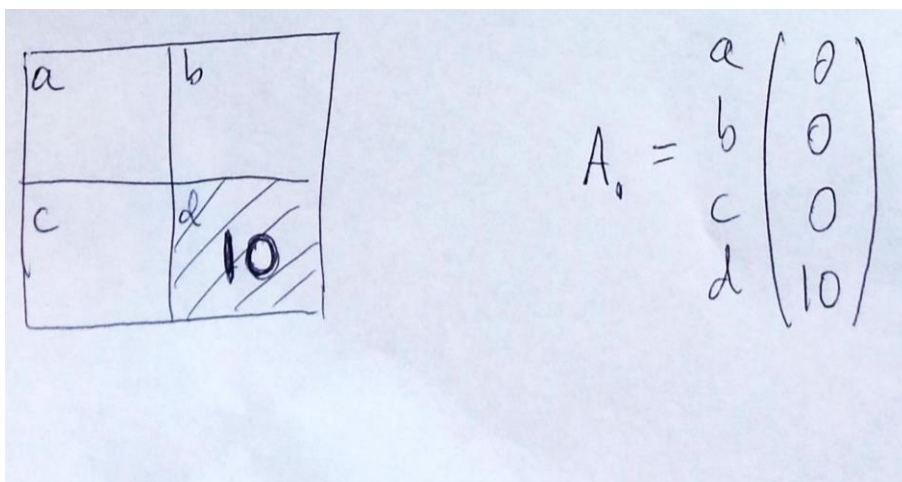
RULE: ROWS add to 1

**\*\* I assume that both ways of transport are the same in probability. Will discuss later.**

Let us say that an outbreak happens in D, where 10 people are infected.

Let us ONLY plot how the 10 people will move around in days.

We firstly represent the outbreak as matrix  $A_1$



To find out what happens after  $n$  days, we apply:

$$I_n = A_1 M_1^n$$

Where  $I_n$  is the infected people in each box.

Using Julia, we can do:

```

1 M = [0.6 0 0.3 0.1;
2       0 0.8 0 0.2;
3       0.3 0 0.4 0.3;
4       0.1 0.2 0.3 0.4]
5
6 A = [0 0 0 10]'
7
8 I = M*A

```

WARNING: imported binding for I overwritten in module Main

> I

✓ 4x1 Array{Float64,2}:

```

1.0
2.0
3.0
4.0

```

> |

The result after 1 day is  $\begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix}$

After 2 days,

$$I_2 = A_1 M_1^2 = \begin{pmatrix} 1.9 \\ 2.4 \\ 3.7 \\ 3 \end{pmatrix}$$

After 3 days and then 100 days:

$$I_3 = A_1 M_1^3 = \begin{pmatrix} 2.25 \\ 2.52 \\ 2.55 \\ 2.68 \end{pmatrix} \quad I_{100} = A_1 M_1^{100} = \begin{pmatrix} 2.5 \\ 2.5 \\ 2.5 \\ 2.5 \end{pmatrix}$$

In other words,

$$\lim_{n \rightarrow \infty} A_1 M_1^n = I_\infty = \begin{pmatrix} \frac{\sum A_1}{no.elements} \\ \frac{\sum A_1}{no.elements} \\ \dots \end{pmatrix} \text{ (for symmetric Transition$$

matrix)

Or, as time goes to infinity, the infected people spread out evenly. **\*\*\* NOTE: ONLY TRUE if transition matrix or  $M_1$  has symmetry (will discuss later about non-symmetry)**

Now, how about mass gatherings? Wouldn't that affect the transition matrix? And how about non-equal weightings (eg:

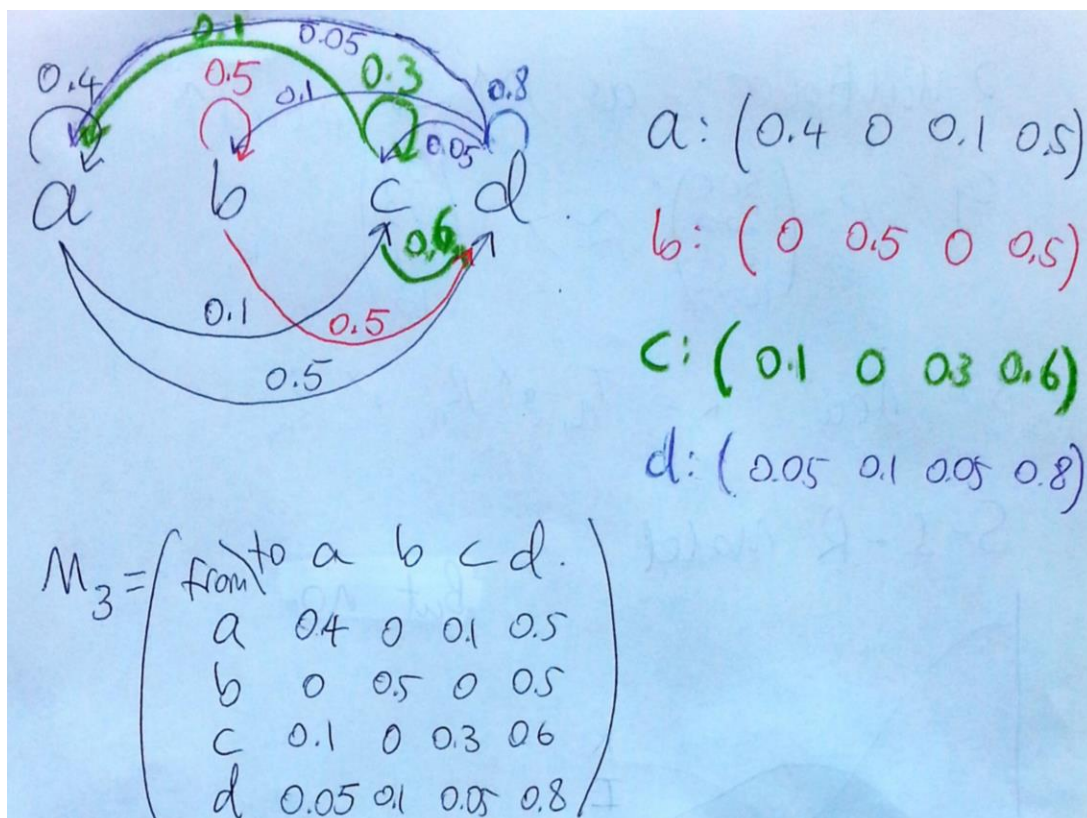
urbanisation causes people to move more into cities than they can move out.)

Let's say that an event occurred in D at time 3. Then the old matrix is:

$$M_1 = \begin{pmatrix} 0.6 & 0 & 0.3 & 0.1 \\ 0 & 0.8 & 0 & 0.2 \\ 0.3 & 0 & 0.4 & 0.3 \\ 0.1 & 0.2 & 0.3 & 0.4 \end{pmatrix}$$

We will also assume D is a city, so urbanisation causes unequal movement of people:

The new one is:



We can see how:

1. ROWS add to 1



## 2. NOT symmetric

So, using

$$I_n = A_1 M_1^n$$

We can see that

$$I_3 = A_1 M_1^2 M_3$$

```

1 M = [0.6 0 0.3 0.1;
2       0 0.8 0 0.2;
3       0.3 0 0.4 0.3;
4       0.1 0.2 0.3 0.4]
5
6 Mc = [0.4 0 0.1 0.5;
7        0 0.5 0 0.5;
8        0.1 0 0.3 0.6;
9        0.05 0.1 0.05 0.8]
10
11 A = [0 0 0 10]
12
13 I = A*M^2*Mc

```

Output: 1x4 Array{Float64,2}:  
1.18 1.5 1.15 6.17

In other words, we get:

$$I_3 = A_1 M_1^2 M_3 = \begin{pmatrix} 1.18 \\ 1.5 \\ 1.15 \\ 6.17 \end{pmatrix}$$

Which makes sense, as D has an event, and is the urban city.

Let's say at times 3, 6, 10 and 15 (days), there is an event.

Let's see what happens.

$$I_3 = A_1 M_1^2 M_3 M_1^2 M_3 M_1^3 M_3 M_1^4 M_3 = \begin{pmatrix} 1.32 \\ 1.56 \\ 1.10 \\ 6.00 \end{pmatrix}$$

This means, that for each day, we can adjust the probabilities as we deem fit.

We have solved:

1. People moving
2. Road, port, plane, port movement
3. Mass gatherings movement

With:

$$I_n = A_1 M_1 M_2 M_3 \times \dots \times M_{n-1} M_n$$

---

## Growth and Decay

---

Now, our next big challenge is to incorporate growth and decay into our model.

So, after gathering data on:

1. Pregnancy chance (or children born per women)
2. Death ratio (deaths per person)

We can formulate a growth-decay model, where:

$$P_0 + \frac{dP}{dt} = P_t$$

$$\frac{dP}{dt} = \text{births} - \text{deaths}$$

Realise we DON'T need IMMIGRATION, as our model already takes care of that!

Now, an issue arises.

How do we account for the extra population? When people move, do they keep moving, or do some die or stop moving?

These problems aren't very easy to solve.

I thought about using a SIR model (Susceptible, Infected, Recovered), but it is horrible for Geographic Spread, as this model is based on entire population centres (meaning no movement of people). This is horrible for our case.

So, we need to produce another epidemiology model focused around on other concepts.

I split this new concept into parts (albeit with assumptions)

1. Take the original population and let it keep running, but with deaths on the way.
2. Each day, take the births and run a separate model on them, with deaths on the way.

Eg:

$$P_0 = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix}$$

$$B(t) = \text{births} = \begin{pmatrix} 0.1 \\ 0.1 \\ 0.1 \\ 0.2 \end{pmatrix} t \quad D(t) = \text{deaths} = \begin{pmatrix} 0.04 \\ 0.04 \\ 0.04 \\ 0.06 \end{pmatrix} t$$

The new model for JUST original population

$$P_0(t) = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix} - D(t) = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix} - \begin{pmatrix} 0.04 \\ 0.04 \\ 0.04 \\ 0.06 \end{pmatrix} t$$

So, we can use this inside our Markov Model.

Now, for the newborns:

1. Do babies move as often as adults? → maybe NO (assumption)

With this in mind, we can scale off the births with a certain scalar.

And then, we can combine it with the original population (we take babies to move similarly with adults, just there are less of them to consider)

So,

$$B(t) = \beta \begin{pmatrix} 0.1 \\ 0.1 \\ 0.1 \\ 0.2 \end{pmatrix} t$$

We can take  $\beta = 0.1$  or whatever (meaning only 10% of babies move around vigorously)

So, our end model:

$$P(t) = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix} - D(t) + B(t) = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix} - \begin{pmatrix} 0.04 \\ 0.04 \\ 0.04 \\ 0.06 \end{pmatrix} t + \beta \begin{pmatrix} 0.1 \\ 0.1 \\ 0.1 \\ 0.2 \end{pmatrix} t$$

Or in equation terms:

$$P(t) = P_0 + t(B\beta - D)$$

So, using:

$$I_n = A_1 M_1 M_2 M_3 \times \dots \times M_{n-1} M_n$$

But, another issue arises. The births and deaths happen on another time set, meaning the Markov process cannot work properly (since it uses one input ONLY).

So, we adjust by separating the births and deaths.

We treat as if both deaths and births “move” around like what normal people do.

So, we first get the probability matrix  $T_n$

$$T_n = M_1 M_2 M_3 \times \dots \times M_{n-1} M_n$$

(Note the removal of A1)

At time 1:

$$I_1 = M_1 * P_0 + M_1 * B\beta - M_1 * D$$

$$I_1 = M_1(P_0 + B\beta - D)$$

At time 2:

$$I_2 = I_1 * M_2 + M_2(B\beta - D)$$

$$I_2 = (I_1 + (B\beta - D))M_2$$

At time 3:

$$I_3 = (I_2 + (B\beta - D))M_3$$

Meaning, we have derived a new infection model with growth.

$I_n = (I_{n-1} + (B\beta - D))M_n$  (where  $\beta$  is a scalar for newborns)

```

1 M = [0.6 0 0.3 0.1;
2 0 0.8 0 0.2;
3 0.3 0 0.4 0.3;
4 0.1 0.2 0.3 0.4]
5
6 Mc = [0.4 0 0.1 0.5;
7 0 0.5 0 0.5;
8 0.1 0 0.3 0.6;
9 0.05 0.1 0.05 0.8]
10
11 A = [1 2 3 4]
12
13 beta = 0.7
14 D = [0.04 0.04 0.04 0.06]
15 B = [0.1 0.1 0.1 0.2]
16
17 n = [[Mc] [M] [M] [Mc] [M] [Mc] [M] [Mc] [M]]
18
19 J = A
20 for x in n
21     J = (J+(B*beta-D))*x
22 end

```

Console output:

```

> J
✓ 1x4 Array{Float64,2}:
 1.85817  2.88072  3.04633  3.74478

```

Once again, we see how the population in fact is correct:

We started off with 10 people. Now, there are 11.52 people after 9 days.

# NEW: ROUTE DERIVATION

So how do we derive the probabilities for each route? First, we need assumptions:

1. People want to stay where they are.
2. People want to move to higher population areas more.
3. Road > Port > Air travel (Cheaper and more accessible)
4. Lower living standards tend to make people more moveable in order to search for better opportunities.
5. BUT - we also know that more liveable areas tend to travel to places more often.
6. Thus: evens out.

So, assume an area A is given by the coordinates: [1,1]

Then, from that area, it has connections for road, no ports and 1 airline.

$$\text{So: } R = \begin{bmatrix} 1 \\ 1 \\ 0 \\ \dots \\ 0 \end{bmatrix} \quad P = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \dots \\ 0 \end{bmatrix} \quad A = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \dots \\ 1 \end{bmatrix}$$

We also know the population:  $Pop = \begin{bmatrix} 6000000 \\ 1000000 \\ 1000000 \\ \dots \\ 5000000 \end{bmatrix}$

Now, the assumptions about movement shows 0.985 staying

potential:  $Stay = \begin{bmatrix} 0.985 \\ 0.985 \\ 0.985 \\ \dots \\ 0.985 \end{bmatrix}$

Now, how do we assign the left over probability to each section? We will assume that people like to move to more populated areas.

Then, we can easily use element wise multiplication:

$$R = \begin{bmatrix} 1 \\ 1 \\ 0 \\ \dots \\ 0 \end{bmatrix} \quad [Pop] = \begin{bmatrix} 6000000 \\ 1000000 \\ 1000000 \\ \dots \\ 5000000 \end{bmatrix}$$

$$\text{Then, } R \% Pop = \begin{bmatrix} 1 \\ 1 \\ 0 \\ \dots \\ 0 \end{bmatrix} \cdot \times \begin{bmatrix} 6000000 \\ 1000000 \\ 1000000 \\ \dots \\ 5000000 \end{bmatrix} = \begin{bmatrix} 6000000 \\ 1000000 \\ 0 \\ \dots \\ 0 \end{bmatrix}$$

$$P \% Pop = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \dots \\ 0 \end{bmatrix} \cdot \times \begin{bmatrix} 6000000 \\ 1000000 \\ 1000000 \\ \dots \\ 5000000 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \dots \\ 0 \end{bmatrix}$$

$$A \% Pop = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \dots \\ 1 \end{bmatrix} \cdot \times \begin{bmatrix} 6000000 \\ 1000000 \\ 1000000 \\ \dots \\ 5000000 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \dots \\ 5000000 \end{bmatrix}$$



Then, we can assume these ratios:

Roads = 1.6. Ports = 0.4. Air = 0.1.

Thus, we times the leftover probability to the ratios

$$R_{left} = 0.125 \times \frac{R_{ratio}}{\sum ratio}$$

$$P_{left} = 0.125 \times \frac{P_{ratio}}{\sum ratio}$$

$$A_{left} = 0.125 \times \frac{A_{ratio}}{\sum ratio}$$

Where:

$$\sum ratio = R_{ratio} + P_{ratio} + A_{ratio}$$

Then, we use these new leftover ratios and times it to the matrix with scaling.

$$[n, m]_R = (R\%Pop \div \sum R\%Pop) \times R_{left}$$

Or, in total:

$$[n, m]_Z = (Z\%Pop \div \sum Z\%Pop) \times Z_{left}$$

Or:

$$[n, m]_Z = \frac{0.125 \times Z_{ratio} \times \{Z \times [Pop] \div (\sum Z \times [Pop])\}}{R_{ratio} + P_{ratio} + A_{ratio}}$$

Remember:

$$Z_{ratio} = 1.6, 0.4, 0.1$$

Once we get  $[n, m]_Z$  for  $[n, m]_{Road}$ ,  $[n, m]_{Port}$ ,  $[n, m]_{Air}$

We then add these together to form a total matrix.

$$[M \text{ start}] = [n, m]_{Road} .+ [n, m]_{Port} .+ [n, m]_{Air}$$

Then, we find the total of this matrix, and find the leftover we need to add onto the original place.

So:

$$[M \text{ left}] = \sum_{i=1}^{i=nrow} [M \text{ start}]x_{k,i} \quad (\text{change } k \text{ for rowsum})$$

All we do now, is we times this leftover to an identity matrix.

So:

$$eye = I_{\dim(M \text{ start})}$$

And,

$$addon = eye .\times [M \text{ left}]$$

Finally, we get the needed Markov Matrix:

$$M_1 = addon .+[M \text{ start}]$$

Or:

$$M_1 = [M \text{ start}] .+ \left[ I_{\dim(M \text{ start})} .\times \sum_{i=1}^{i=nrow} [M \text{ start}]x_{k,i}(\text{rowsum}) \right]$$

Where

$$[M \text{ start}] = [n, m]_{Road} .+ [n, m]_{Port} .+ [n, m]_{Air}$$

---

# MSIR MODEL

---

**IMPORTANT: A new way of thinking about the basic SIR model.**

In the SIR Model, simple differential equations are used to map WHOLE population dynamics (eg an entire country or the whole world). BUT - it is NOT probabilistic but rather DIFFERENTIAL (changes over time but NOT respect to probability)

So, we can use a new method to approach the old SIR Model.

We can use a **Markov SIR Model. (NOT GREENWOOD MODEL, but related)**

In a Markov SIR Model or I'll call it MSIR Model, there are 3 nodes in each area (Susceptible, Infected, Recovered).

Remember the model we made about human movement?

$$I_n = (I_{n-1} + (B\beta - D))M_n$$

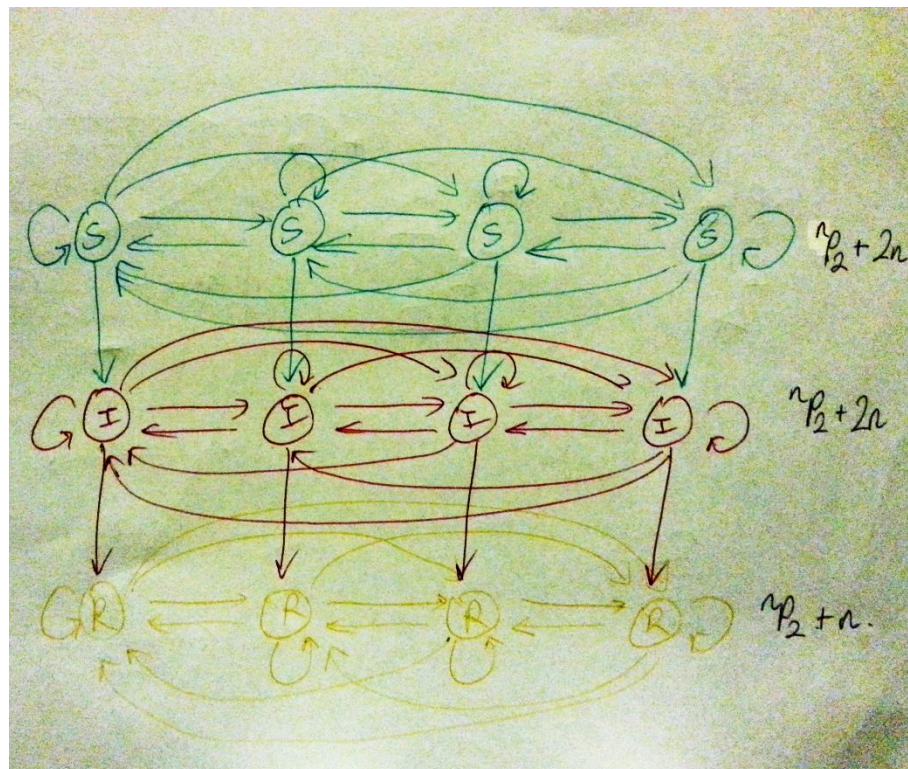
Instead of just monitoring human movement, we can add 2 more dimensions.

1. Let's say a human has a chance of being infected.
  - a. If INFECTED:

- i. Can travel to another location and meet a person
  - 1. Person has chance of being infected.
    - a. IF INFECTED:
      - i. ....
    - b. If NOT INFECTED:
      - i. Can travel to another location and either meet:
        - 1. INFECTED person
          - a. Has chance of being infected.
            - i. IF INFECTED:
              - 1. ....
          - 2. NOT INFECTED person
            - a. Can travel to another location....
- 2. We also know they can have a chance of being recovered (dead, immune)

Let's use A, B, C, D as areas.

Then to draw an entire MSIR diagram.



To work out how many states there are:

1. S: itself, rest =  $1 + 3 + 1$  (to infected)
2. I: itself, rest =  $1 + 3 + 1$  (to recovered)
3. R: itself, rest =  $1 + 3$

So, as there are 4 areas:

1.  $5 * 4 = 20$
2.  $5 * 4 = 20$
3.  $4 * 4 = 4$

Total = 56

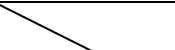
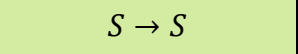
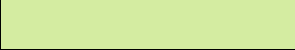
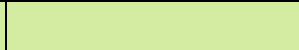
I found that:

Or:

$$States = 3n^2 + 2n$$

Remember in our introduction:

We can express the MSIR model as a table:

		S				I				R			
		a	b	c	d	a	b	c	d	a	b	c	d
From S	a	$S \rightarrow S$				$S \rightarrow I$							
	b												
	c												
	d												
From I	a					$I \rightarrow I$				$I \rightarrow R$			
	b												
	c												
	d												
From R	a									$R \rightarrow R$			
	b												
	c												
	d												

Now, a very big problem arises.

How are we going to derive these separate matrices and combine them into a massive matrix?

1.  $[S \rightarrow S]$  Matrix
2.  $[S \rightarrow I]$  Matrix
3.  $[I \rightarrow I]$  Matrix
4.  $[I \rightarrow R]$  Matrix
5.  $[R \rightarrow R]$  Matrix

Notice the one rule we need to follow:

- ALL ROWS need to add to 1.

## $[S \rightarrow I]$ Matrix

This matrix is just the infection rates of specific areas. This matrix will be calculated by using the mosquito probability index

$$p(x) = x_s x_t x_r x_l$$

However, we need to determine the real ATTACK or INFECTION rate.

Now, we use this probability matrix of mosquitoes, and map their existence in the vicinity. Obviously, to account for mosquito growth, we don't need to, as the probability ratio is enough.

Using  $x_s$  or guessing, let's say there was 1000 mosquitoes in A and 2000 in B.

Then:

$$AR_n = (1000 \quad 2000) \times p(x)$$

We can account for growth if you want:

$$MD_n = (MD_{n-1} + (Mosq_B \omega - Mosq_D)) p(x)$$

MD is for Mosquito Distribution, and  $\omega$  is infant movement scalar.

Once we get a mosquito count for A and B, the higher one has more chance of infection.

So, once we ascertain the amount of mosquitoes, we can produce a weighting ratio to determine how many mosquitoes are seen are each citizen in an area.

We can do this by:

$$M/S_n = \frac{MD_n}{S_n}$$

Where M/S is Mosquito to Mosquito to Susceptible ratio, MD is mosquito distribution and S is susceptible.

Clearly, if the ratio is higher, that means there are more mosquitoes to one person, and thus the higher the chance of spreading.

But, an M/S ratio is not enough to determine the “real” infection potential.

We need to apply automatic algorithmic correcting infection rate in order to accurately guess the infection rate.



I propose we use a susceptible-infection ratio analysis.

BUT first, we have encountered a hurdle. Pretend an area has 100 mosquitoes whilst another 0.3. How do we scale this?

We need to utilise a scaling approach. We can do this by using another normal distribution function.

$$f(M/S) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(M/S-\mu)^2}{2\times\sigma^2}}$$

To test this out, let's say the values are: 0.4, 0.5, 3, 4, 1.2, 10, 12, 0.9, 6.

Then the:

$$\mu = \text{mean} = \frac{1}{n} \sum_{i=1}^n x_i$$

$$\sigma = SD = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2}$$

So, the mean = 4.2222. The SD = 4.04444.

Meaning:

$$f(M/S) = \frac{1}{4.04444\sqrt{2\pi}} e^{-\frac{(M/S-4.2222)^2}{2\times 4.04444^2}}$$

But, we can't use this. Instead we need to utilise the CDF (Cumulative Distribution Function)

$$F(M/S) = \frac{1}{2} \left[ 1 + \operatorname{erf} \left( \frac{M/S - \mu}{\sigma\sqrt{2}} \right) \right]$$

$$F(M/S) = \frac{1}{2} \left[ 1 + \frac{1}{2} \int_{-\frac{M/S-\mu}{\sigma\sqrt{2}}}^{\frac{M/S-\mu}{\sigma\sqrt{2}}} e^{-t^2} dt \right]$$

Now, therein lies the problem of actually making that work (as we have to use numerical approximations).

First, convert our results into a scalable factor

So:

$$z = \frac{M/S - \mu}{\sigma}$$

Using <https://www.hindawi.com/journals/mpe/2012/124029/>

$$CDF \approx \frac{1}{1 + e^{\left[ \left( -\frac{358z}{23} \right) + 111 \tan^{-1} \left( \frac{27z}{294} \right) \right]}}$$

$$AR_n \approx \frac{1}{1 + e^{\left[ \left( -\frac{358(M/S-\mu)}{23\sigma} \right) + 111 \tan^{-1} \left( \frac{27(M/S-\mu)}{294\sigma} \right) \right]}}$$

0.4	0.00606	10	0.99957
0.5	0.00693	12	0.99997
3	0.165	0.9	0.0118
4	0.427	6	0.918
1.2	0.0175		

So we have the possible “Attack Rate”

~~Next, we need an infected-to-susceptible ratio.~~

$$r = \frac{I_n}{S_n}$$

~~But, also, we need to account for the possible changes in infection rate. Thus, we need a better ratio. This will incorporate the potential change in the infected pool. (So if the percentage increase is higher, then we will scaling accordingly— NOTE— this extra input is OPTIONAL. We can remove it on request.)~~

So:

$$r_n = \frac{I_n}{S_n} + \frac{dI_n}{dS_n} IR_{n-1} = \frac{I_n}{S_n} + \frac{I_n - I_{n-1}}{S_n - S_{n-1}} IR_{n-1}$$

~~Now, to find the real infected rate, all we do is apply:~~

$$IR_n = AR_n r_n$$

$$IR_n = AR_n \left( \frac{I_n}{S_n} + \frac{I_n - I_{n-1}}{S_n - S_{n-1}} IR_{n-1} \right)$$

~~Now, to confirm if this function works, let's say at time 0, in area 1, there was NO infected to spread.~~

~~So:~~

$$IR_n = 0.04 \left( \frac{0}{40100} + \frac{0-0}{40100-40000} 0 \right)$$

$$IR_n = 0$$

Which is CORRECT!

~~But, let's say there was suddenly one new infected who travelled there.~~

$$IR_{n+1} = 0.042 \left( \frac{1}{40202} + \frac{1-0}{40202-40100} 0 \right)$$

$$IR_{n+1} = 0.000000104472$$

Which works again!

~~Now, if we times the infection rate to the susceptible, we will have 0.042 new infected people.~~

~~So next:~~

$$IR_{n+2} = 0.0421 \left( \frac{1.042}{40304} + \frac{1.042-1}{40304-40202} 0.000000104472 \right)$$

$$IR_{n+2} = 0.000000108843$$

Which increased! YES it works!

~~To confirm once more, say 1000 were infected at t0 with LOTS of mosquitoes.~~

$$IR_n = 0.55 \left( \frac{1000}{40100} + \frac{1000-0}{40100-40000} 0 \right)$$

$$IR_{n+2} = 0.0137$$

~~So extra 550 people.~~

$$IR_{n+1} = 0.551 \left( \frac{1550}{40202} + \frac{1550-1000}{40202-40100} 0.0137 \right)$$

$$IR_{n+2} = 0.0623$$

YAY!

## Revised Version

I checked to see if my old method works. It doesn't sadly (since the matrix turns negative)

Instead, I propose a different method that actually works.

We first take the infected to population ratio:

$$r = \frac{I_n}{S_n + I_n + R_n}$$

Then, we formulate some arbitrary ratio that we can use as the proposed mosquito infection rate.

$$MIRC = 0.5$$

We then formulate our infection rate with the formula:

$$IR = r \times MIRC \times AR_n$$

Then,

$$[S \rightarrow I] = Identity .* IR$$

$$[S \rightarrow I] = Identity .* \frac{I_n}{S_n + I_n + R_n} * MIRC * AR_n$$

## [S → S] Matrix

Remember our old matrix

$$M_1 = \begin{pmatrix} 0.6 & 0 & 0.3 & 0.1 \\ 0 & 0.8 & 0 & 0.2 \\ 0.3 & 0 & 0.4 & 0.3 \\ 0.1 & 0.2 & 0.3 & 0.4 \end{pmatrix}$$

This is the matrix we saw on how A gets to B and so on with aeroplane, water, road eg. Movements.

We will use this in our formulation of the  $[S \rightarrow S]$  matrix.

Now, if you look at the table, we realise how a  $[S \rightarrow I]$  matrix exists. Therefore, to follow the row add to 1 rule, we need to scale the  $[S \rightarrow S]$  matrix appropriately.

This is simple.

First, minus the  $[S \rightarrow I]$  matrix from 1, and then scale the percentage accordingly. ELEMENT WISE.

$$[S \rightarrow S] = (Identity - [S \rightarrow I]) \times M_n$$

NOTICE:  $[S \rightarrow S]$  can change if there is a mass gathering event.

## $[I \rightarrow R]$ Matrix

This matrix is derived empirically using another scaling method.

First, we need to research what is Zika's recovery rate. They will probably provide a range: eg: from 0.0001 to 0.02 depending on living conditions.

As like before, we need the CDF.

$$LS_n \approx \frac{1}{1 + e^{\left[ \left( -\frac{358(L-\mu)}{23\sigma} \right) + 111 \tan^{-1} \left( \frac{27(L-\mu)}{294\sigma} \right) \right]}}$$

LS is living standards, and L is the living conditions matrix we got from the mosquito one.

To ascertain the certain recovery rate, all we need is:

$$[I \rightarrow R] = R_{lower} + LS_n(R_{higher} - R_{lower})$$

Where lower and higher are the recovery rate bounds.

## $[I \rightarrow I]$ Matrix

### Revised Version

Now, this is the hardest one to find. We need to satisfy 2 assumptions:

1. ROWS add to 1.
2. We guess that infected people TEND to stay more at home or stay in their region of influence (hospitals eg).

So, we need to ADD a new factor and also scale backwards.

First, we need to compute the leftover fraction from the subtraction of  $[I \rightarrow R]$  matrix.

So:

$$[left] = 1 . - \sum_{i=1}^{i=nrow} [I \rightarrow R]x_{k,i} \quad (\text{change } k \text{ for rowsum})$$

Or we say find the 1- rowsum.

Then, we need to scale the  $[S \rightarrow S]$  for convenience.

$$[scaled] = [left] . \times [S \rightarrow S]$$

After that, we need to find the scaled factor for the rest of the matrix nodes:

$$[other] = ([scaled] . \times (1 - Identity)) \times RC$$

Where  $RC$  is the reduction constant eg: 0.5

We then utilise this other's scaling, and add it upon the self-nodal probability as we said that they tend to stay home more.

$$[corrected] = [other] . + \left( [scaled] . + \sum_{i=1}^{i=nrow} [other]x_{k,i} \right) . \times Identity$$

And finally, we combine our findings into one matrix:

$$[I \rightarrow I] = \left[ \left( [left] . - \sum_{i=1}^{i=nrow} [corrected]x_{k,i} \right) . + [corrected] \right] . \\ \times Identity . + ([corrected] . \times Identity)$$

~~Then, we need to produce an "extra" function.~~

~~I did this by: (we can change later)~~

$$[SF] = \frac{1}{nrow} . \times ([left] . - [S \rightarrow S]) . \times Identity$$

~~Now, the issue now is that if we add the SF (scaling factor), we get more than 1 (which is NOT allowed).~~

~~So, to reduce this problem, we need to find how much is increased:-~~

$$[extra] = [left] + \sum_{i=1}^{i=nrow} ([SF] + [S \rightarrow S])_{k,i} - (rowsum)$$

~~Then, we need just the other  $[S \rightarrow S]$  probabilities that we didn't consider (ie the non-home-based one)~~

~~So:~~

$$[rest] = (1 . - Identity) . \times [S \rightarrow S]$$

~~Then, we need to scale to each other to find how much we need to minus from each.~~

$$[error] = [extra] . \times [rest] . \div \sum_{i=1}^{i=nrow} [rest]x_{k,i} - (rowsum)$$

~~Finally, we need to minus the error and add the extra onto the original  $[S \rightarrow S]$  matrix.  
YAY!~~

$$[I \rightarrow I] = [SF] . + [S \rightarrow S] . - [error]$$

## $[R \rightarrow R]$ Matrix

Finally, we get the last matrix to compute. Now, we get 3 options:

1. These people can't go somewhere else, as their family might be infected.
2. These people want to quickly move out to get away from the infected.
3. No change in behaviour.

Now, each of these scenarios require differing matrices. So, I propose we just average them out.

1. These people can't go somewhere else, as their family might be infected.

To get this matrix, we need to add an extra factor of "SF".

$$[SF_{R++}] = \frac{\delta}{nrow} \cdot \times ([left] \cdot - [S \rightarrow S]) \cdot \times Identity$$

Where Delta is the extra factor >1. We can choose 2, 1.5 etc.

Then, we need to find the extra we added.

$$[extra_{R++}] = 1 \cdot - \sum_{i=1}^{i=nrow} ([SF_{R++}] + [S \rightarrow S])_{k,i} \quad (rowsum)$$

Using the  $[rest]$  matrix, we find that



$$[error_{R++}] = [extra_{R++}] \cdot \times [rest] \cdot \div \sum_{i=1}^{i=nrow} [rest]x_{k,i} \quad (rowsum)$$

And,

$$[R \rightarrow R_{++}] = [SF_{R++}] \cdot + [S \rightarrow S] \cdot - [error_{R++}]$$

## 2. These people want to quickly move out to get away from the infected.

This requires an inverse of the method we just did in 1., but will be scaled by 1 only.

$$[SF_{R-}] = \frac{1}{nrow} \cdot \times ([left] \cdot - [S \rightarrow S]) \cdot \times Identity$$

$$[red_{R-}] = 1 \cdot - \sum_{i=1}^{i=nrow} (-[SF_{R-}] + [S \rightarrow S])_{k,i} \quad (rowsum)$$

$$[error_{R-}] = [red_{R-}] \cdot \times [rest] \cdot \div \sum_{i=1}^{i=nrow} [rest]x_{k,i} \quad (rowsum)$$

So:

$$[R \rightarrow R_-] = [S \rightarrow S] \cdot - [SF_{R-}] \cdot + [error_{R-}]$$

## 3. No change in behaviour.

This one is just the original  $[S \rightarrow S]$  matrix BEFORE scaling.

$$[R \rightarrow R_{norm}] = M_n$$

So, to ascertain the final  $[R \rightarrow R]$  matrix, we get the average.

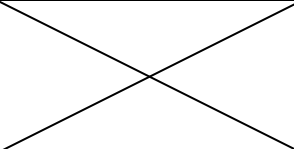
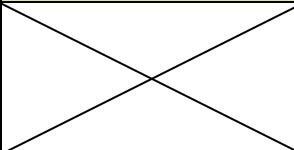
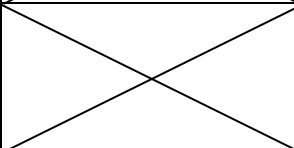
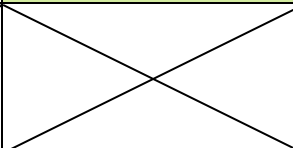
$$[R \rightarrow R] = \frac{[R \rightarrow R_{++}] \cdot + [R \rightarrow R_-] \cdot + [R \rightarrow R_{norm}]}{3}$$

## APPLYING MSIR (9)

Now, after we have derived these matrices, we need to combine them into one ultimate matrix.

Notice how the matrix will constantly change due to the self-correcting nature of  $[I \rightarrow I]$ .

First, we need to follow the structure:

From \ To		S				I				R			
		a	b	c	d	a	b	c	d	a	b	c	d
S	a	$S \rightarrow S$				$S \rightarrow I$							
	b												
	c												
	d												
I	a					$I \rightarrow I$				$I \rightarrow R$			
	b												
	c												
	d												
R	a									$R \rightarrow R$			
	b												
	c												
	d												

This is done by “concatenation” in Julia.

```

z = g.*(1.-eye(nrow,nrow))
P1 = hcat(s_s,s_i,z)
P2 = hcat(z,i_i,i_r)
P3 = hcat(z,z,r_r)

p = vcat(P1,P2,P3)

```

Once we concatenate the matrices into one gigantic one, we can use this self-correcting matrix!

Using:

$$I_n = (I_{n-1} + (B\beta - D))M_n$$

We can find that:

$$SIR_n = (SIR_{n-1} + (SIR_B\beta - SIR_D))p_n$$

Where SIR is the matrix for S,I,R.

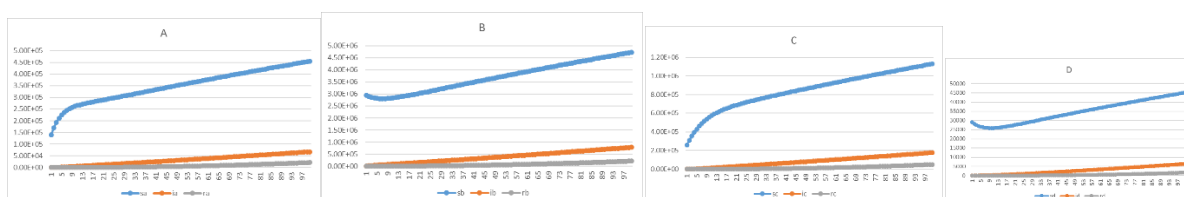
SIR-B is births. SIR-D is deaths. Pn is the gigantic matrix.

And we are done!!!

~~To test this out, I have a simpler version of it (have not yet produced self-correcting version as of yet)~~

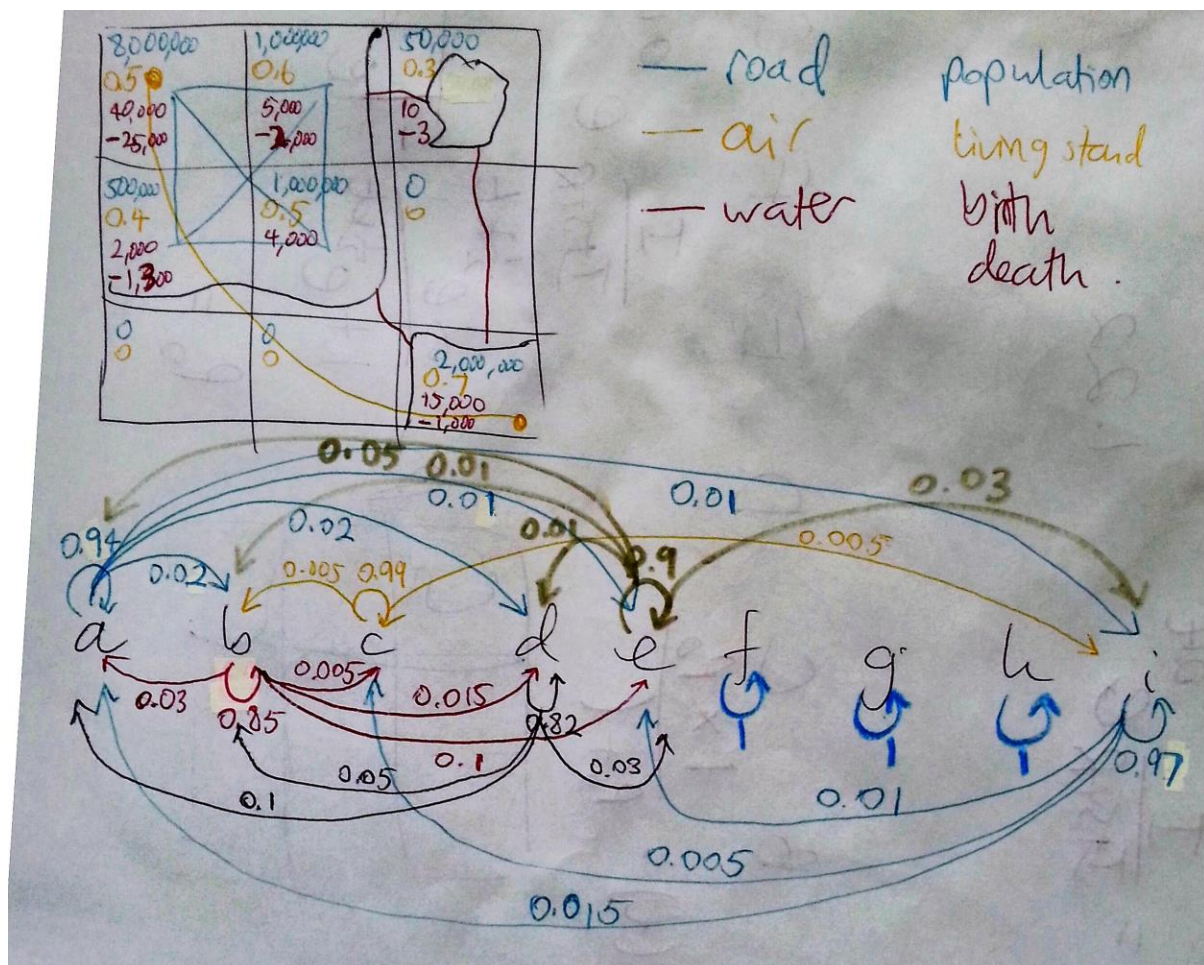
~~I used 4 areas (A,B,C,D) with a low infection rate and a low recovery rate. I made C the city. I made the pandemic happen in D first (a small island state). Graphs over 100 units (days?)~~

~~**NOTICE:** ALL areas have INFECTION RATES (I HAVENT MADE THE SELF CORRECTING MODEL YET) Just testing if the normal model works as intended.~~

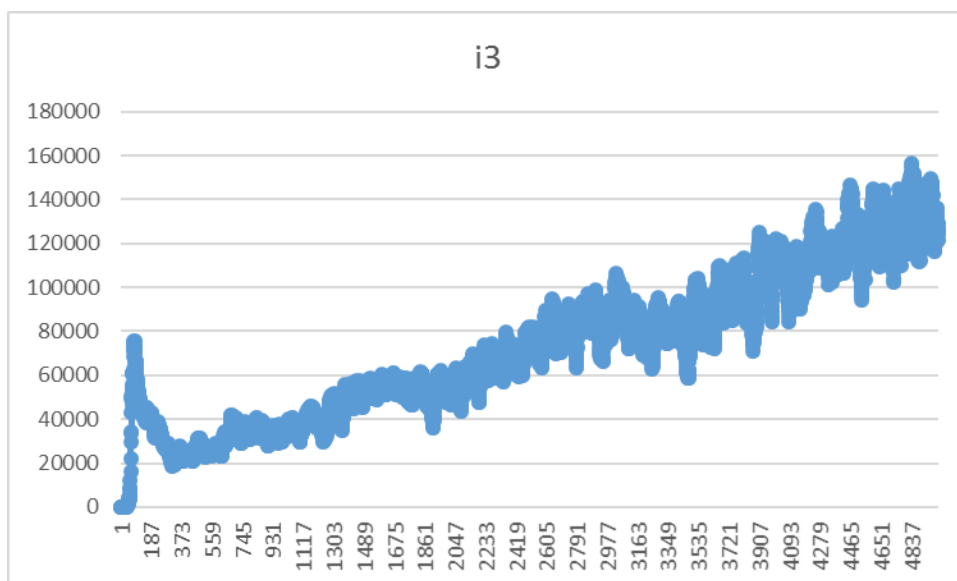
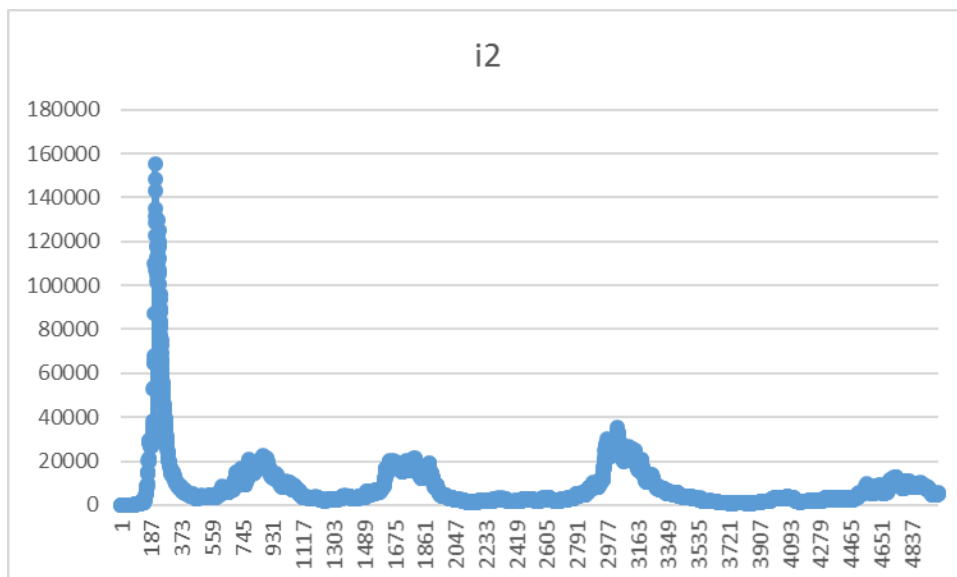
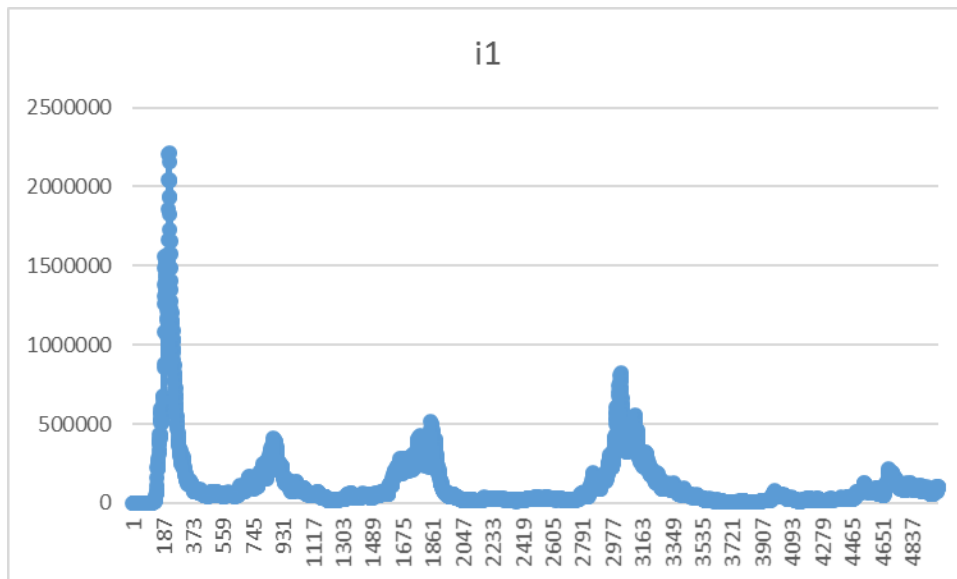


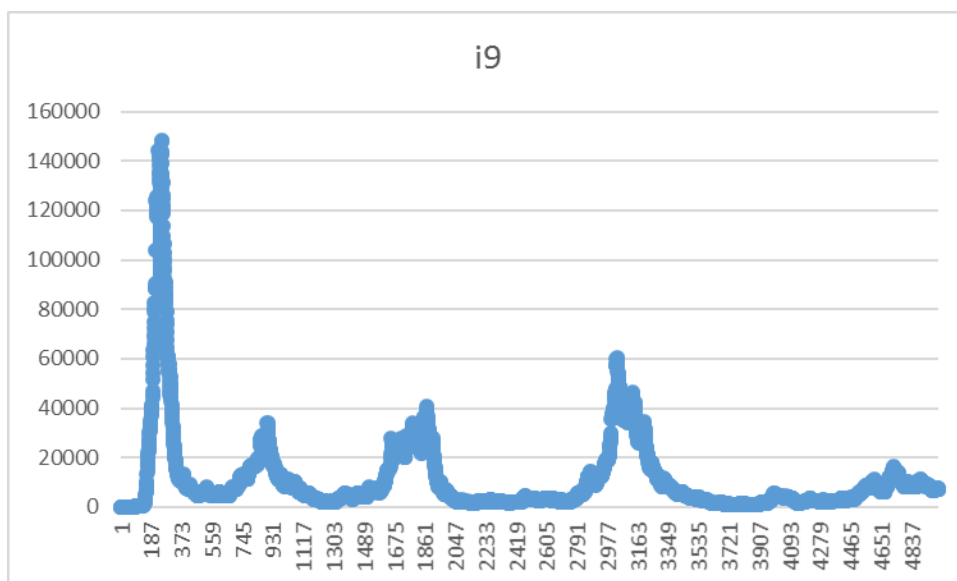
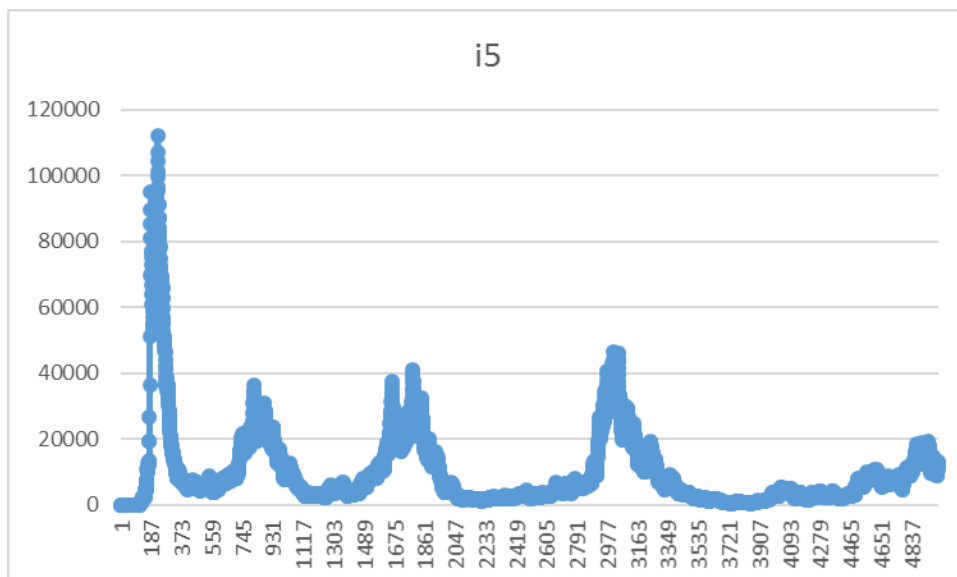
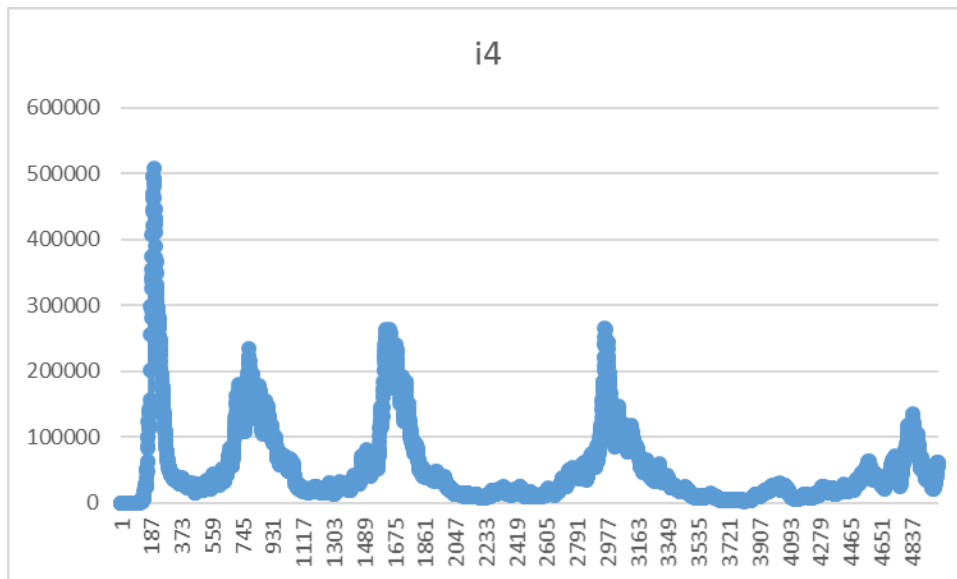
~~It would be better if there was the error correcting one.~~

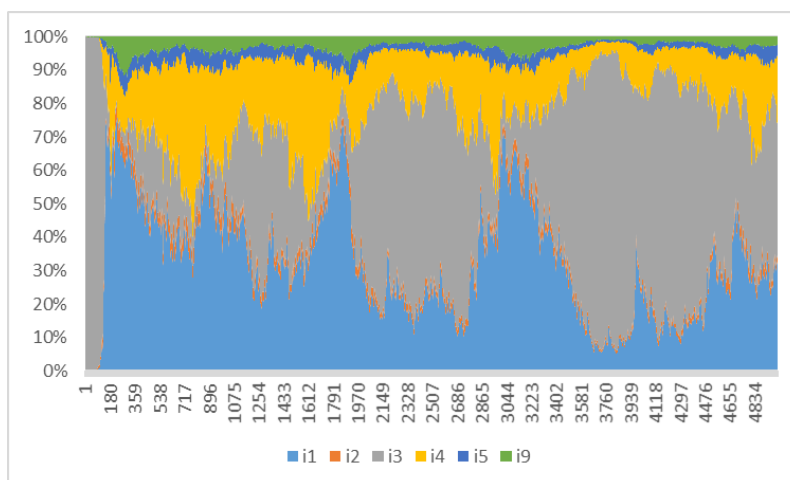
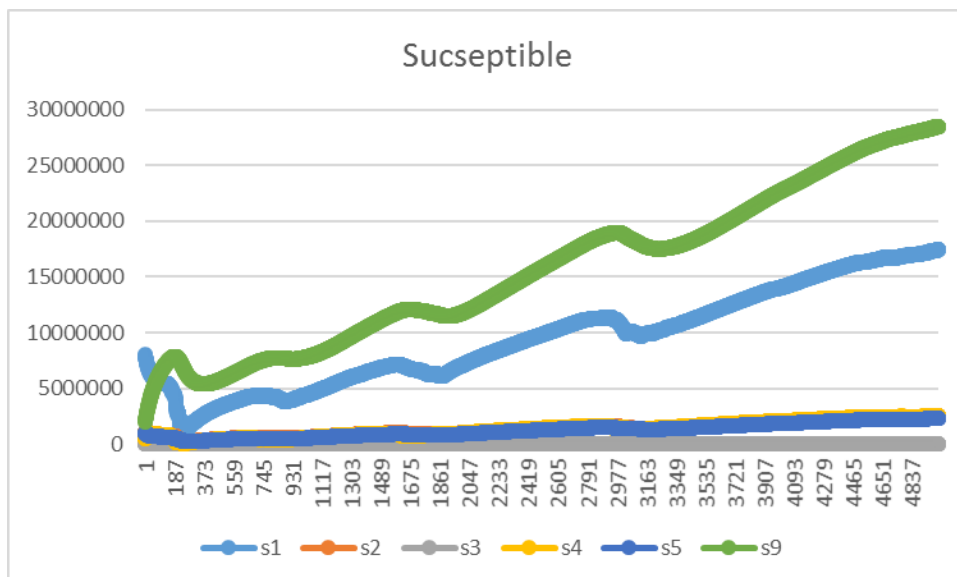
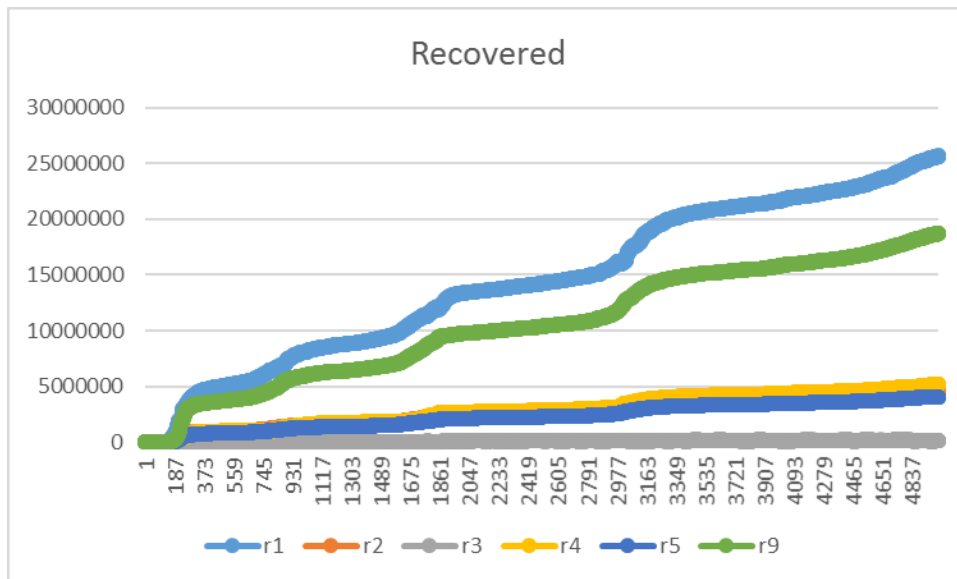
The only issue now is area 3 seems a bit strange. The virus seems to be always there due to their low living standards.



Also, A is a big city but with lower living standards. F is a city with higher living standards.





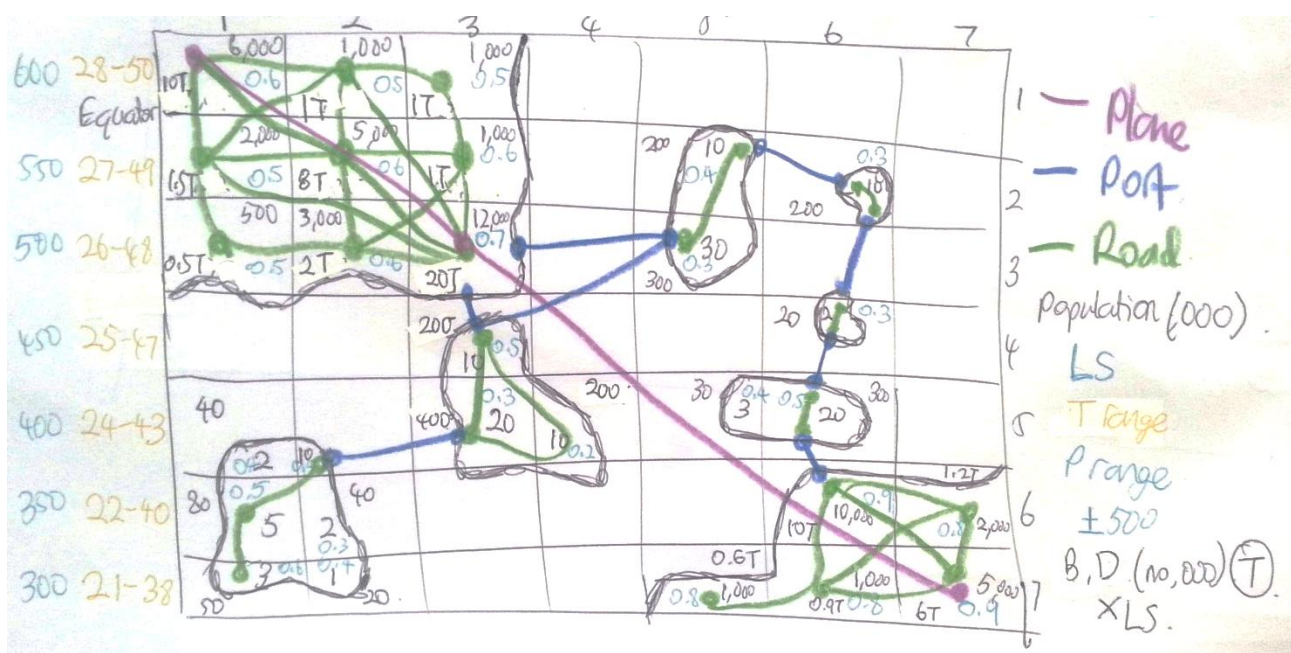


We can see at first, 3 is infected. Then it spread to the rest of the areas.

We can see how 2 gets infected first, then 1. Then, 4 is infected after some delay, but after the delay, it is earlier.



To alleviate some strains of hand doing this, I devised a different encoding system, which is explained below:





Now, we can map our connections by:

$$[1,1] \rightarrow [1,2]$$

$$[1,1] \rightarrow [2,1]$$

And so on,

But, then the problem arises when we want to make this into a matrix saying:

From \ To	1,1
1,1	

And so on. So, to correct this hassle, I devise an easy clumped hand version:

```
road = [11 122122
12 11132122
13 1223
21 1112223331
22 111221233233
23 13223233
31 2132
32 31223323
33 3221321223
35 25
25 35
43 5354
53 4354
52 61
61 5271
75 76
76 666777
66 766777
67 667677
77 666776]
```

```
aeroplane = [11 33
33 1177
77 33]
```

```
port = [33 43
33 35
43 33
35 33
53 52
52 53
25 26
26 25
26 46
46 26
34 56
56 34
56 66
66 56]
```

As you can see, it is wayyyyyy easier than doing it for EVERY 49 combinations.

So, instead of wasting all my time encoding all  $49 \times 49 \times 3 = 7203$  squares, I only needed to do those that I needed, or around only 100 or so. Meaning I saved myself 99% of my time.

Remember, the matrix is full of zeroes, and thus writing them is the pain. If I didn't need to write the zeroes myself, then I can save lots of time.

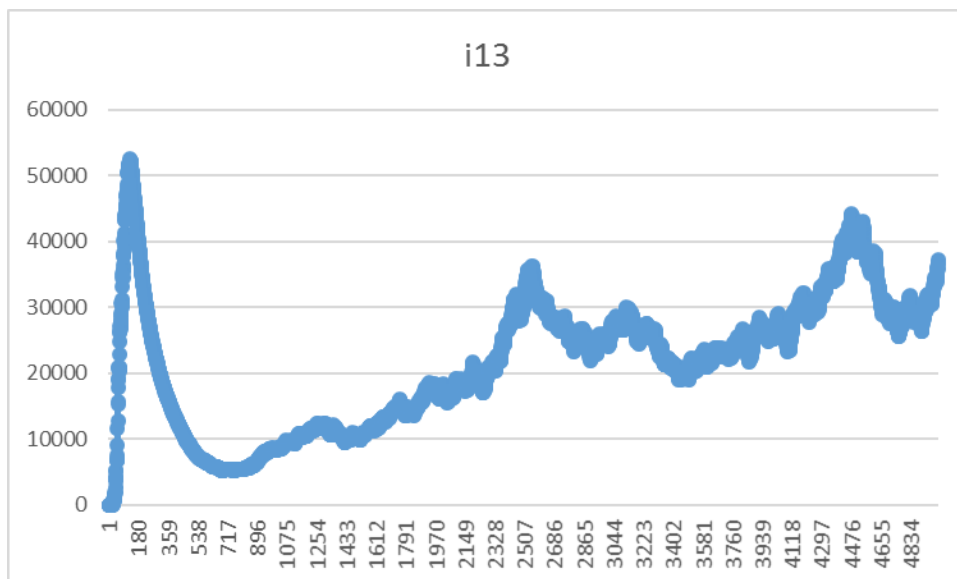
Now, to translate this encoding system into the matrix, we apply some clever programming.

```
122 # Aeroplane
123
124 aeromax = size(aeroplane)[1]
125 aerostart = 1
126 while aerostart <= aeromax
127     place = aeroplane[aerostart,1]
128     pdg = digits(place)
129     pdg2 = pdg[end:-1:1,end:-1:1]
130     down = (pdg2[1]-1)*7+pdg2[2]
131
132     other = aeroplane[aerostart,2]
133     digs = digits(other)[end:-1:1,end:-1:1]
134     len = size(digs)[1]
135     for xyz in range(1,convert(Int64,len/2))
136         down2 = (digs[(xyz*2)-1]-1)*size(pop)[1]+digs[(xyz*2)]
137         aeroplanes[down,down2]=1
138     end
139     aerostart += 1
140 end
141 currentaero = (1-eye(size(pop)[1]^2)).*aeroplanes.*pop_all
142 totalaero = sum(currentaero,2)
143 aerox = 0.1 / (1.6+0.4+0.1)
144 ratioaero = currentaero ./ totalaero
145
146 aerocorrect = copy(ratioaero)
147 for m=1:size(aerocorrect,1)
148     for l=1:size(aerocorrect,2)
149         isnan(aerocorrect[m,l]) ? aerocorrect[m,l]=0 : aerocorrect[m,l]=aerocorrect[m,l]
150     end
```

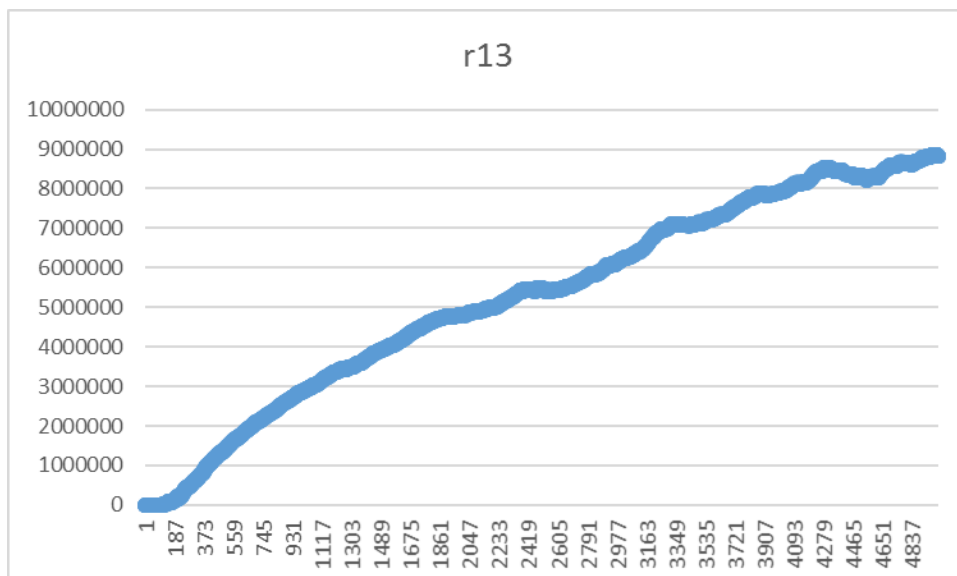
In the end, we get our desired  $49 \times 49$  matrix, saving 99% time.

# NEW: APPLYING MSIR (49)

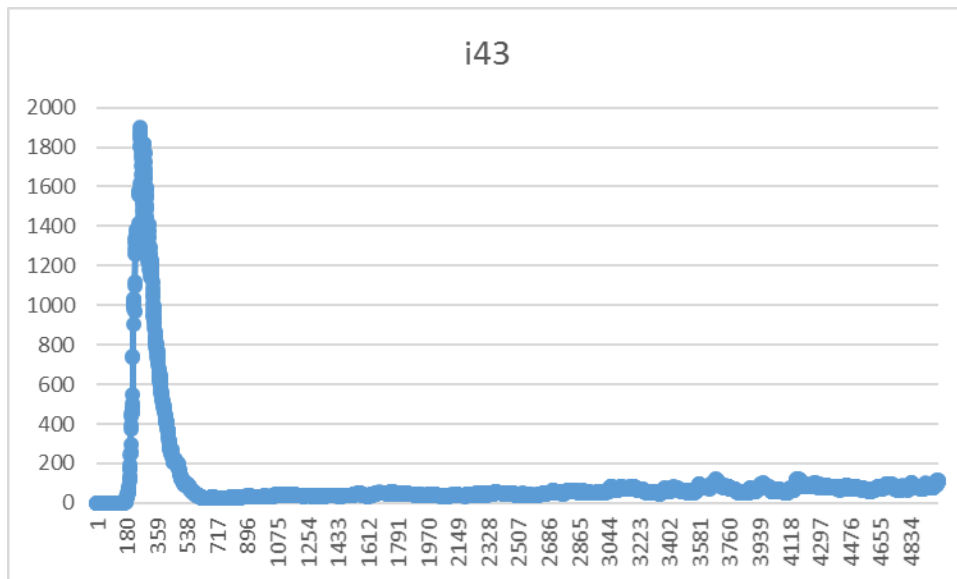
Now, after applying the iteration 5000 times (took like 1 minute), we get the following data: (I started the virus in the small island [2,6] or number 13.



We can see how the virus spreads to a high after 1 period to a 50ish period mark.



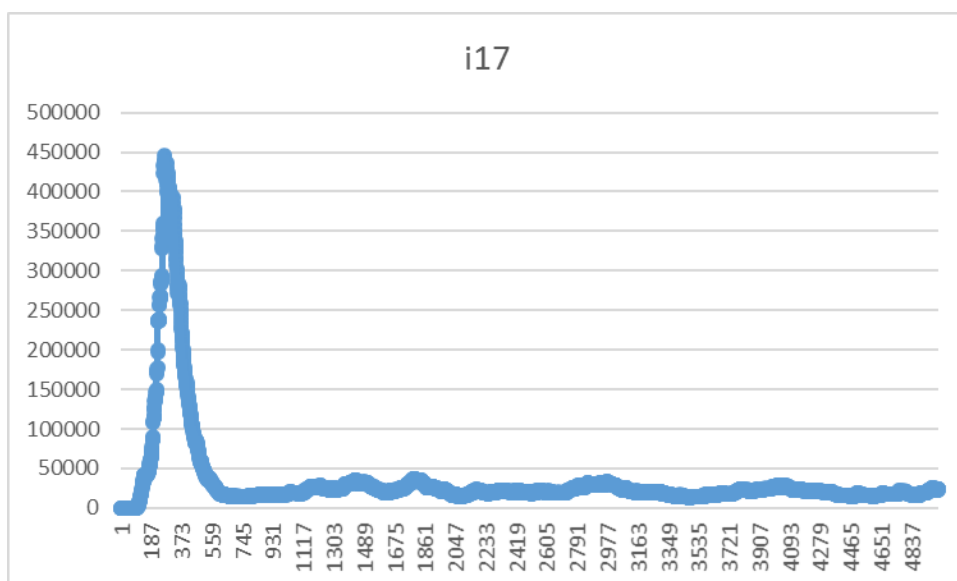
Now, let us see what happens to the furthest to the beginning: area 43 or [7,1]



We can see how after 180 periods (a long lag), it increases. But, the future increase is

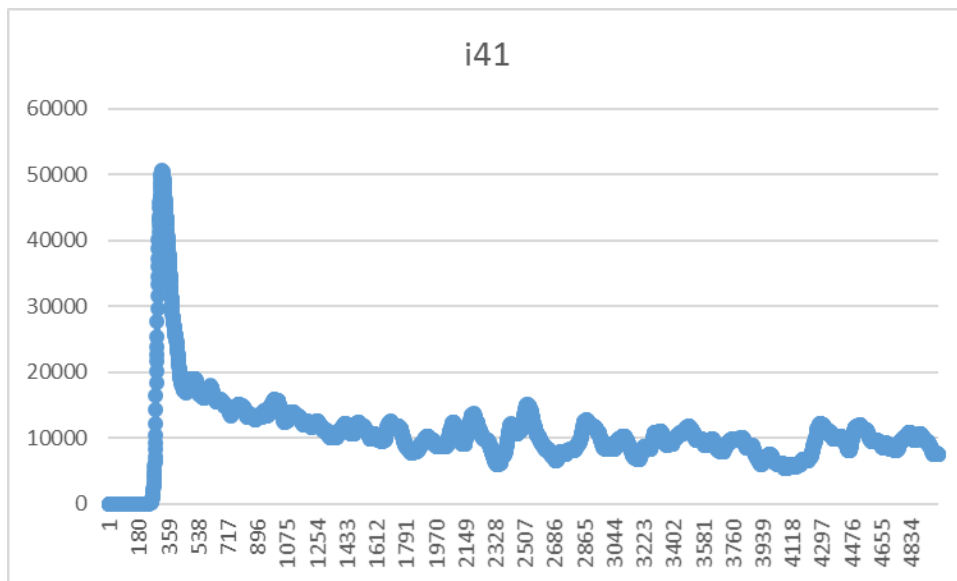
very minimalistic. (meaning it is hard to get there)

Now, the main cities are [3,3](17) (the megacity 12,000,000 pop but lower living standards) and [6,6](41) (the megacity 10,000,000 pop but higher living standards)



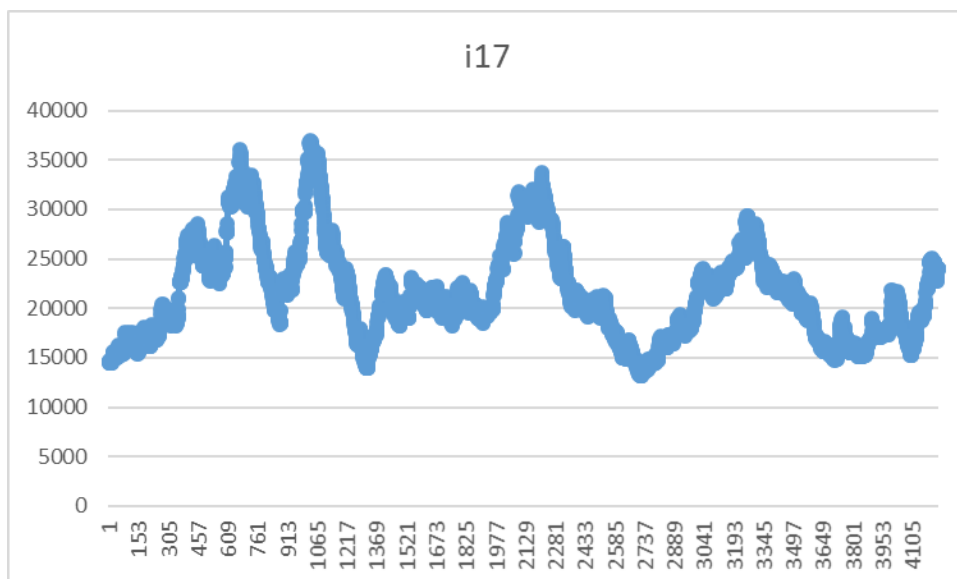
The lower living standard megacity still has leftover infections. The max leftover is 50,000 or

around 0.004 of the population.



Now, 41 may look worse than 17, but you can clearly see a reduction trend. And, the highest

post infection rate is around 15,000, or only 0.0015, which is 62.5% less than the megacity. This is clearly indicative of better living standards.



Now, if I zoom into i17, we see how the trend of infections is NOT reducing. Meaning it is not subsiding (indicative of larger population base with low living standards). Starts at around 900 period.

---

# CONCLUSION

---

Clearly, our way of trying to figure out the dynamic nature of an epidemic has allowed us to solve a massive hurdle in trying to model a real world system.

We have implemented a “world-first” (as I have not seen this on the internet) method of incorporating:

1. Mosquito probability
  - a. Temperature
  - b. Precipitation
  - c. Living Standards
  - d. Environmental Probability
2. Geographical Movement
  - a. Road, port, air
3. SIR Models
4. Markov Systems
5. Growth and Decay
6. Self-correcting infection rates
7. Mass gatherings
8. Statistically sound scaling

Into our MSIR Model!

I will try to make the real model sometime later.

I specialise in mathematical modelling, but I am not very good in getting the real world data. I hope we can get it!

# METHODS AND FORMULAS

$$t = 13.5 \left( \frac{1}{7\sqrt{2\pi}} e^{-\frac{(T-27)^2}{2 \times 7^2}} \right)$$

$$r = 100 \left( \frac{1}{50\sqrt{2\pi}} e^{-\frac{(P-250)^2}{2 \times 50^2}} \right)$$

$$T_{n,m} = (1 - t_{n,n}) \times \frac{t_{n,m}}{-t_{n,n} + \sum_{k=1}^{k=m} t_{n,k}}$$

$$p(x) = x_s x_t x_r x_l$$

$$L_{inv} = 1 - L$$

$$I_n = A_1 M_1 M_2 M_3 \times \dots \times M_{n-1} M_n$$

$$I_n = (I_{n-1} + (B\beta - D))M_n$$

$$States = 3n^2 + 2n$$

$$MD_n = \left( MD_{n-1} + (Mosq_B \omega - Mosq_D) \right) p(x)$$

$$\mu = mean = \frac{1}{n} \sum_{i=1}^n x_i$$

$$\sigma = SD = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2}$$

$$AR_n \approx \frac{1}{1 + e^{\left[ \left( -\frac{358(M/S-\mu)}{23\sigma} \right) + 111 \tan^{-1} \left( \frac{27(M/S-\mu)}{294\sigma} \right) \right]}}$$

$$IR_{\overline{n}} = AR_{\overline{n}} \left( \frac{I_{\overline{n}}}{S_{\overline{n}}} + \frac{I_{\overline{n}} - I_{\overline{n-1}}}{S_{\overline{n}} - S_{\overline{n-1}}} IR_{\overline{n-1}} \right)$$

$$[S \rightarrow I] = Identity \cdot \frac{I_n}{S_n + I_n + R_n} * MIRC * AR_n$$

$$[S \rightarrow S] = (Identity - [S \rightarrow I]) \times M_n$$

$$LS_n \approx \frac{1}{1 + e^{\left[ \left( -\frac{358(L-\mu)}{23\sigma} \right) + 111 \tan^{-1} \left( \frac{27(L-\mu)}{294\sigma} \right) \right]}}$$

$$[I \rightarrow R] = R_{lower} + LS_n(R_{higher} - R_{lower})$$

$$\cancel{[left] = 1 \cdot - \sum_{i=1}^{i=nrow} [I \rightarrow R] x_{k,i} \quad (change\ k\ for\ rowsum)}$$

$$\cancel{[SF] = \frac{1}{nrow} \cdot \times ([left] \cdot - [S \rightarrow S]) \cdot \times Identity}$$

$$\cancel{[extra] = -[left] + \sum_{i=1}^{i=nrow} ([SF] + [S \rightarrow S])_{k,i} \quad (rowsum)}$$

$$\cancel{[rest] = (1 \cdot - Identity) \cdot \times [S \rightarrow S]}$$

$$\cancel{[error] = [extra] \cdot \times [rest] \cdot \div \sum_{i=1}^{i=nrow} [rest] x_{k,i} \quad (rowsum)}$$

$$\cancel{[I \rightarrow I] = [SF] \cdot + [S \rightarrow S] \cdot - [error]}$$

$$[left] = 1 \cdot - \sum_{i=1}^{i=nrow} [I \rightarrow R] x_{k,i} \quad (change\ k\ for\ rowsum)$$

$$[scaled] = [left] \cdot \times [S \rightarrow S]$$

$$[other] = ([scaled] \cdot \times (1 - Identity)) \times RC$$

$$[corrected] = [other] \cdot + \left( [scaled] \cdot + \sum_{i=1}^{i=nrow} [other] x_{k,i} \right) \cdot \times Identity$$



$$[I \rightarrow I] = \left[ \left( [left] . - \sum_{i=1}^{i=nrow} [corrected] x_{k,i} \right) . + [corrected] \right] .$$

$$\times Identity . + ([corrected] . \times Identity)$$

$$[SF_{R++}] = \frac{\delta}{nrow} . \times ([left] . - [S \rightarrow S]) . \times Identity$$

$$[extra_{R++}] = -1 + \sum_{i=1}^{i=nrow} ([SF_{R++}] + [S \rightarrow S])_{k,i} \quad (rowsum)$$

$$[error_{R++}] = [extra_{R++}] . \times [rest] . \div \sum_{i=1}^{i=nrow} [rest] x_{k,i} \quad (rowsum)$$

$$[R \rightarrow R_{++}] = [SF_{R++}] . + [S \rightarrow S] . - [error_{R++}]$$

$$[SF_{R-}] = \frac{1}{nrow} . \times ([left] . - [S \rightarrow S]) . \times Identity$$

$$[red_{R-}] = 1 - \sum_{i=1}^{i=nrow} (-[SF_{R-}] + [S \rightarrow S])_{k,i} \quad (rowsum)$$

$$[error_{R-}] = [red_{R-}] . \times [rest] . \div \sum_{i=1}^{i=nrow} [rest] x_{k,i} \quad (rowsum)$$

$$[R \rightarrow R_-] = [S \rightarrow S] . - [SF_{R-}] . + [error_{R-}]$$

$$[R \rightarrow R_{norm}] = M_n$$

$$[R \rightarrow R] = \frac{[R \rightarrow R_{++}] . + [R \rightarrow R_-] . + [R \rightarrow R_{norm}]}{3}$$

$$SIR_n = (SIR_{n-1} + (SIR_B \beta - SIR_D)) p_n$$

$$[n, m]_Z = \frac{0.125 \ Z_{ratio} \cdot \{Z \cdot [Pop] \div (\sum Z \cdot [Pop])\}}{R_{ratio} + P_{ratio} + A_{ratio}}$$

$$Z_{ratio} = 1.6 \ , \ 0.4 \ , \ 0.1$$

$$[M \ start] = [n, m]_{Road} \cdot + \ [n, m]_{Port} \cdot + \ [n, m]_{Air}$$

$$M_1 = [M \ start] \cdot + \left[ I_{\dim(M \ start)} \cdot \sum_{i=1}^{i=nrow} [M \ start] x_{k,i}(rowsum) \right]$$