Mathematics and Statistics Research Competition 2020

Question 3 – Presentation

Solution:

Given rules

Rules numbers	
1.	$yx = xyc^{k1}$
2.	$zx = xyzc^{k2}$
3.	$zy = x^{-1}zc^{k3}$
4.	$xc = cx$
5.	$yc = cy$
6.	$zc = cz$
7.	$Z^6 = c^{K4}$
8.	$x^{-1}x = xx^{-1} = 1$

From the above given rule 4,

$xc^2 = xc*c$

$= cx*c$

$= c\,(xc)$

$= c\,(cx)$

$= ccx$

$= c^2x$

So, at the same way, we can have

$xc^{ki} = c^{ki}x \quad (i = 1, 2, 3, 4)$

Similarly, from the given rule 5 and rule 6, we can have

$yc^{ki} = c^{ki}y \quad (i = 1, 2, 3, 4)$

$zc^{ki} = c^{ki}z \quad (i = 1, 2, 3, 4)$

Therefore, we can have

Rule 9: $xc^{ki} = c^{ki}x$; $yc^{ki} = c^{ki}y$; $zc^{ki} = c^{ki}z$; (i = 1, 2, 3, 4)

From the given equation $1 = (z^3x)^2$, we can derivate as following:

Derivation	Remark
$1 = (z^3x)^2$	
$=(z^3x)(z^3x)$	
$= z^2 (zx) z^3x$	
$= z^2 (xyzc^{k2}) z^3x$	As the given $zx = xyzc^{k2}$
$= z (zx)yzz^3x\ c^{k2}$	As the rule 9
$= z(xyzc^{k2})\ yz^4xc^{k2}$	$zx = xyzc^{k2}$
$= zxy(zy)\ z^4xc^{k2}c^{k2}$	As the rule 9
$= zxyx^{-1}zc^{k3}z^4xc^{k2}c^{k2}$	$zy = x^{-1}zc^{k3}$
$= zxy\ x^{-1}\ zz^4x\ c^{k2}\ c^{k2}c^{k3}$	As the Rule 9
$= zxyx^{-1}z^5xc^{(2k2+\ k3)}$	$c^{k2}\ c^{k2}c^{k3} = c^{(2k2+\ k3)}$
$= z\ (xy)\ x^{-1}z^5xc^{(2k2+\ k3)}$	
$= z\ (yxc^{-k1})\ x^{-1}z^5xc^{(2k2+\ k3)}$	As $yx = xyc^{k1}$ .Therefore, $xy = yxc^{-k1}$
$= z\ (yx)\ x^{-1}z^5x\ c^{-k1}c^{(2k2+\ k3)}$	As the rule 9
$= zyxx^{-1}z^5xc^{(2k2\ +\ k3\ -\ k1)}$	$c^{-k1}c^{(2k2+\ k3)} = c^{(2k2\ +\ k3\ -\ k1)}$
$= (zy)\ z^5xc^{(2k2+\ k3-\ k1)}$	As the given Rule 8: $xx^{-1} = 1$
$= x^{-1}zc^{k3}z^5xc^{(2k2+\ k3-\ k1)}$	$zy = x^{-1}zc^{k3}$

$= x^{-1}zz^5x\ c^{k_3}c^{(2k_2+\ k_3-\ k_1)}$	As the rule 9
$= x^{-1}zz^5xc^{(2k_2+\ 2k_3-\ k_1)}$	$c^{k_3}c^{(2k_2+\ k_3-\ k_1)} = c^{(2k_2\ +\ 2k_3\ -\ k_1)}$
$= x^{-1}z^6xc^{(2k_2+\ 2k_3-\ k_1)}$	
$= x^{-1}c^{k_4}xc^{(2k_2+2k_3-\ k_1)}$	$z^6 = c^{\ k_4}$
$= x^{-1}x\ c^{k_4}c^{(2k2+2k3-\ k1)}$	As the Rule 9
$= x^{-1}x\ c^{(2k_2\ +\ 2k_3\ +\ k_4-\ k_1)}$	$c^{k_4}c^{(2k_2+2k_3-\ k_1)} = c^{(2k_2\ +\ 2k_3\ +\ k_4-\ k_1)}$
$= c^{(2k_2+\ 2k_3-\ k_1+\ k_4)}$	$x^{-1}x = 1$

As anything to the power of 0 equals 1, $c^{(2k_2\ +\ 2k_3\ +\ k_4\ -\ k_1)} = 1$.

Therefore, **$2k_2 + 2k_3 + k_4 - k_1 = 0$**

School – GWSC

Students - Christabel Yue Yi Liu & Cindy Wu

Melbourne Uni Research Competition

Question 7:

In this problem, we are given two mappings:

$$T_1(x) = 2 - \frac{1}{x} \text{ and } T_2(x) = 1 - \frac{1}{x}$$

and their inverses:

$$T_1{}^{-1}(x) = \frac{1}{2-x} \text{ and } T_2{}^{-1}(x) = \frac{1}{1-x}$$

We must find a formula (if one exists) with these four mappings which reduces any number $\frac{a}{b}$ to 0.

Let us substitute $\frac{a}{b}$ into these mappings and assume the output is $\frac{a'}{b'}$

$T_1\left(\frac{a}{b}\right) = 2 - \frac{b}{a} = \frac{2a}{a} - \frac{b}{a} = \frac{2a-b}{a}$ so $a' = 2a - b$ and $b' = a$

$T_2\left(\frac{a}{b}\right) = 1 - \frac{b}{a} = \frac{a}{a} - \frac{b}{a} = \frac{a-b}{a}$ so $a' = a - b$ and $b' = a$

$T_1{}^{-1}\left(\frac{a}{b}\right) = \frac{1}{2-\frac{a}{b}} = \frac{1}{\frac{2b-a}{b}} = \frac{b}{2b-a}$ so $a' = b$ and $b' = 2b - a$

$T_2{}^{-1}\left(\frac{a}{b}\right) = \frac{1}{1-\frac{a}{b}} = \frac{1}{\frac{b-a}{b}} = \frac{b}{b-a}$ so $a' = b$ and $b' = b - a$

We will also work out $T_2{}^{-1}[T_1\left(\frac{a}{b}\right)]$ as this will help us in the future.

$T_2{}^{-1}\left[T_1\left(\frac{a}{b}\right)\right] = T_2{}^{-1}\left[\frac{2a-b}{a}\right] = \frac{1}{1-\frac{2a-b}{a}} = \frac{1}{\frac{a}{a}-\frac{2a-b}{a}} = \frac{1}{\frac{b-a}{a}} = \frac{a}{b-a}$ so $a' = a$ and $b' = b - a$

We assume that the fraction $\frac{a}{b}$ is in its simplest form, if $\frac{a}{b} < 0$, then $a < 0$ and $b > 0$, and $b \neq 0$ so the fraction is not infinite.

Let us define an iterative method which uses the following rules:

1. If a = 0 then we are done as $\frac{a}{b}$ will equal 0.
2. If $a < 0$ (so $b > 0$ and $\frac{a}{b} < 0$) then apply $T_2{}^{-1}\left(\frac{a}{b}\right) = \frac{b}{b-a}$.
3. If $a \geq b$ (so $\frac{a}{b} \geq 1$) then apply $T_2\left(\frac{a}{b}\right) = \frac{a-b}{a}$
4. If $\frac{1}{2} \leq \frac{a}{b} < 1$ then apply $T_1\left(\frac{a}{b}\right) = \frac{2a-b}{a}$
5. If $0 < \frac{a}{b} < \frac{1}{2}$ then apply $T_2{}^{-1}\left[T_1\left(\frac{a}{b}\right)\right] = \frac{a}{b-a}$

These rules should be applied to the original number, $\frac{a}{b}$ until it has been reduced to 0.

An example has been shown which will reduce $\frac{3}{7}$ to 0 using the iterative method:

Since $0 < \frac{3}{7} < \frac{1}{2}$ we must apply ${T_2}^{-1}\left[T_1\left(\frac{3}{7}\right)\right] = \frac{3}{7-3} = \frac{3}{4}$

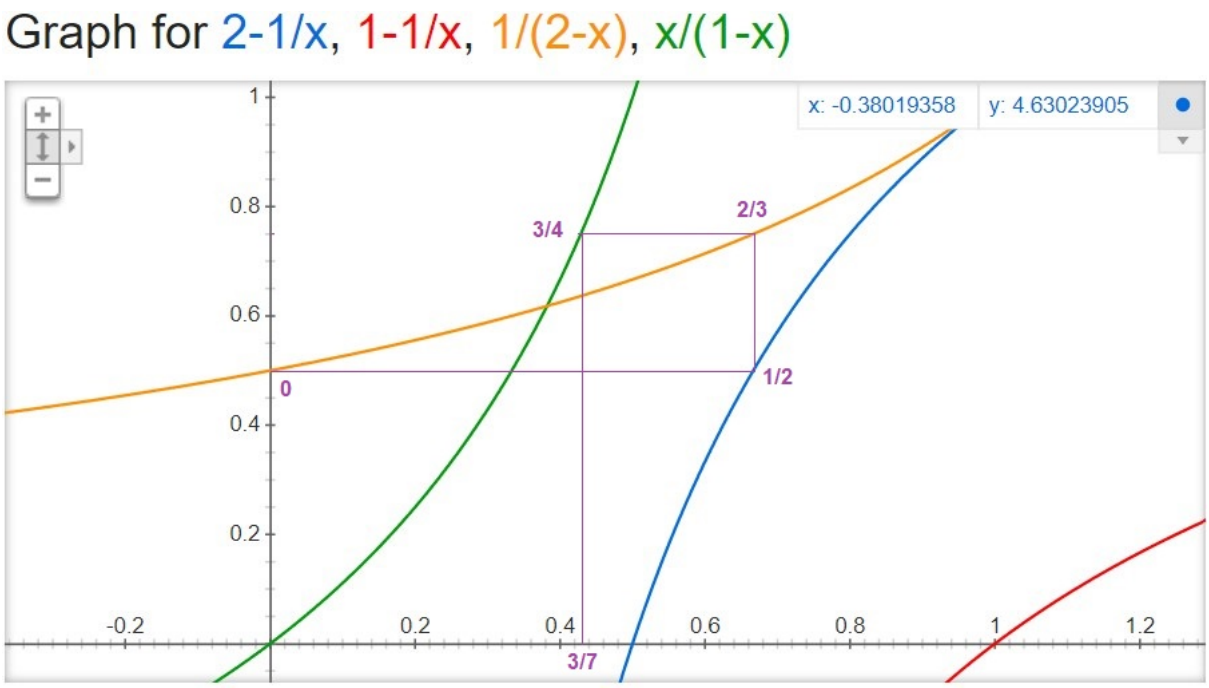
Since $\frac{1}{2} \leq \frac{3}{4} < 1$ we must apply $T_1\left(\frac{3}{4}\right) = \frac{6-4}{3} = \frac{2}{3}$

Since $\frac{1}{2} \leq \frac{2}{3} < 1$ we must apply $T_1\left(\frac{2}{3}\right) = \frac{4-3}{2} = \frac{1}{2}$

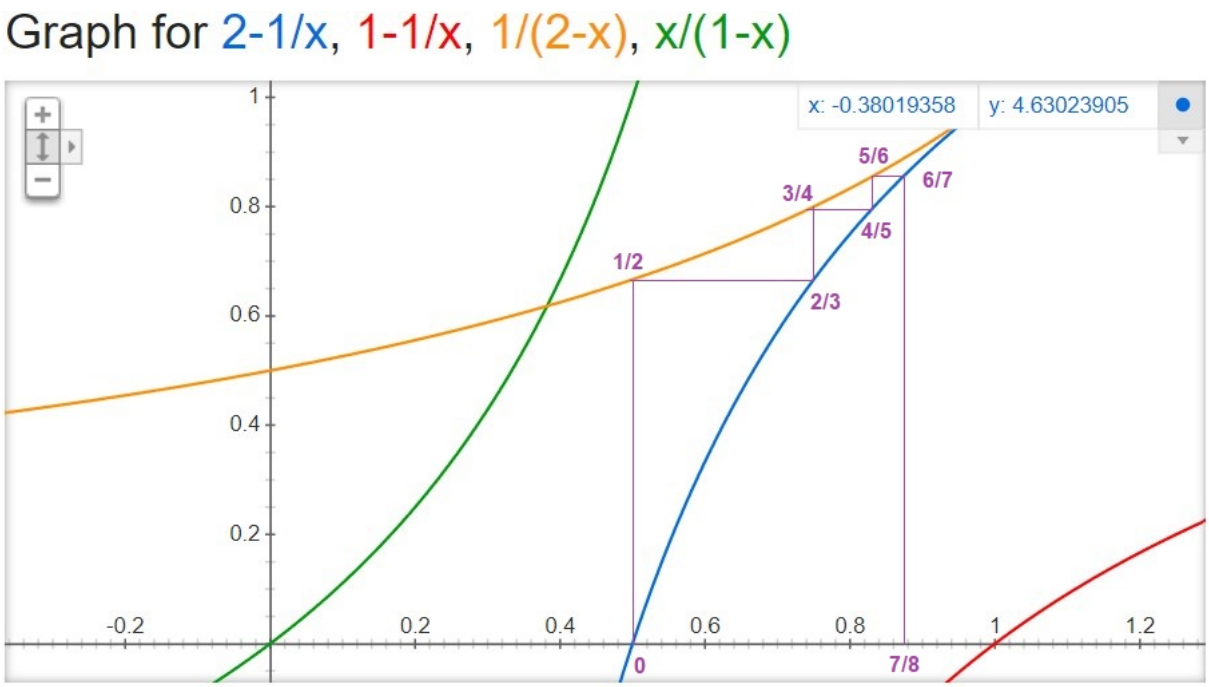
Since $\frac{1}{2} \leq \frac{1}{2} < 1$ we must apply $T_1\left(\frac{1}{2}\right) = \frac{2-2}{1} = \frac{0}{1} = 0$

Since the number is now 0, we are done and $\frac{3}{7}$ has now been reduced to 0.

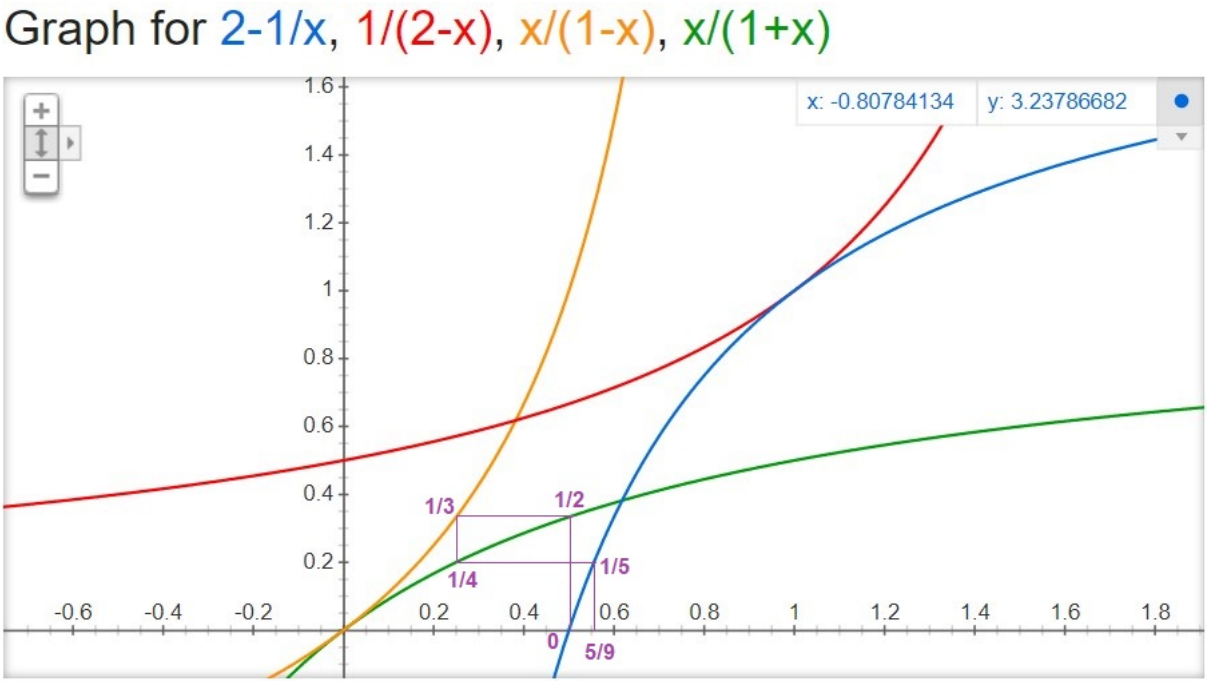
Graphs showing $\frac{3}{7}$, $\frac{7}{8}$ and $\frac{5}{9}$ being reduced to 0 using this method are shown below.



$\frac{3}{7}$ being reduced to 0



$\frac{7}{8}$ being reduced to 0



$\frac{5}{9}$ being reduced to 0

We will now prove why this iterative method will work for all fractions $\frac{a}{b}$.

1. If $\frac{a}{b} = 0$ then we are done as $\frac{a}{b}$ has already been reduced.

2. If $\frac{a}{b} < 0$ then we should apply ${T_2}^{-1}\left(\frac{a}{b}\right) = \frac{b}{b-a}$. Since $a < 0$ and $-a > 0, b-a > b > 0$. So, $0 < \frac{b}{b-a} < 1$. Now, we can apply Rule 4 or 5 to $\frac{b}{b-a}$ depending on whether $\frac{b}{b-a} < \frac{1}{2}$ or not.

3. If $\frac{a}{b} \geq 1$ then we should apply $T_2\left(\frac{a}{b}\right) = \frac{a-b}{a}$. Since $a \geq b, a-b \geq 0$. Also, $a-b \leq a$ so $0 \leq \frac{a-b}{a} < 1$. Either $\frac{a-b}{a} = 0$ (thus we are done) or $0 < \frac{a-b}{a} < 1$. Now, we can apply Rule 4 or 5 to $\frac{a-b}{a}$, depending on whether $\frac{a-b}{a} < \frac{1}{2}$ or not.

4. If $\frac{1}{2} \leq \frac{a}{b} < 1$ then we should apply $T_1\left(\frac{a}{b}\right) = \frac{2a-b}{a} = 2 - \frac{b}{a}$. $\frac{a}{b} \geq \frac{1}{2}$, $2a \geq b$, $2a - b \geq 0$, also $\frac{b}{a} \leq 2$, $\frac{b}{a} - 2 \leq 0$, $2 - \frac{b}{a} \geq 0$. Furthermore, $\frac{a}{b} < 1$, $\frac{b}{a} > 1$, $\frac{b}{a} - 1 > 0$, $\frac{b}{a} - 2 > -1$, $2 - \frac{b}{a} < 1$. So, $0 \leq 2 - \frac{b}{a} < 1$. Now, we can apply Rule 1, 4 or 5 to $2 - \frac{b}{a}$, depending on whether $2 - \frac{b}{a} = 0, or\ < \frac{1}{2}, or > \frac{1}{2}$.

5. If $0 < \frac{a}{b} < \frac{1}{2}$ then we should apply ${T_2}^{-1}\left[T_1\left(\frac{a}{b}\right)\right] = \frac{a}{b-a}$. $\frac{a}{b} < \frac{1}{2}$, $2a < b$, $2a - b < 0$, $b - 2a > 0$, $b - a > 0$, $\frac{a}{b-a} > 0$. Furthermore, $b - 2a > 0$, $b - a > a$, $\frac{b-a}{a} > 1$, $\frac{a}{b-a} < 1$. So, $0 < \frac{a}{b-a} < 1$. Now, we can apply Rule 4 or 5 to $\frac{a}{b-a}$, depending on whether $\frac{a}{b-a} < \frac{1}{2}$ or not.

We will now show that the iterative method will result in the number being reduced to 0 and not create an infinite loop. Steps 1,2, and 3 either result in the reduction and the iteration stops or 4 and 5 being applied. We thus focus on steps 4 and 5.

Let us define $n = b - a$.

We first consider step 4 applied when $\frac{1}{2} \leq \frac{a}{b} < 1$. $n = b - a > 0$ since $b > a$. Also $n < b$ and $n \leq a$..We apply $T_1\left(\frac{a}{b}\right) = \frac{2a-b}{a} = \frac{a-(b-a)}{b-(b-a)} = \frac{a-n}{b-n}$. This means that the numerator and denominator of the output will always be smaller than the numerator and denominator of $\frac{a}{b}$.

If we can reduce the number to $\frac{1}{2}$, we are done as we can apply $T_1\left(\frac{a}{b}\right) = \frac{2a-b}{a}$ to it to make it 0. Therefore, we want the denominator and numerator to be as small as possible.

When $\frac{1}{2} \leq \frac{a}{b} < 1$, let us calculate $\frac{a}{b} - \frac{(a-n)}{(b-n)}$.

$\frac{a}{b} - \frac{(a-n)}{(b-n)} = \frac{[a(b-n)-(a-n)b]}{b(b-n)} = \frac{ab-an-ab+bn}{b(b-n)} = \frac{(b-a)n}{ab} = \frac{(b-a)^2}{ab} > 0$

So, $\frac{a}{b} - \frac{(a-n)}{(b-n)} > 0$, $\frac{(a-n)}{(b-n)} < \frac{a}{b}$, $\frac{a-(b-a)}{b-(b-a)} < \frac{a}{b}$, $\frac{2a-b}{a} < \frac{a}{b}$, $T_1\left(\frac{a}{b}\right) = \frac{2a-b}{a} < \frac{a}{b}$

Therefore, every time the fraction $\frac{a}{b}$ is applied to $T_1\left(\frac{a}{b}\right) = \frac{2a-b}{a}$ (following Rule 4) the output will be smaller than $\frac{a}{b}$. The iteration with rule 4 will continue until either $\frac{a}{b} = \frac{1}{2}$ and the fraction is subsequently reduced or $0 < \frac{a}{b} < \frac{1}{2}$ and rule 5 applies.

We now consider step 5 applied when $0 < \frac{a}{b} < \frac{1}{2}$. Let us compare $\frac{a}{b}$ and $\frac{a}{b-a}$.

$\frac{a}{b-a}$ is the output of the mapping ${T_2}^{-1}\left[T_1\left(\frac{a}{b}\right)\right] = \frac{a}{b-a}$ (when following Rule 5) and we can see that compared to $\frac{a}{b}$, the numerator is constant and the denominator decreases. The iteration with rule 5 will continue until $\frac{1}{2} \leq \frac{a}{b} < 1$and rule 4 applies.

Therefore, when applying Rule 4 and 5 of the iterative method to a fraction the numerator will either remain the same or decrease and denominator will always decrease. This means that eventually, they will converge to 0 and $\frac{a}{b}$ will be reduced to 0.

Thus, we have shown that this iterative method will reduce any number $\frac{a}{b}$ to 0.

DETERMINANT TIC TAC TOE

Shuana Lu, Serena Tam and Selina Zhou

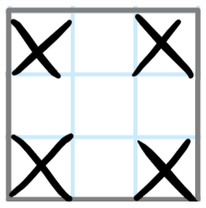
In Determinant Tic Tac Toe, the winner is determined by the determinant of the matrix grid at the end of the game. If the determinant is 0, then Player 0 wins. If not, Player 1 wins. The expression used to find the determinant is $x_1(y_2z_3 - y_3z_2) - y_1(x_2z_3 - x_3z_2) + z_1(x_2y_3 - x_3y_2)$. The following slides provide optimised methods for Player 0 to always win any game of Determinant Tic Tac Toe.

The playing grid:

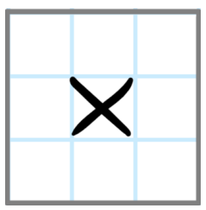
$$\begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{bmatrix}$$

Terms used in the answers:

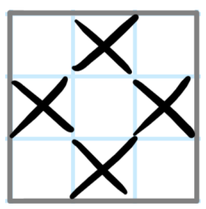
Corner Positions:



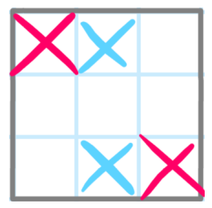
Centre Position:



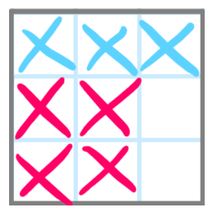
Side Positions:



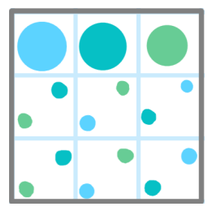
Red: Diagonally opposite
Blue: Directly across



Red: 2x2 square
Blue: 3 in a row



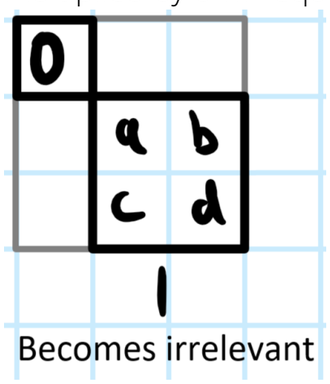
Larger dots: Top row element
(represents x1, y1, and z1)
Smaller dots: 2x2 squares
(represents x2, x3, y2, y3, z2, and z3)



PART A: Is there a method for Player 0 to always win?

Key Ideas

A 2x2 square is irrelevant when its corresponding element in the top row is '0' - the determinant will be multiplied by 0 (anything multiplied by 0 will equal 0).



Making 0 the determinant of a 2x2:

Player 0 should ensure there are two adjacent '0's. This could be a&c, a&b, d&b, or d&c.

a	b
c	d

Example:

0	1
0	1

Having 3 '0's in a row or column:

Bottom two rows: all the 2x2 squares have determinants of 0.

1	1	1
0	0	0
0	1	1

Top two rows: all the 2x2 squares are irrelevant.

0	0	0
1	1	0
0	1	1

Column: one 2x2 square is irrelevant and the others have determinants of 0.

0	0	1
0	1	0
0	1	1

Having a full 2x2 square of '0's:

Bottom two rows: all 2x2 squares have determinants of 0.

1	1	1
0	0	1
0	0	1

Top two rows: two 2x2 squares are irrelevant and the other one will have a determinant of 0.

1	0	0
1	0	0
0	1	1

Winning Patterns:

Each of the outcomes on the next page will match one of these patterns:

'+' Shape

0	1	0
1	1	1
0	1	0

This is similar to when there is a full 2x2 square of '0's in the top two rows.

T-Shape

1	1	1
0	1	0
0	1	0

This is essentially a 2x2 square of '0's in a 3x3 grid

1	0	0
1	1	1
1	0	0

This is similar to when there is a full 2x2 square of '0's in the top two rows.

Square of '0's

0	1	1
1	0	0
1	0	0

See "Key Ideas" above.

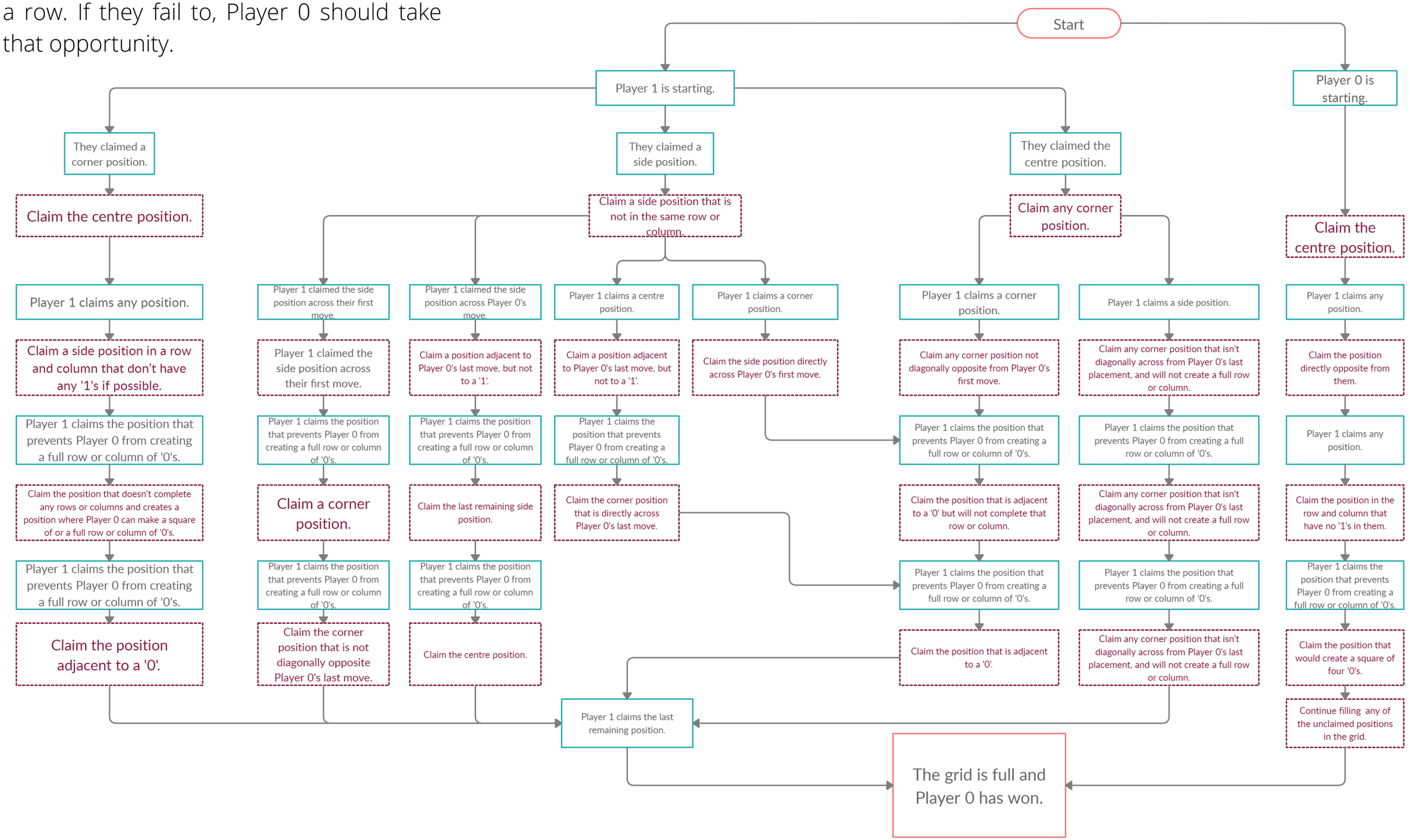
3-in-a-row

1	1	1
0	0	0
0	1	1

See "Key Ideas" above.

Flowchart for Player 0 to always win a game of Determinant Tic Tac Toe:

If Player 1 plays optimally, then they will block any attempts to achieve three '0's in a row. If they fail to, Player 0 should take that opportunity.



PART B: Does this method work in an nxn grid when n > 3?

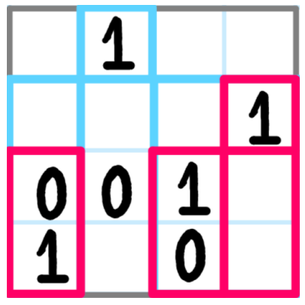
Answer:

No, Player 0 cannot use the same method as they did when n = 3.

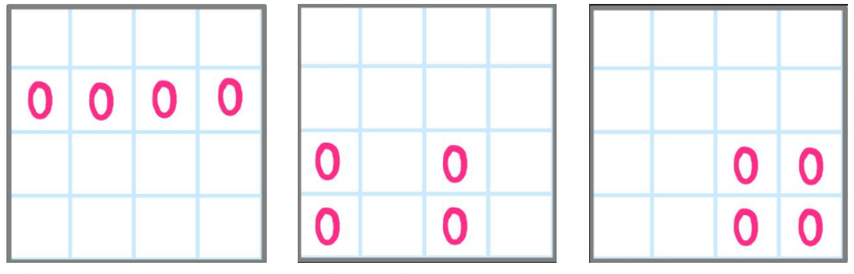
When n > 3, the patterns to win more difficult to reach - the larger square gives Player 1 a better opportunity to block attacks.

Additionally, Player 0 will need to account for many more combinations of 2x2 and 3x3 grids than when n = 3.

2x2 squares can be created with squares that aren't necessarily adjacent.



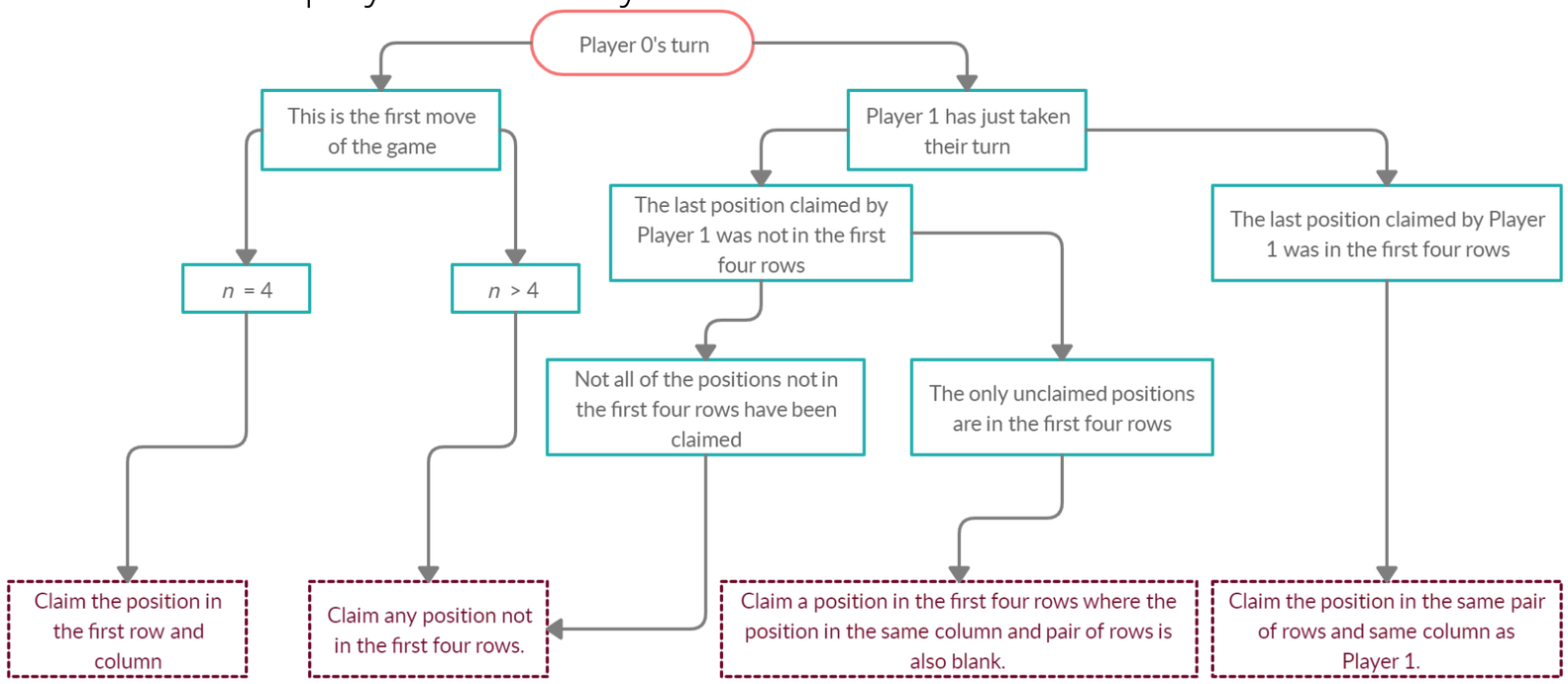
Some possible patterns for Player 0 to win:



Player 1 can easily block attempts to reach these patterns.

However, is there a way for Player 0 to always win?

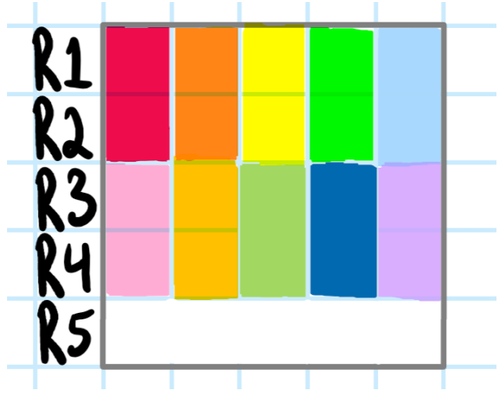
Method for player 0 to always win when n>3:



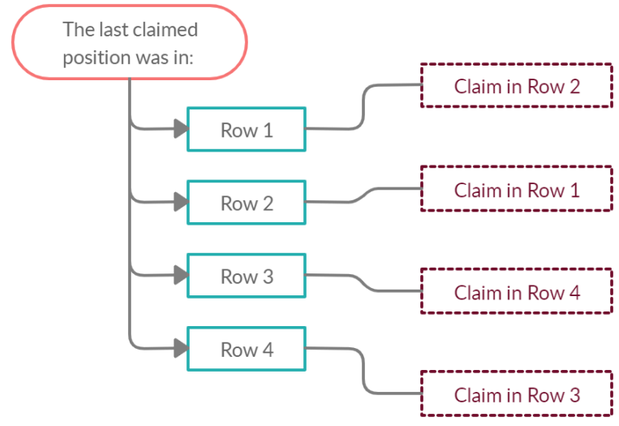
Pair One - Row 1 and Row 2

Pair Two - Row 3 and Row 4

Claim the position in the same colour as Player 1's last move.



Determining which row the claimed position should be in:



Linear Dependence:

When the vectors in a matrix are linearly dependent, the determinant of that matrix is 0.
Here is an example:

```
A -> [4 6]
B -> [2 3]
```

These two vectors are linearly dependent because A = 2B.
The matrix therefore has the determinant of 0.
This can be tested with the equation of:
4x3 - 6x2 = 12-12 = 0

Another example:

```
A -> [3 4 7]
B -> [2 5 1]
C -> [5 9 8]
```

These three vectors are linearly dependent because A+B = C, and so this matrix also has a determinant of 0.

When following this method, the vectors to take note of are:
R1 -> (row 1) R2 -> (row 2)
R3 -> (row 3) R4 -> (row 4)

These vectors are linearly dependent because
R1+R2 = R3 + R4 = (a vector of all '1's)
Any elements outside of those are irrelevant because the determinant is already 0.

Example where Player 1 begins and Player 0 follows the method given:

0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
0	1	0	1	1
1	1	0	1	0

R1 -> [0 1 1 0 1]
R2 -> [1 0 0 1 0]
R3 -> [1 0 1 0 0]
R4 -> [0 1 0 1 1]

R1 + R2 -> [1 1 1 1 1]
R3 + R4 -> [1 1 1 1 1]
R1 + R2 = R3 + R4

Conclusion

The question we addressed required us to find a method for a player of Determinant Tic Tac Toe to consistently ensure a victory, for both nxn grids where n=3 and n>3. In Part A, we developed a single method where if Player 0 plays optimally, they will always win regardless of what Player 1 does. In Part B, we found that Player 0 is unable to utilise the same method, prompting us to further our understanding. Using the concept of linear dependence, we formed a new method that ensured Player 0 would win when n > 3. The sources we used for research are listed below:

References (APA Style)

Khan Academy. (2009). Introduction to linear independence | Vectors and spaces | Linear Algebra | Khan Academy [Video file]. Retrieved from https://www.youtube.com/watch?v=CrV1xCWdY-g&feature=youtu.be.

Linear Independence. (n.d.). In Wikipedia. Retrieved from https://en.wikipedia.org/wiki/Linear_independence.

Testing for Linear Dependence of Vectors. (1996). In Wikipedia. Retrieved from http://sites.science.oregonstate.edu/math/home/programs/undergrad/CalculusQuestStudyGuides/vcalc/lindep/lindep.html.

Health Risk Factors between Socioeconomic Groups

Synthia Ekram, Aris Han and Hui Jan Kwan

Year 9
Methodist Ladies' College

Abstract

The mortality from coronary heart disease, cerebrovascular disease, lung cancer and chronic obstructive pulmonary disease is higher among adults in low socioeconomic groups compared to high socioeconomic groups in Australia. This might be due to the difference in the prevalence of risk factors of these diseases in different socioeconomic groups. We aimed to examine if there is a statistically significant difference in health risk factors between all socioeconomic groups, and to explore in particular, the degree of difference between the highest and lowest socioeconomic groups in Australia. Utilising Microsoft excel we analysed our data collected from the National Health Survey of Australia using the Chi Square Test of Independence. There was a significant difference in inactivity in all socioeconomic groups (p=0.03). Among the highest and lowest socioeconomic groups, we found a statistically significant difference in inactivity (p=0.002), obesity (p=0.05) and daily smoking (p=0.02). There was no significant difference in high blood pressure and risky drinking between socioeconomic groups or between the highest and lowest socioeconomic groups. Policies should be developed to bridge the gap between these health risk factor inequalities between socioecomomic groups especially between the highest and lowest groups.

Research question

Is there a difference in health risk factors between all socioeconomic groups, and especially what the degree of difference among the highest and lowest socioeconomic groups in Australia?

Background and Significance

Socioeconomic groups are based on the area of residence using the Australian Bureau of Statistics (ABS) Index of Relative Socio-economic Disadvantage[1]. These socioeconomic groups are based on higherchical ranking and this hierarchy may result in inequalities in health. In most countries people from high socioeconomic groups lives longer than people from low socioeconomic groups[2]. Similarly, in Australia, men and women in the highest socioeconomic group were found to live 2.6 years longer than men and women in the lowest socioeconomic group[3]. Mortality from coronary heart disease, cerebrovascular disease, lung cancer and chronic obstructive pulmonary disease is much higher in low socioeconomic groups compared to high socioeconomic groups in Australia[1]. The risk factors for these diseases are: physical inactivity, obesity, high blood pressure, risky drinking and daily smoking[4,5].

This has not only been in Australia. Studies from around the world have used a similar hypothesis that changes in individual risk factors among socioeconomic groups, such as physical inactivity,

smoking, obesity, and alcohol abuse, are contributing factors for disparities of mortality among these socioeconomic groups[2]. Thus, we were interested in examining these health risk factors because it is important to increase life expectancy, quality of life and plan a way to improve the health system so that it can cater to everyone's individual needs. Consequently, if we want to fulfil the health needs of a specific group of the population, we need to know what the risk factors of diseases are in this group.

As a result, we aimed to examine if there is a difference in health risk factors between different socioeconomic groups, especially if the difference is more prominent between the highest and lowest socioeconomic groups in Australia. This research will help us identify if there is a difference in health risk factors between all socioeconomic groups and if we find a difference, we can offer more targeted management for these risk factors in the appropriate socioeconomic group.

Analysis

We chose the Australian Institute of Health and Welfare (AIHW) data on the "Prevalence of health risk factors, by socioeconomic group" to perform our analysis (Appendix A)[1]. This was a large scale survey conducted in all states and territories and across urban, rural and remote areas of Australia (other than very remote areas) from July 2014 to June 2015, and included around 19,000 people in nearly 15,000 private dwellings. Since these were categorical data we performed the Chi-Square Test to check if there is a statistically significant difference in the prevalence of health risk factors by socioeconomic groups. The Chi-Square test is most useful when analysing survey data, and since the AIHW data were collected from the National Health Survey of Australia, we preferred to do the Chi-Square Test. The formula for Chi square test is:

$$\chi^2 = \sum \frac{(O-E)^2}{E}$$

The assumption of Chi-Square test is that the data in the cells should be frequencies, or counts of cases rather than percentages. The categories of the variables are mutually exclusive[6]. Thus we converted the health risk factors data provided in percentages in the AIWH table to frequencies or count of observation in that group.

We did these analyses in two stages; first, we examined the difference between all socioeconomic groups and then between the lowest and highest socioeconomic groups. All of our tests are two-tailed and the level of significance was $p \leq 0.05$. This means that if the p-value is more than 0.05, there is no significant difference between the test groups.

Where the Chi-Square test was significant in showing the difference between health risk factors and socioeconomic groups, we performed the Cramer's V or phi (φ) test. Cramer's V (or Cramer's Phi) is a measure of the association between nominal variables. The formula for the Cramer's V test is:

$$\text{Cramer's V} = \sqrt{\frac{\chi^2}{n \cdot \min(r-1, c-1)}}$$

The interpretation for Cramer's V is: 0.01 means small difference, 0.30 means medium difference and 0.50 means larger difference since we only have 2 columns.

We performed all these tests using Microsoft Excel.

Results

Table 1a: Chi-Square Test of independance for prevalence of inactivity or insufficient activity by all socioeconomic groups

Category	Inactive	Not inactive	Total
	Observed		
Lowest	76.1	23.9	100
2	69.1	30.9	100
3	68.3	31.7	100
4	61.9	38.1	100
Highest	55.8	44.2	100
Total	331.2	168.8	500
	Expected		
Lowest	66.24	33.76	
2	66.24	33.76	
3	66.24	33.76	
4	66.24	33.76	
Highest	66.24	33.76	
	Test results		
P-value	0.031194747		
Pearson's chi	10.61914738		
Cramer's V	0.145733643 ≈ 0.1		

- Socioeconomic groups are based on the area of residence using the Australian Bureau of Statistics (ABS) Index of Relative Socio-economic Disadvantage (where 1 denotes the lowest group and 5 denotes the highest group).
- Inactive or insufficiently active classification is based on self-reported exercise undertaken for fitness, sport or recreation in the last week.

Table 1b: Chi-Square Test of independance of prevalence of inactivity or insufficient activity between the lowest and highest socioeconomic group

Category	Inactive	Not inactive	Total
	Observed		
Lowest	76.1	23.9	100
Highest	55.8	44.2	100
Total	131.9	68.1	200
	Expected		
Lowest	65.95	34.05	
Highest	65.95	34.05	
	Test results		
P-value	0.002452751		
Pearson's Chi	9.175508968		
Cramer's V	0.214190441 ≈ 0.2		

- Socioeconomic groups are based on the area of residence using the Australian Bureau of Statistics (ABS) Index of Relative Socio-economic Disadvantage (where 1 denotes the lowest group and 5 denotes the highest group). Analyses were done for lowest vs highest groups.
- Inactive or insufficiently active classification is based on self-reported exercise undertaken for fitness, sport or recreation in the last week.

Table 1 shows the Chi-Square Test of independance of the prevalence of inactivity or insufficient activity by all socioeconomic groups (Table 1a) and the prevalence of inactivity or insufficient activity between lowest and highest socioeconomic group (Table 1b). The level of physical activity was different according to socioeconomic groups. The lowest socioeconomic group had the highest rate of physical inactivity (76.1%) and the highest socioeconomic status group has the lowest rate of physical inactivity (55.8%). When we compared all of the socioeconomic groups, we found the difference for the prevalence of inactivity was statistically significantly different between these groups (p=0.03). However, the magnitude of this difference between the groups is small (Cramer's V ≈ 0.1). Similarly, the difference between the level of inactivity was different between the lowest and highest socioeconomic groups (p=0.002). The magnitude of the difference between these two groups was small (Cramer's V ≈ 0.2).

Table 2a: Chi-Square Test of independance of prevalence of obesity in all socioeconomic groups

Category	Obese	Not obese	Total
	Observed		
Lowest	33.7	66.3	100
2	30.2	69.8	100
3	29.1	70.9	100
4	25.4	74.6	100
Highest	21.4	78.6	100
Total	139.8	360.2	500
	Expected		
Lowest	27.96	72.04	
2	27.96	72.04	
3	27.96	72.04	
4	27.96	72.04	
Highest	27.96	72.04	
	Test Results		
P-value	0.353207605	Not significant	

- Socioeconomic groups are based on the area of residence using the Australian Bureau of Statistics (ABS) Index of Relative Socio-economic Disadvantage (where 1 denotes the lowest group and 5 denotes the highest group).
- Obesity classification is based on measured weight and height.

Table 2b: Chi-Square Test of independance of Prevalence of obesity in the lowest and highest socioeconomic group

Category	Obese	Not obese	Total
	Observed		
Lowest	33.7	66.3	100
Highest	21.4	78.6	100
Total	55.1	144.9	200
	Expected		
Lowest	27.55	72.45	
Highest	27.55	72.45	
	Test Results		
P-value	0.05156475		
Pearson's Chi	3.789834406		
Cramer's V	0.137655992		

- Socioeconomic groups are based on the area of residence using the Australian Bureau of Statistics (ABS) Index of Relative Socio-economic Disadvantage (where 1 denotes the lowest group and 5 denotes the highest group). Analyses were done for lowest vs highest groups
- Obesity classification is based on measured weight and height.

Table 2 shows the Chi-Square Test of independance for the prevalence of obesity in all socioeconomic groups (Table 2a) and the prevalence of obesity between the lowest and highest socioeconomic group (Table 2b). The level of obesity was different according to socioeconomic groups. The lowest socioeconomic group had the highest obesity rate (33.7%) and the highest socioeconomic status group has the lowest obesity rate (21.4%). When we compared all of the socioeconomic groups in regards to obesity prevalence, we found the difference for the prevalence

of obesity was not statistically significant between these groups (p=0.35). The difference for prevalence of obesity in the lowest and highest socioeconomic group was statistically significantly different (p=0.05). The Cramer's V showed that this difference was low (Cramer's V≈0.1).

Table 3a: Chi-Square Test of independance of Prevalence of High blood pressure in all socioeconomic groups

Category	HBP	Not HBP	Total
	Observed		
Lowest	25.5	74.5	100
2	25	75	100
3	21.1	78.9	100
4	23.1	76.9	100
Highest	20.8	79.2	100
Total	115.5	384.5	500
	Expected		
Lowest	23.1	76.9	
2	23.1	76.9	
3	23.1	76.9	
4	23.1	76.9	
Highest	23.1	76.9	
	Test Results		
P-value	0.902052822	Not significant	

- Socioeconomic groups are based on the area of residence using the Australian Bureau of Statistics (ABS) Index of Relative Socio-economic Disadvantage (where 1 denotes the lowest group and 5 denotes the highest group).
- High blood pressure classification is based on measured blood pressure. For the 24% of respondents aged 18 years and over who did not have their blood pressure measured, imputation was used to obtain blood pressure.

Table 3b: Chi-Square Test of independance of Prevalence of high blood pressure in the lowest and highest socioeconomic group

Category	HBP	Not HBP	Total
	Observed		
Lowest	25.5	74.5	100
Highest	20.8	79.2	100
Total	46.3	153.7	200
	Expected		
Lowest	23.15	76.85	
Highest	23.15	76.85	
	Test Results		
P-value	0.430739997	Not significant	

- Socioeconomic groups are based on the area of residence using the Australian Bureau of Statistics (ABS) Index of Relative Socio-economic Disadvantage (where 1 denotes the lowest group and 5 denotes the highest group). Analyses were done for lowest vs highest group
- High blood pressure classification is based on measured blood pressure. For the 24% of respondents aged 18 years and over who did not have their blood pressure measured, imputation was used to obtain blood pressure.

Table 3 shows the Chi-Square Test of independance of the prevalence of high blood pressure in all socioeconomic groups (Table 3a) and the prevalence of high blood pressure between the lowest and highest socioeconomic groups (Table 3b). The prevalence of high blood pressure was different according to socioeconomic groups. The lowest socioeconomic group had the highest number of people with high blood pressure (25.5%) and the highest socioeconomic status group has the lowest prevalence of high blood pressure (20.8%). When we compared all of the socioeconomic groups, we found the difference for the prevalence of high blood pressure was not statistically significant (*p*=0.9). Similarly, the difference we observed for the prevalence of high blood pressure in the lowest and highest socioeconomic group was not statistically significant (*p*=0.43).

Table 4a: Chi-Square Test of independance of Prevalence of risky drinking in all socioeconomic groups

Category	Drinking	Not drinking	Total
	Observed		
Lowest	15.8	84.2	100
2	16.8	83.2	100
3	17.3	82.7	100
4	17.9	82.1	100
Highest	17.6	82.4	100
Total	85.4	414.6	500
	Expected		
Lowest	17.08	82.92	
2	17.08	82.92	
3	17.08	82.92	
4	17.08	82.92	
Highest	17.08	82.92	
	Test Results		
P-value	0.995711121	Not significant	

- Socioeconomic groups are based on the area of residence using the Australian Bureau of Statistics (ABS) Index of Relative Socio-economic Disadvantage (where 1 denotes the lowest group and 5 denotes the highest group).
- Lifetime risky drinking classification is based on self-reported alcohol use.

Table 4b: Chi-Square Test of independance of Prevalence of risky drinking in the lowest and highest socioeconomic group

Category	Drinking	Not drinking	Total
	Observed		
Lowest	15.8	84.2	100
Highest	17.6	82.4	100
Total	33.4	166.6	200
	Expected		
Lowest	16.7	83.3	
Highest	16.7	83.3	
	Test Results		
P-value	0.732912954	Not significant	

- Socioeconomic groups are based on the area of residence using the Australian Bureau of Statistics (ABS) Index of Relative Socio-economic Disadvantage (where 1 denotes the lowest group and 5 denotes the highest group). Analyses were done for lowest vs highest groups
- Lifetime risky drinking classification is based on self-reported alcohol use.

Table 4 shows the Chi-Squared Test of the prevalence of risky drinking in all socioeconomic groups (Table 4a) and the prevalence of risky drinking between the lowest and highest socioeconomic group (Table 4b). The level of risky drinking was different according to socioeconomic groups. The lowest socioeconomic group had the lowest rate of risky drinking (15.8%) andthe highest socioeconomic status group has the highest rate of risky drinking (17.6%). When we compared all of the socioeconomic groups, we found the difference for the prevalence of high blood pressure was not statistically significant (p=0.996). Similarly, the difference for the prevalence of risky drinking in the lowest and highest socioeconomic group was not statistically significant (p=0.73).

Table 5a: Chi-Squared Test of Prevalence of daily smoking in all socioeconomic groups

Category	Smoking	Not smoking	Total
		Observed	
Lowest	17.7	82.3	100
2	14.6	85.4	100
3	12	88	100
4	10.1	89.9	100
Highest	6.5	93.5	100
Total	60.9	439.1	500
		Expected	
Lowest	12.18	87.82	
2	12.18	87.82	
3	12.18	87.82	
4	12.18	87.82	
Highest	12.18	87.82	
		Test Results	
P-value	0.14572205	Not significant	

- Socioeconomic groups are based on the area of residence using the Australian Bureau of Statistics (ABS) Index of Relative Socio-economic Disadvantage (where 1 denotes the lowest group and 5 denotes the highest group).

- Daily smoking classification is based on self-reported tobacco use.

Table 5b: Chi-Square Test of independance of Prevalence of daily smoking in the lowest and highest socioeconomic group

Category	Smoking	Not smoking	Total
		Observed	
Lowest	17.7	82.3	100
Highest	6.5	93.5	100
Total	24.2	175.8	200
		Expected	
Lowest	12.1	87.9	
Highest	12.1	87.9	
		Test results	
P-value	0.015166617		
Pearson's Chi	5.897009186		
Cramer's V	0.171712102 ≈ 0.2		

- Socioeconomic groups are based on the area of residence using the Australian Bureau of Statistics (ABS) Index of Relative Socio-economic Disadvantage (where 1 denotes the lowest group and 5 denotes the highest group). Analyses were done for lowest vs highest groups
- Daily smoking classification is based on self-reported tobacco use.

Table 5 shows the Chi-Square Test of independance of the prevalence of daily smoking in all socioeconomic groups (Table 5a) and the prevalence of daily smoking between the lowest and highest socioeconomic group (Table 5b). The level of daily smoking was different according to socioeconomic groups. The lowest socioeconomic group had the highest rate of smoking (17.7%) andthe highest socioeconomic group has the lowest smoking rate (6.5%). When we compared all of the socioeconomic groups, we found the difference for the prevalence of smoking was not statistically significant (p=0.15). However, the difference for prevalence of smoking in the lowest and highest socioeconomic group was statistically significant (p=0.02). The Cramer's V showed that this difference was low (Cramer's V ≈ 0.2).

Discussion

We found that there was a significant difference in inactivity in all socioeconomic groups but no significant difference in the prevalence of obesity, high blood pressure, risky drinking and daily smoking were found. When we compared the highest and lowest socioeconomic groups, we found a significant difference in inactivity, obesity and daily smoking, but no significant difference in high blood pressure and risky drinking.

Our results are supported by other studies. For example, all over the world obesity, high blood pressure, risky drinking and smoking are higly prevalent in low socioeconomic group than high socioeconomic groups[7].

In previous research, it has been shown that countries that provide universal health coverage, i.e. Canadia, people from low socioeconomic groups use less health care than their health care needs[8]. Since Australia has the similar health care system to that of Canada's, it might be that Australians from low socioeconomic group use the health care system less and have poor control on disease risk factors. Thus, closing the gaps between the highest and lowest socioeconomic groups in health behaviours especially for obesity and daily smoking should be a priority in Australia. Furthermore, irrespective of socioeconomic groups, the government needs to make policies to encourage all Australians to participate more in physical activities. Even in the highest socioeconomic group, the highest prevalence of inactivity is almost 60%, a percentage which was increased further with each socioeconomic group and was highest in the lowest socioeconomic group (76%). Additionally, more research should be done to to understand more about why these gaps exist between the socioeconomic groups.

Conclusion

The health risk factors of Australian people differ according to socioeconomic groups. The difference is most prominent between the highest and lowest socioeconomic groups. Policies should be developed to bridge the gap between these health risk factors, especially for inactivity, obesity and smoking.

References

1. ABS (Australian Bureau of Statistics) 2015. National Health Survey: first results, 2014-15. Cat. no. 4364.0.55.001. Canberra: ABS. Data cube table 6.3.
2. Annual Review of Public Health. Increasing Disparities in Mortality by Socioeconomic Status. Vol. 39:237-251.
3. Clarke P & Leigh A 2011. Death, dollars and degrees: socioeconomic status and longevity in Australia. Economic Papers 30(3):348–55.
4. Fryar CD, Chen T-C, Li X. Prevalence of uncontrolled risk factors for cardiovascular disease: United States, 1999–2010
5. Ada's Medical Knowledge Team. Cardiovascular Disease Risk Factors. November 20, 2018
6. McHugh M. L. (2013). The chi-square test of independence. Biochemia medica, 23(2), 143–149. https://doi.org/10.11613/bm.2013.018.
7. Psaltopoulou T, Hatzis G, Papageorgiou N, Androulakis E, Briasoulis A, Tousoulis D. Hellenic J Cardiol. 2017 Jan-Feb;58(1):32-42.
8. Dunlop S, Coyte PC, McIsaac W. Soc Sci Med. 2000 Jul;51(1):123-33.

Appendix A

			Socioeconomic group						
Health risk factor	**Year**		**Lowest**	**2**	**3**	**4**	**Highest**	**Australia**	**Rate ratio: lowest/highest socioeconomic group**
Inactive or insufficiently active	2014–15	% aged 18+ [95% CI]	76.1 [74.0–78.2]	69.1 [66.6–71.6]	68.3 [64.8–71.8]	61.9 [59.1–64.7]	55.8 [53.2–58.4]	65.0 [63.6–66.4]	1.4
Obese	2014–15	% aged 18+ [95% CI]	33.7 [31.1–36.2]	30.2 [27.9–32.4]	29.1 [27.0–31.3]	25.4 [23.3–27.4]	21.4 [19.0–23.8]	27.9 [26.9–28.8]	1.6
High blood pressure	2014–15	% aged 18+ [95% CI]	25.5 [23.7–27.3]	25.0 [22.7–27.3]	21.1 [19.7–22.5]	23.1 [20.7–25.5]	20.8 [18.5–23.1]	22.9 [21.9–23.9]	1.2
Lifetime risky drinking	2016	% aged 14+ [95% CI]	15.8 [14.5–17.1]	16.8 [15.4–18.2]	17.3 [15.8–18.8]	17.9 [16.4–19.4]	17.6 [16.1–19.1]	17.1 [16.5–17.7]	0.9
Daily smoking	2016	% aged 14+ [95% CI]	17.7 [16.3–19.1]	14.6 [13.2–16.0]	12.0 [10.8–13.2]	10.1 [9.0–11.2]	6.5 [5.7–7.3]	12.2 [11.6–12.8]	2.7

Melbourne University
Mathematics and Statistics Research Competition 2020

Presbyterian Ladies' College
Question 2 - Colouring Squares

Team Members: Liah Wu, Yunaa Tae

July 26, 2020

CONTENTS

1. Question 2 - Colouring Squares Report 1

2. Program and Results 17

2.1 Program Detail 17

2.2 Program Source Code 17

2.3 Test Cases and Results 18

Bibliography 19

Appendix 21

1. Question 2 - Colouring Squares

The Problem

For an $m \times n$ grid, and $k \geq 2$ colours, how many ways are there to colour a grid so that no neighbouring* squares are the same colour?

**Neighbouring squares are squares immediately to the left and right, or directly above and below, the original square.*

Solution

Different cases

We will solve the problem algebraically, moving in one direction, from the top left square to the bottom right square. We will move down the left most column, then go back up into the second column from the left, and start at the top square, but never move to the left or upwards. Also, the number of possible colours that a square could be will be written in the square itself, and the number of possibilities is dependent on the squares already filled with some number of colours.

To start off, look at a $1 \times n$ grid, as shown below.

k	k - 1	k - 1	k - 1	k - 1

Say there are k colours. Then going from left to right, the first square (left-most) would have k possible colours. The second square, which cannot be filled with the same colour as the first square, can be filled with (k-1) colours. Same goes to the third square, which could be filled with any of the k colours except for the colour to the left (which is the second square) which means the total possible colours is also (k-1). Following this idea, for a $1 \times n$ grid, there are (n-1) values of (k-1), and as the first square has k possibilities, multiplied by the k possibilities for the first square, giving, $k(k - 1)^{n - 1}$ ways to colour a $1 \times n$ grid.

However, when $m>1$, there can be 2 general cases for the number of solutions possible. They are shown below.

Note: When saying "diagonally adjacent" we mean a square and the square that is diagonally opposed to it (sharing only 1 vertex, but no edges), as shown in the green and blue coloured squares in the grid below. Also, this case only exists when $k \geq 3$*, as the diagonally adjacent colours must be both different to each other, and different to the square to the left/top respectively, as shown in yellow in the diagram below.*

Case 1

No squares on the grid which are diagonally adjacent are coloured the same.

We will use a 4×3 grid to show this. As proved above, the number of combinations for the first column is k for the left-most square, and then (k - 1)for the following squares, as shown below.

k			
k - 1			
k - 1			

Now, going down to the second column, the top square on the second column couldn't be the same colour as the top-left block, or the diagonally adjacent square to the bottom-left. This gives (k - 2) possibilities for this square. The second square from the left on the second row could not be the colour of the square above, next to, or diagonally adjacent to (two squares since from the squares of the colours we already determined, both of which are in the first row, the top-left and bottom-left), so this square could be coloured in (k-4) ways. This goes all the way until we reach the third and last row. When we get to the last row, the only squares that affect the one we're on about are the squares to the top, left and the diagonally adjacent square to the top-left, so this then gives

(k-3) possible colourings. Repeating this process with the next row gives the same pattern of colouring, as shown below.

k	k - 2	k - 2	k - 2
k - 1	k - 4	k - 4	k - 4
k - 1	k -3	k - 3	k - 3

But after some thinking, you will realise that there are a lot of mistakes in multiplying this all together. Look at the grid below, and focus on the square x. Before, we assumed that there were (k - 4) possibilities, but we know that while $b \neq x$, $d \neq x$, we do not know that $b \neq d$, so while x might be (k - 4), it could also be (k - 3), and if x was (k- 3), the squares near it would have another value, since it means that b and d would have to be equal.

b	c	d	
a	x		

From this, it is evident that there is no such obvious formula for Case 1 grids, and even if there was a formula, it would be quite unpleasant.

Case 2

There are squares on the grid which are diagonally adjacent and are coloured the same.

The *2×2* grid shown below is a typical case of a pair of diagonally adjacent squares coloured the same.

	a
a	

In a general case, this is what it would look like (in terms of possible colourings):

k	1
k - 1	k - 1

This is as the first column, has one k and (k-1) as proven above. The top square in the second column would have 1 possibility, because it would have to be dependent upon the diagonally adjacent square that we have already determined has (k-1) possible colourings. No matter what the diagonally adjacent square is coloured, it would have to be the same colour. However, the bottom right square also has (k-1) possibilities, as the square to the top and left of it is the same colour, so that square can only not be of one colour, and hence (k-1) possibilities.

However, the formula is not straightforward either, because in the grid, the number of diagonally-adjacent squares being the same colour and and where they would be placed would vary, which would mean the formula would be an unpleasant one, if one was even possible. Therefore, a general solution must be formed.

The General Solution

To reach a general solution, we will look at the cases where $m = 1$, 2, 3 to get a sense of what could be done to form a general method, or hopefully a formula. Previously, we have already solved the case for an $m \times n$ graph if $m = 1$, so we will start at $m = 2$ and then move on to solve the case for if $m = 3$.

$2 \times n$ grid

k_{a}	
k_{b}	

As explained above, the number of possibilities for the first column would be $k(k\text{-}1)$. In the next column, the top square must be of a different colour as the square to the top-left, so in this case it must not be colour k_{a}. Now, let's split the colour choice of the top-right square into two cases - if it is k_{b} (the same colour as the square diagonally-opposite to the bottom-left) or if it is a different colour to k_{a} and k_{b}.

Case 1

If the top square of the second column was coloured with k_b as shown below.

k_{a}	k_{b}
k_{b}	

If the colour was k_{b}, then the square below could be $k_{\mathrm{a,c,d...}}$ etc, which is just any colour that is not k_{b}. This is shown below.

k_{a}	k_{b}
k_{b}	$k_{\mathrm{a,c,d...}}$

For the top-left square, you can choose any colour, hence there are k possibilities. Since in this case the squares on the top-right and bottom left are of the same colour, and are both adjacent to the top-left square, there are (k-1) possible colourings in total for both of those squares. For the bottom-right square, it is adjacent to two squares, however because of the case, they are of the same colour. Hence, the bottom right square also has (k-1) possibilities of colouring it. Hence, this gives the number of possibilities per square as follows.

k	1
(k-1)	(k-1)

Therefore, if the top square of the second column was coloured with k_{b}, there would be (k-1) possible colourings for the second column, and in a 2×2 grid case, there would be k(k-1)^2 possible colourings of the entire grid.

Case 2

If the top block of the second column was coloured with k_x (where $x \neq a, b$), as shown below.

k_{a}	$k_{\mathrm{x}\ (x \neq a, b)}$
k_{b}	

Say the top square of the second column was any other colour apart from k_{a} (because it is adjacent to a square using that colour) and k_{b} (not the same as case 1), this means that the block to the bottom of that square (the bottom-right square) could be any colour that wasn't the same colour as the one to its top or k_{b} (the square to the left), as shown below.

k_{a}	$k_{\mathrm{x}\ (x \neq a, b)}$
k_{b}	$(k_{\mathrm{a,c,d,...}})$

For the top-left square, you can choose any colour, hence there are k possibilities. Since in this case the squares on the top-right and bottom left are of different colours, the bottom square in the first column has then (k-1) possibilities, and because it's not of the same colour as the top-right square, the top-right square has (k-2) possibilities. The bottom-right square is adjacent to two squares, both of which in this case are of different colours, and hence also have (k-2) possibilities. This is shown below.

k	(k-2)
(k-1)	(k-2)

In case 2, the first column has $k(k$-1) possibilities, and the second column has (k-2)(k-2) possible colourings. Hence, there are (k-2)^2 possible colourings for the second column, if the top square wasn't $k_{\mathrm{b,a}}$, and we have the formula for a 2×2 grid, which is $k(k$ - 1)(k- 2)^2.

Now, we are going to piece together the formula that covers the total cases for a 2×n grid. By looking at the 2×2 grid case, we discover the second column could be coloured in ((k - 1) + (k-2)^2) = k^2 - 3k + 3 ways. By repeating the process above for a 2×3 grid, we discover that still the number of possibilities for the 3rd column would be (k^2 - 3k + 3). This is because the colouring of the next column would be only dependent on how the previous column was coloured, not on how any other columns could be coloured, so we could do this (n-1) times for a 2×n grid, and find the formula for total number of possible colourings of a 2×n grid as $k(k$ - 1)(k^2 - 3k + 3)$^{n\text{ - }1}$.

$3\times n$ grid

Unlike the $2\times n$ grid, where there must be 2 different colours, a column could contain 2 or 3 different colours. Let a, b, c represent different colours. Then the first column could be in two forms: a-b-a, or a-b-c (going down the column). Call "a-b-c" "*Type* 1", t_1 for short, and "a-b-a" as "*Type* 2", t_2 for short. On the grid, if the previous column, or the column to its left is of t_1, then the next column could be of either form, so either of t_1 or t_2. Similarly, if the previous column was of form t_2, then the next column can also be of either form t_1 or t_2.

So first we must know how many possible combinations there are in the instance that a t_1 follows a t_1 or t_2 column, and also the number of possibilities for any t_2 that follows a t_1 or t_2 coloured column. Let m_{n} be the number of possible colourings of a $3\times n$ grid ending with a column of t_1 (ie. the right-most column in the grid is of type t_1), and let x_{n} be the number of possible colourings of a $3\times n$ grid that ends in a column of t_2 (ie the right-most column in the grid is of type t_2).

This means that $m_{\mathrm{n+1}}$ would equal the total number of ways that go from a t_1 in the previous column to t_1 in the last column multiplied by the total number of possibilities m_{n}, plus the number of ways a t_2 is followed by a t_1, multiplied by the number of possibilities of x_{n}. Following a similar fashion, $x_{\mathrm{n+\,1}}$ would be the number of ways a t_2 could follow a t_2 multiplied by x_{n}, plus ways to get from t_1 to t_2 multiplied by the number of possibilities m_n. As an equation, this would look like as shown below.

$$m_{\mathrm{n+1}} = (\text{ways } \mathrm{t}_1 \rightarrow t_1)m_{\mathrm{n}} + (\text{ways } t_2 \rightarrow t_1)x_{\mathrm{n}}$$
$$x_{\mathrm{n+1}} = (\text{ways } t_2 \rightarrow \mathrm{t}_2)x_{\mathrm{n}} + (\text{ways } t_1 \rightarrow t_2)m_{\mathrm{n}}$$

To calculate, all of the above can be simplified into matrices. But first, we need to know the number of possible colourings for a t_1 that follows either a t_1 or t_2, and also the number of possibilities for any t_2 that follows a t_1 or t_2.

Note: When using '→', we mean the number of ways the column type to the left of the arrow could be added after the column type to the right of the arrow. ie. $t_1 \rightarrow$ t_1 means the number of ways a t_1 column could appear to the right of another t_1 column, to add another column to the already existing grid.

$t_1 \rightarrow t_1$

a		a
b	$\rightarrow$	b
c		c

There are 4 cases for when a t_1 follows a t_1 column.

Case 1
all 3 colours in the next column are different to the previous column (ie. not a, b or c)

That means the top block of the next column only cannot be a, b or c, and hence has k - 3 possible colours to choose from. The next one down also cannot be any of the 3 colours a, b or c, but also cannot be the same colour as the square above, so k - 4. The last block must not be the same colour as a, b, c and also not the same as the above 2 squares (as identified by this case), and hence have k - 5 possible colourings. This is shown below.

a	k - 3
b	k - 4
c	k - 5

Therefore, there are $(k - 3)(k - 4)(k - 5) = k^3 - 12k^2 + 47k - 60$ total combinations that satisfy the rules of case 1.

Case 2

2 colours in the next column are different

No matter where the same colour is placed in the next column, the remaining two squares can't be any of the three colours a, b or c, so there are k - 3 possibilities, and hence the last square has k - 4 possibilities.

Then, no matter which colour remained the same, it could be in 2 other positions, and the remaining 2 empty blocks could then be filled with (k-3)(k-4) = k^2 - 7k + 12 possible combinations of 2 other colours.

As for the repeated colour, there are 3 possible colours to choose from to repeat (a, b or c), and for each colour there are 2 possible places where it can be placed, as it cannot be placed next to one of the same colour in the previous column. Hence, there are $3 \times s \times (k-3)(k-4) = 6 \times (k^2 - 7k + 12) = 6k^2$ - 42k + 72 possibilities.

Case 3

only one colour in the next column is different

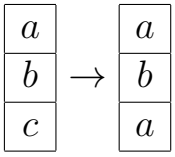
There would be three ways of choosing which the one different colour would be, and also there are three ways of choosing where the two of the original colours would be placed. The remaining block, as it must be a different colour to the original 3, has k - 3 possible ways of doing it. Now, multiplying the number of choices the 2 remaining colours could be and where they are placed, with the number of possibilities the remaining colour could be, there are a total of $9 \times (k-3) = 9k$ - 27 ways of colouring using this case.

Case 4
all colours in the next column are the same as the ones in the previous

When all three colours stay the same as the previous row (ie. using colours a, b and c ONLY), then there are only 2 ways of arranging the 3 colours that satisfy the rules. Those are b-c-a and c-a-b (from top to bottom). Hence, there are 2 solutions to this case.

Summing up the number of possibilities from a t_1 to another t_1 type colouring using the 4 cases above, we get k^3 - $12k^2$ + $47k$ - 60 + $6k^2$ - $42k$ + 72 + $9k$ - 27 + 2 = k^3 - $6k^2$ + $14k$ - 13 possibilities in total if we go from t_1 to t_1.

$t_1 \rightarrow t_2$



There are 3 cases possible when going from t_1 to t_2.

Case 1
all colours in the next column are different.

As there are only two colours used (a and b), then the two remaining squares have k - 3 and k - 4 different possibilities, since 3 colours are used in the previous t_1 column. This gives (k - 3)(k - 4) = k^2 - $7k$ + 12 possibilities.

Case 2
one colour in the next column is different to the ones in the previous column

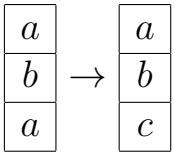
This case can be split into 2 subcases. If the colour that is the same as the previous column was a or c, then a or c could only be placed in the middle, and the remaining colour could be either of the k - 3 that haven't been used. Since this occurs to both a and c, there are $2(k$ - $3)$ = $2k$ - 6 possibilities. If b was the colour remaining, then b could only be at the top and bottom, and the middle block could be filled with k-3 colours. Summing these up gives $2k$ - 6 + k - 3 = $3k$ - 9 possibilities.

Case 3
the colours in the next column are all from the previous column

As no colour can be next to the same colour, b must be placed on the top and bottom block, and since a or c can be placed in the middle block, there are 2 total possibilities.

Therefore, summing the 3 cases up, we get k^2 - $7k$ + 12 + $3k$ - 9 + 2 = k^2 - $4k$ + 5 possibilities in total.

$t_2 \rightarrow t_1$



Case 1
all colours in the next column are different

Since there are only 2 colours in type t_2, columns, if the next column was to have all different colours, then the top square would have (k-2) possible colours, the next one would have (k-3) possible colours, while the last one would have (k-4) colours. Therefore, the total number of possible colourings would be (k - 2)(k - 3)(k - 4) = k^3 - $9k^2$ + $26k$ - 24.

Case 2

two colours in the next column are different from the ones that appeared in the previous column

Say 'a' was the colour that remained the same. Then 'a' could only be in the middle spot in the next column. This leaves the top and bottom blocks empty, and we could fill them with $(k - 2)(k - 3) = k^2 - 5k + 6$ possible arrangements of different colours. However, if 'b' remained the same, it could be in the top or bottom grids, and the 2 leftover grids could be filled with $(k - 2)(k - 3) = k^2 - 5k + 6$ possible arrangements of 2 colours. Hence, since there are 2 possible places where 'b' could be, the total number of possibilities is $2(k^2 - 5k + 6) = 2k^2 - 10k + 12$ possibilities. In total, there are $k^2 - 5k + 6 + 2k^2 - 10k + 12 = 3k^2 - 15k + 18$ total possibilities for this case.

Case 3

one colour in the next column is different

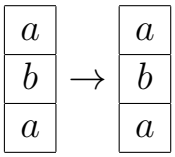
As one colour is different,that means that all the colours used in the previous column (a and b) must be used. There are only two placements of a and b, as shown below.

a			
b	or	a	
		b	

The block leftover in each case could be coloured in $(k - 2)$ ways, giving a total of $2(k - 2) = 2k - 4$ ways.

Summing the three cases up, we get k^3 - $9k^2$ + $26k$ - 24 + $3k^2$ - $15k$ + 18 + $2k$ - 4 = k^3 - $6k^2$ + $13k$ - 10 possibilities in total.

$t_2 \rightarrow t_2$



By observation, the bottom row can be ignored as it is of the same colouring scheme as the top row. Hence, this one can just be looked at as a $2 \times n$ grid. As explained above when figuring out the formula for a $2 \times n$ grid, there are k^2 - $3k$ + 3 possible colourings for the second column.

Hence, from the four formulas, we now know that:
m_{n+1}= (k^3 - $6k^2$ + $14k$ - 13)m_n + (k^3 - $6k^2$ + $13k$ -10)x_n
x_{n+1}= (k^2 - $3k$ + 3)x_n + (k^2 - $4k$ + 5)m_n.

To put them in a matrix, we correspond the values for $t_1 \rightarrow t_1$ and $t_1 \rightarrow t_2$ with $k(k$ - $1)(k$ - $2)$, since at the very start there are $k(k$ - $1)(k$ - $2)$ possible arrangements of t_1 columns(the first column on the most left). Similarly, we correspond the values for $t_2 \rightarrow t_1$ and $t_2 \rightarrow t_2$ with $k(k$ - $1)$, since that is the number of possible t_2 arrangements for the first column. We raise the square matrix (the matrix with the values we've calculated) to the $(n$-$1)th$ power, because we repeat the multiplication of this matrix $(n$-$1)$ times (excluding the first column). This becomes,

$$\begin{bmatrix} k^3 - 6k^2 + 14k - 13 & k^3 - 6k^2 + 13k - 10 \\ k^2 - 4k + 5 & k^2 - 3k + 3 \end{bmatrix}^{n-1} \begin{bmatrix} k(k-1)(k-2) \\ k(k-1)) \end{bmatrix}$$

To get the total number of possible solutions, the resulting 1×2 matrix will have two numbers, which need to be added together to reach the total number of solutions (the m_{n} and x_{n}).

This method can be applied to all other m×n grids, to find the number of colourings for a $m \times n$ grid. If $m = 4$, there are 5 possible types of ways a column could appear.

If we write them out from top to bottom with letters:
a-b-a-b
a-b-a-c
a-b-c-a
a-b-c-b
a-b-c-d

Then we would need to work out the number of colourings for each type of column following another type. As m gets larger, this also gets larger. For $m = 4$, we would need to calculate 25 values, however some could be retained from $m = 3$, like when $t_2 \rightarrow t_2$ was the same as the number of possibilities for the next column of a $2 \times n$ grid.

The General Solution - Summary

From this, the general method should be quite clear. For any $m \times n$ grid, we find the possible ways a singular column could appear, and then calculate the number of ways any 'type' of column could follow all other types. After doing this, we map them out in a square matrix, and correspond the values of each *'type'* following another with the possible ways 1 column could appear in any type (to show the column at the start). The matrix showing the number of ways different 'types' could be at the start should have only 1 column, generally in the form of $k(k - 1)...(k - i)$ depending on the number of different colours used. This matrix should follow the square matrix, as when matrices have to be multiplied together, the first matrix should have the same number of columns as the second matrices rows. Then, we raise the square matrix with the values we have calculated to the power of the number of columns taking away 1, because the starting column (the left-most column in the grid) has already been calculated in the matrix with a single column. The matrices would look something like the matrices listed out below.

$$\begin{bmatrix} t_1 \to t_1 & t_2 \to t_1 & ... & t_n \to t_1 \\ t_1 \to t_2 & ... & ... & t_n \to t_2 \\ ... & t_2 \to t_{n-1} & ... & ... \\ t_1 \to t_n & t_2 \to t_n & ... & t_n \to t_n \end{bmatrix}^{n-1} \begin{bmatrix} (total)t_1 \\ ... \\ ... \\ (total)t_n \end{bmatrix}$$

Finally, we calculate all of the possibilities we have listed out altogether, and add up the entries of the matrix we finally get, to reach the final total number of possible colourings of a grid. Also, m does not always have to be used to calculate the number of ways a column could appear in a $m \times n$ grid if that's how it's placed. It is always better to use n if $n > m$.

However, since this could become quite a tedious process if numbers become quite large, so a program on your computer or device could be used to reach the answer quicker. Details of the program are found on the following pages.

2. Program and Results

The purpose of the program is to check the number of combinations and visually display the results. It contains one source code file named 'question_2.py'. The program was developed using PyCharm as Development Environment.

2.1 Program Detail

The program was developed using Python version 3.7 within the PyCharm development environment. It consists of one main program, which calls 2 functions. It was written follow brute force strategy with python libraries. The main functions from libraries used to achieve the result are 'product' function to obtain Cartesian product and 'array' function to help generate and find all combinations.

First, the program will ask user for the values of row, column, and k. Then it will pass this value to the main program for execution and hence produce the result.

2.2 Program Source Code

There is only one program called "question_2.py" and it consists of the main program, and two functions called "total_possibilities" and "display_results". The purpose of "total_possibilities" function is to find all the possible combinations for the specified n, m and k values. "displa y_results" is used to display the results in a grid. The main program is used to ask for input of n, m and k, and to display the results.

Refer to the Appendix for the program source code.

2.3 Test Cases and Results

There are total of 10 cases were carried out. Since the number of test cases can go on indefinitely, only a number amount of cases have been carried out to ensure that the calculation is correct. The table below shows each test case result from the program.

Test Case	**m**	**n**	**k**	**Result**
1	1	3	4	36
2	4	1	3	24
3	2	2	2	2
4	2	2	4	84
5	2	2	5	260
6	2	2	6	630
7	2	3	2	2
8	2	3	3	54
9	3	3	3	246
10	3	2	6	13230

Bibliography

How to insert PDF into LaTeX – PDFConverters Official Website. 2020. How to insert PDF into LaTeX – PDFConverters Official Website. [ONLINE] Available at: https://www.pdfconverters.net/how-to/insert-pdf-to-latex/. [Accessed 23 July 2020].

itertools — Functions creating iterators for efficient looping — Python 3.8.5 documentation. 2020. itertools — Functions creating iterators for efficient looping — Python 3.8.5 documentation. [ONLINE] Available at: https://docs.python.org/3/library/itertools.html. [Accessed 23 July 2020].

Mathematics Stack Exchange. 2020. linear algebra - Converting recursive equations into matrices - Mathematics Stack Exchange. [ONLINE] Available at: https://math.stackexchange.com/questions/858268/converting-recursive-equations-into-matrices. [Accessed 25 July 2020].
Mathematics Stack Exchange. 2020. combinatorics - What's the name of a permutation where repetition is possible? - Mathematics Stack Exchange. [ONLINE] Available at: https://math.stackexchange.com/questions/2550416 /whats-the-name-of-a-permutation-where-repetition-is-possible. [Accessed 23 July 2020].

numpy.vstack — NumPy v1.19 Manual. 2020. numpy.vstack — NumPy v1.19 Manual. [ONLINE] Available at: https://numpy.org/doc/stable/reference/generated/numpy.vstack.html. [Accessed 23 July 2020].

Online LaTeX Equation Editor - create, integrate and download. 2020. Online LaTeX Equation Editor - create, integrate and download. [ONLINE] Available at: https://www.codecogs.com/latex/eqneditor.php. [Accessed 25 July 2020].

Python Numpy Tutorial (with Jupyter and Colab). 2020. Python Numpy Tutorial (with Jupyter and Colab). [ONLINE] Available at: https://cs231n.github.io/python-numpy-tutorial/. [Accessed 23 July 2020].

Stack Overflow. 2020. python - Get the cartesian product of a series of lists? - Stack Overflow. [ONLINE] Available at: https://stackoverflow.com/questions/533905/get-the-cartesian-product-of-a-series-of-lists. [Accessed 23 July 2020].

Steve Sque - Symbols in LaTeX and HTML. 2020. Steve Sque - Symbols in LaTeX and HTML. [ONLINE] Available at: https://www.stevesque.com/symbols/. [Accessed 25 July 2020].

Appendix

Program Source Code

The program source code is on the next page.

```python
# ------------------------------------------------------------------------------ #
#                            Presbyterian Ladies' College                        #
#                              Mentor: Dr. David Treeby                          #
#                                 Liah Wu, Yunaa Tae                             #
# ------------------------------------------------------------------------------ #
from itertools import product
import numpy as np


# ------------------------------------------------------------------------------ #
#                           Find Total Possibilities Function                    #
# ------------------------------------------------------------------------------ #


def total_possibilities(n, m, k):
    # set colours
    letter_num = ["a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m", "n", "o", "p", "q"
                  "u", "v", "w", "x", "y", "z"]

    # set lists
    poss_colours = []
    row_possibilities = []
    result_list = []

    # set possible colours for combinations
    for i in range(k):
        poss_colours.append(letter_num[i])

    # when m > 1
    if m == 1 and n == 1:
        # print answer for m = 1
        for colour in poss_colours:
            result_list.append([colour])

    else:
        # find all combinations for a row
        combinations = list(product(poss_colours, repeat=n))

        # eliminate options that do not fit with question
        left = 0
        right = 1
        for option in combinations:
            allow = True
            while right < n:
                if option[left] == option[right]:
                    allow = False
                    break
                left += 1
                right += 1
            if allow:
                row_possibilities.append(option)
            left = 0
            right = 1

        # Find allowable tilings
        valid = True
        final_list = list(product(row_possibilities, repeat=m))
        total_final_list = len(final_list)
        current_group = 0
        if m == 1:
            result_list = final_list
        else:
            for next_group in range(total_final_list):
                for row in range(m):
                    for col in range(n):
                        # array of previous row
                        case_array = np.array(final_list[next_group])
                        # swaps vertical to become horizontal
                        prepare_array = case_array[:, col]
                        case_list = prepare_array.tolist()
                        no_of_cases = len(case_list)
```

```
72                      for j in range(no_of_cases):
73                          if j < no_of_cases - 1:
74                              # check if the colours are the same or not (horizontal in the list)
75                              if case_list[j] == case_list[j + 1]:
76                                  valid = False
77                                  break
78                  if current_group < next_group and valid:
79                      result_list.append(final_list[next_group])
80                  current_group = next_group
81                  valid = True
82      return result_list
83
84  # ------------------------------------------------------------------------------------------ #
85  #                                 Display Results Function                                   #
86  # ------------------------------------------------------------------------------------------ #
87
88
89  def display_results(possibilities):
90      length = len(possibilities)
91      print(f"Total: {length}")
92      for i in range(length):
93          write_result = str(np.vstack(possibilities[i])).replace("[", "").replace("]", "").replace("
94          print(f"{write_result}\n")
95
96
97  # ------------------------------------------------------------------------------------------ #
98  #                                       Main Program                                         #
99  #                               Ask for input and print out answer                           #
100 # ------------------------------------------------------------------------------------------ #
101
102 if __name__ == "__main__":
103     # ask for input for m, n and k
104     print("------------INPUT------------")
105     print("Please enter n, m and k, where n, m > 0, and k >= 2")
106     n = int(input("n: "))
107     while n < 1:
108         n = int(input("n: "))
109     m = int(input("m: "))
110     while m < 1:
111         m = int(input("m: "))
112     k = int(input("k: "))
113     while k < 2:
114         k = int(input("k: "))
115     print()
116     print("------------OUTPUT------------")
117     # print output
118     display_results(total_possibilities(n, m, k))
```

QUESTION 1

The Best Die

By Jordan Stirzaker and Ravon Chew

Abstract

A dice game is played. Two players take turns selecting one of five eight-sided dice, where player 2 is able to see the dice player 1 has chosen. After both players roll their dice, the player who rolls the highest value wins, while both players lose if they draw.

This game is an example of a complete information game. Both player 1 and player 2 play with the knowledge of the other's motive, and both know player 2 will have access to player 1's choice. Therefore the outcome of this game is more complex than a simple game of chance.

This report investigates which die is the best for either player to choose, and whether the addition of two faces to each die changes this strategy.

Because of the small and finite number of possible permutations, all arrangements of the two dice were modelled using a spreadsheet. Both the probability of winning in each scenario, and the probability of a draw were calculated. It was determined that player 1's ideal choice is dice 1 while player 2's ideal choices are dice 3 or 5 in response to player 1's choice of dice 1. A table of ideal responses to each of the five dice for player 2 was also identified.

Table of Contents

Problem Statement **1**
Assumptions 2
Analysis **2**
Part A 2
Part B 5
Conclusion **6**
Appendix **6**
Part A 6
Part B 9

Problem Statement

For a 2 player game, there are 5 eight-sided dice on a table, all with different numerical values:

1. Die 1 - 4, 4, 4, 4, 4, 6, 6, 7.
2. Die 2 - 3, 3, 3, 3, 5, 5, 8, 8.
3. Die 3 - 1, 2, 2, 2, 8, 8, 8, 8.
4. Die 4 - 1, 1, 3, 6, 6, 6, 6, 7.
5. Die 5 - 0, 0, 5, 5, 5, 5, 6, 7.

Player 1 selects any one die. Player 2, who has seen the die that player 1 has chosen, can then select any one of the remaining 4 dice. Both players roll their dice. The winner is the player who rolls the highest number. If the numbers are the same, both players lose.

(a) What is the best die to choose?

(b) If we were to add two faces to each die, making them a ten sided die, with the two new faces taking the mean∗ value of the eight sided die, would the result change?

Assumptions

Below are explicit declarations of assumptions that can be inferred from the problem statement:

1. The dice are fair, meaning each face has the same probability of being rolled.
2. Each die can only be chosen once in one game.
3. Player 1 selects their die **with** the knowledge that Player 2 knows their selection.
4. Player 2 selects their die with the knowledge of point 3.
5. Player 1 will play optimally, and pick the die which will give them the greatest chance of a win.
6. As a draw will result in both players losing, both players will seek to prevent a permutation where a draw occurs. Hence, both players will consider drawing a "losing" strategy.

Analysis

Part A

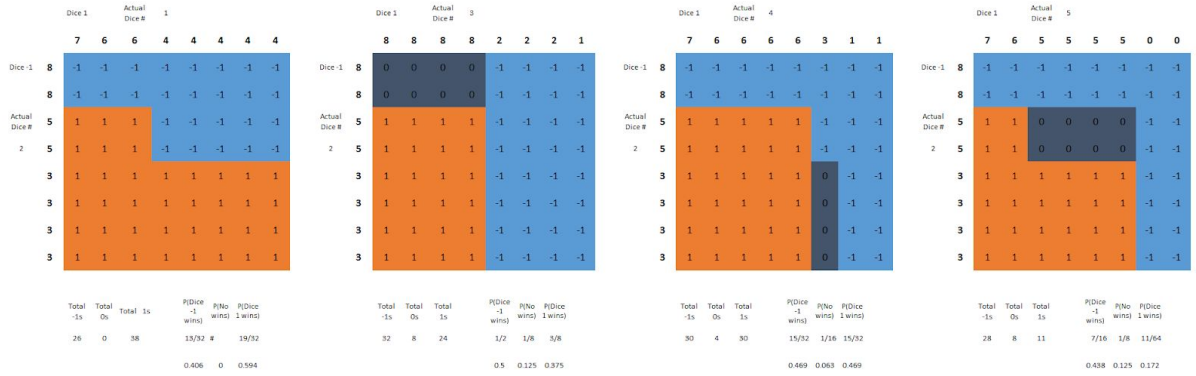
First, it was identified that a total of only ${}^5P_2 = 20$ dice permutations exist, where ${}^nP_k = \frac{n!}{(n-k)!}$.

Hence, as the sample space is not unrealistically large, it was plausible to tackle this problem through the simulation of each dice permutation.

Let the die player 1 picks be dice X, and the die player 2 picks dice Y.

For each side of dice X and each side of dice Y, the probability of dice X being greater than dice Y (and thus, a win state) was collated into one win rate table. An excel spreadsheet was used to record each result (shown in the appendix).

A sample is shown below:



Above: All possible results of dice 2 combined with dice 1, 3, 4, 5.

If dice X beats dice Y, a value of 1 was assigned to the excel square. If the reverse was true, a value of -1 was assigned, whereas a value of 0 was assigned if the two dice drew. The number of resulting -1s, 0s, and 1s were summed, and the probability of each event was calculated, as:

$$\Pr(\text{outcome}) = \frac{\text{Number of favourable outcomes}}{\text{Number of total outcomes}}$$

For example, the probability of dice 2 defeating dice 3 is 1/2, or 50%. The probability of dice 2 losing to dice 1 is 3/8 or 37.5%. The probability of a draw is 12.5%.

In accordance with the problem statement, player 1 wins if dice X beats dice Y, but loses if dice X loses to dice Y **or** there is a draw. Thus, it is only in player 1's interest to consider their chances of **winning**, and not their chances of **not losing** (as this latter scenario includes the draw).

After simulating each of the 20 possible dice permutations, the resulting win rate summary table was created:

DICE X \ DICE Y	1	2	3	4	5
1	-	0.59	0.50	0.44	0.45
2	0.41	-	0.50	0.47	0.44
3	0.50	0.38	-	0.59	0.63
4	0.42	0.47	0.38	-	0.58
5	0.50	0.44	0.38	0.34	-

Above: Probability of Dice X winning against Dice Y, with regards to dice number

In addition, the following probability table of draw rates was created:

DICE X \ DICE Y	1	2	3	4	5
1	-	0.00	0.00	0.14	0.05
2	0.00	-	0.13	0.06	0.13
3	0.00	0.13	-	0.03	0.00
4	0.14	0.06	0.03	-	0.08
5	0.05	0.13	0.00	0.08	-

Above: Probability of a draw between Dice X and Dice Y, with regards to dice number

Without consideration to game theory, it can be seen that the die with the best possible winning odds for player 1 (dice X) is dice 3, assuming it gets to play against dice 5. However, this is a large assumption that disregards turn order.

As player 2 (dice Y) is able to choose their die second, they are given the choice to inspect player 1's choice of die. After observing player 1's dice choice, it is clear that player 2 will not give player 1 this optimal match up. Instead, player 2 will naturally choose the die with the best odds against player 1's die.

Additionally, as this is a game of perfect information, it is possible for both player 1 and player 2 to play optimally. Both players have full knowledge of all the faces on all dice, as well as what the best choice for the other player is to make in response to their own choice.

Since player 1 **knows** that player 2 will choose the die that gives them the best chance of winning in response to player 1, it is best for player 1 to then choose the die that leaves them with the highest chance of winning **if** player 2 maximises their own chance of winning.

The table below shows what die (or dice if multiple dice have the same odds of winning) player 2 should choose in response to player 1's choice of die for each die in the game. The odds of both players winning and the odds of a draw are also shown.

Player 1 Die	Player 2 Die	P(Player 1 Wins)	P(Draw)	P(Player 2 Wins)
1	3	32/64	0/64	32/64
1	5	29/64	3/64	32/64
2	1	26/64	0/64	38/64
3	1	32/64	0/64	32/64
3	2	24/64	8/64	32/64
4	3	24/64	2/32	38/64
5	3	24/64	0/64	40/64

It can be seen that player 1 has the best possible chance of winning if they were to choose dice 1 or dice 3 with odds of 32/64 or 50%. This would only occur if player 2 had no motivation in making player 1 lose and only cared about winning themselves. However, player 2 could also choose dice 5 or dice 2 in response to player 1 choosing dice 1 or dice 3 respectively.

Therefore, it is safer for player 1 to choose dice 1, as it has a 29/64 chance of winning if player 2 opts to minimise player 1's chance of winning, rather than the 24/64 chance of dice 3.

Part B

In Part B, each dice gained an additional two faces, each with the value of the mean of the first eight faces. The problem booklet specified the following: “if the mean value is a decimal, round down to the nearest integer.”

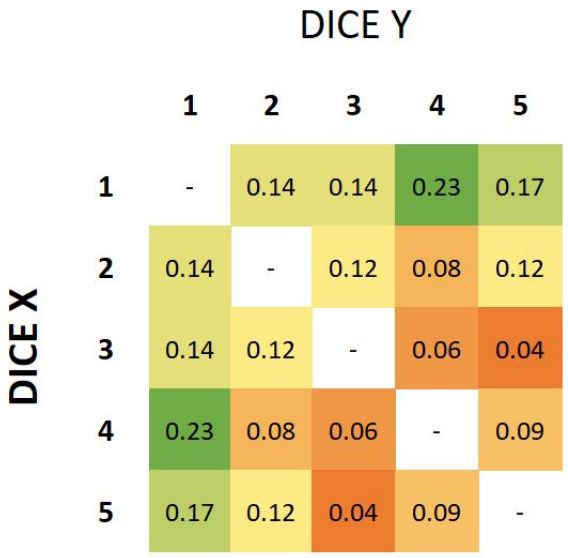
Interestingly, the floored means of all five dice was 4. Hence, an additional two sides with the value of 4 were added to all 5 dice, to form the following:

Dice 1	**7**	**6**	**6**	**4**	**4**	**4**	**4**	**4**	**4**	**4**
Dice 2	**8**	**8**	**5**	**5**	**4**	**4**	**3**	**3**	**3**	**3**
Dice 3	**8**	**8**	**8**	**8**	**4**	**4**	**2**	**2**	**2**	**1**
Dice 4	**7**	**6**	**6**	**6**	**6**	**4**	**4**	**3**	**1**	**1**
Dice 5	**7**	**6**	**5**	**5**	**5**	**5**	**4**	**4**	**0**	**0**

Similarly, the win rates of all individual dice were calculated via Excel spreadsheet.

DICE X \ DICE Y	1	2	3	4	5
1	-	0.52	0.46	0.40	0.39
2	0.34	-	0.48	0.44	0.40
3	0.40	0.40	-	0.52	0.52
4	0.37	0.48	0.42	-	0.51
5	0.44	0.48	0.44	0.40	-

Above: Probability of Dice X winning against Dice Y, with regards to dice number



Above: Probability of a draw between Dice X and Dice Y, with regards to dice number

The win rates were not that different. However, the draw rates of each dice were significantly different. This was to be expected, as the values added to each die all had the same values, and each addition made up ⅛ of each dice's total number of sides.

Another table was constructed, showing all the possible die that player 1 could choose and the optimal die for player 2 to choose in response:

Player 1 Die	**Player 2 Die**	**P(Player 1 Wins)**	**P(Draw)**	**P(Player 2 Wins)**
1	5	39/100	17/100	44/100
2	1	34/100	14/100	52/100
3	2	40/100	12/100	48/100
4	3	42/100	6/100	52/100
5	3	44/100	4/100	52/100

It can be seen clearly from the table that the best choice for player 1 is to choose dice 5. This gives them a 44/100, or 44% chance of winning if player 2 chooses dice 3 in response. If player 1 indeed chooses dice 5, this gives player 2 a 52/100 or 52% chance of winning, which is also the best possible chance of winning for them.

Interestingly, if player 1 plays suboptimally, player 2 likely has a lower chance of winning. This is due to the higher probability of a draw compared to part A, in which neither player wins.

Conclusion

Based on individual case analysis of the possible permutations and win rates, and assuming that both players play optimally, player 1's ideal dice choice in part A is dice 1 (29/64 or 32/64 chance of winning). On the other hand, player 2's ideal dice choices in part A are dice 3 and 5 in response to player 1's choice, giving them a 32/64 chance of winning.

In part B, player 1 is instead most likely to win if they choose dice 5 (with a 44% chance of winning), compared to player 2's optimal choice of dice 3 (52% chance of winning) in response.

Appendix

Part A

Dice 1	7	6	6	4	4	4	4	4
Dice 2	8	8	5	5	3	3	3	3
Dice 3	8	8	8	8	2	2	2	1
Dice 4	7	6	6	6	6	3	1	1
Dice 5	7	6	5	5	5	5	0	0

Figure 0: List of dice

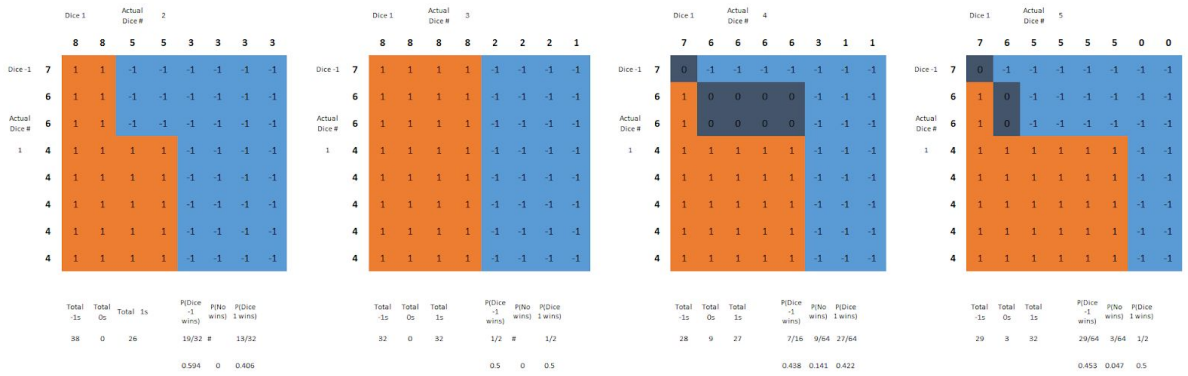


Figure 1: Win rates of Dice 1

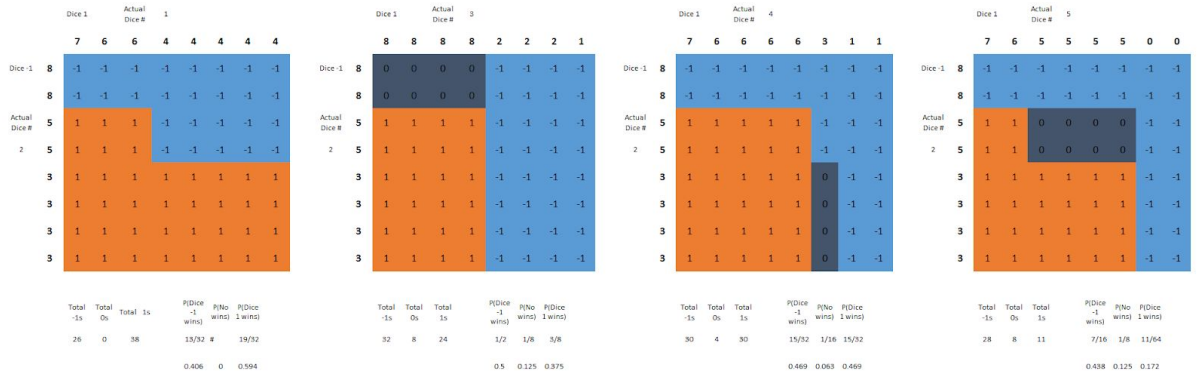


Figure 2: Win rates of Dice 2

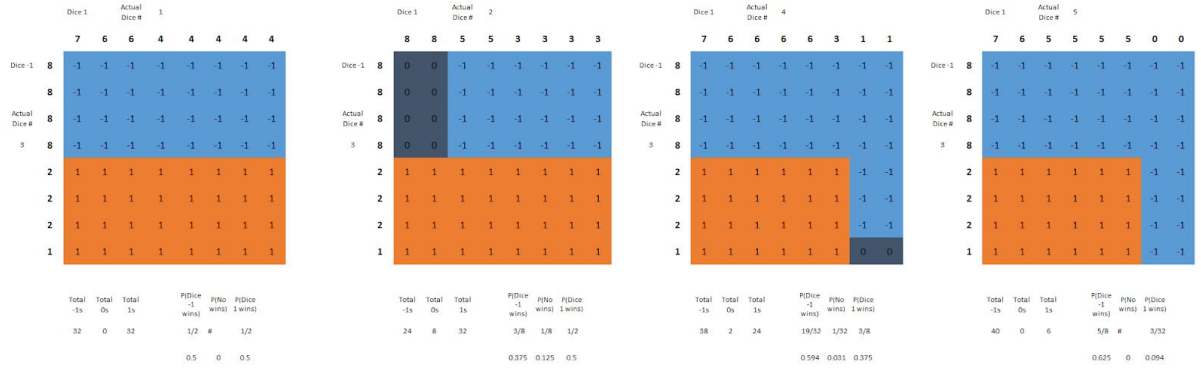


Figure 3: Win rates of dice 3

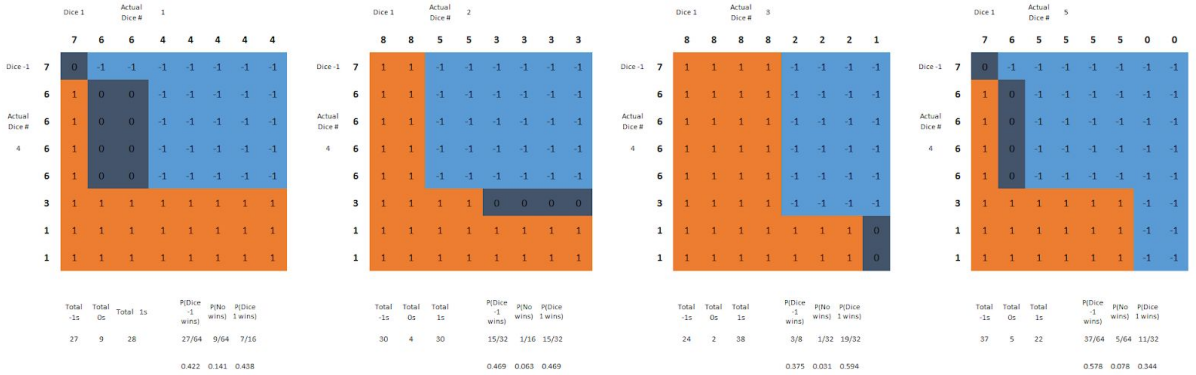


Figure 4: Win rates of dice 4

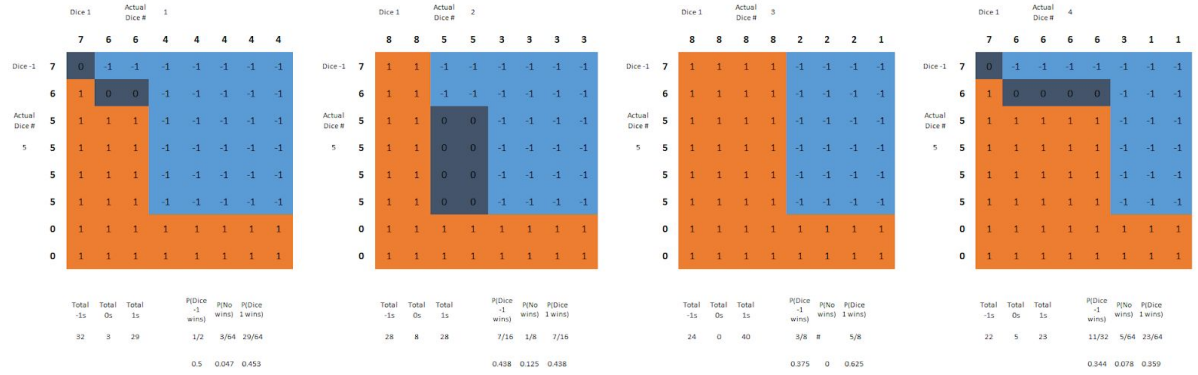


Figure 5: Win rates of dice 5

Figure 6: Win rates of Dice X against Dice Y

Figure 7: Draw rates of Dice X against Dice Y

Part B

Dice 1	7	6	6	4	4	4	4	4	4	4
Dice 2	8	8	5	5	4	4	3	3	3	3
Dice 3	8	8	8	8	4	4	2	2	2	1
Dice 4	7	6	6	6	6	4	4	3	1	1
Dice 5	7	6	5	5	5	5	4	4	0	0

Figure 0: List of dice

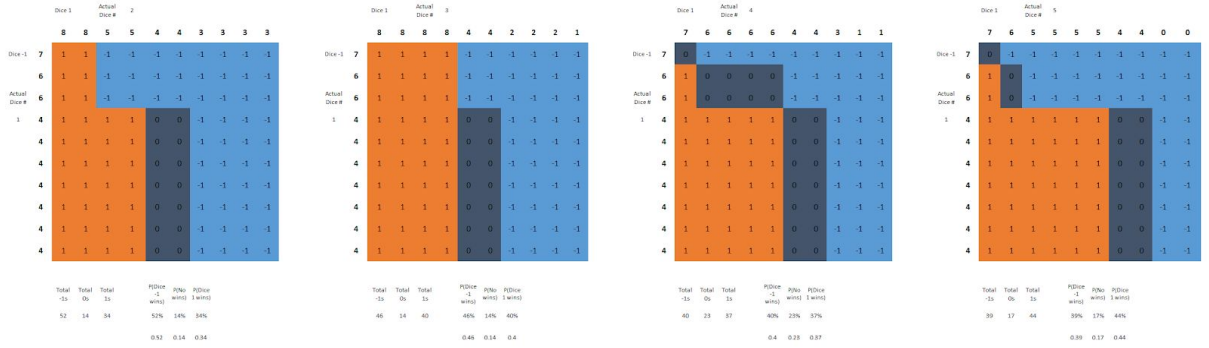


Figure 1: Win rates of dice 1

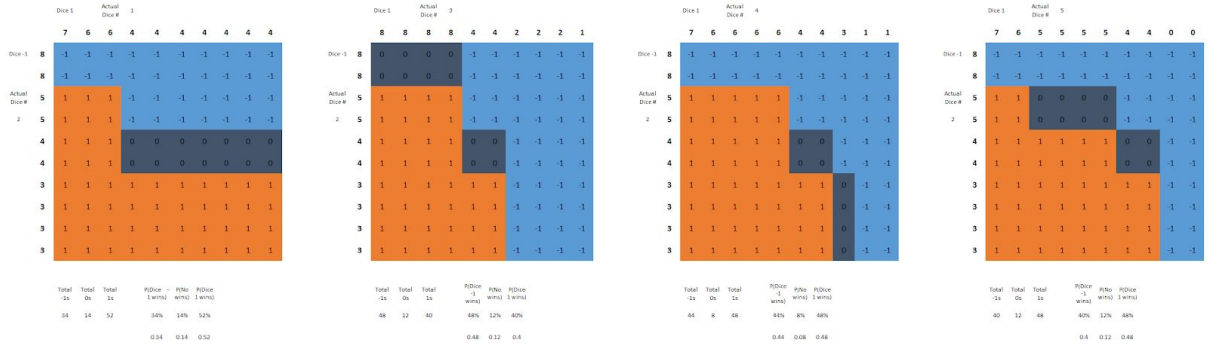


Figure 2: Win rates of dice 2

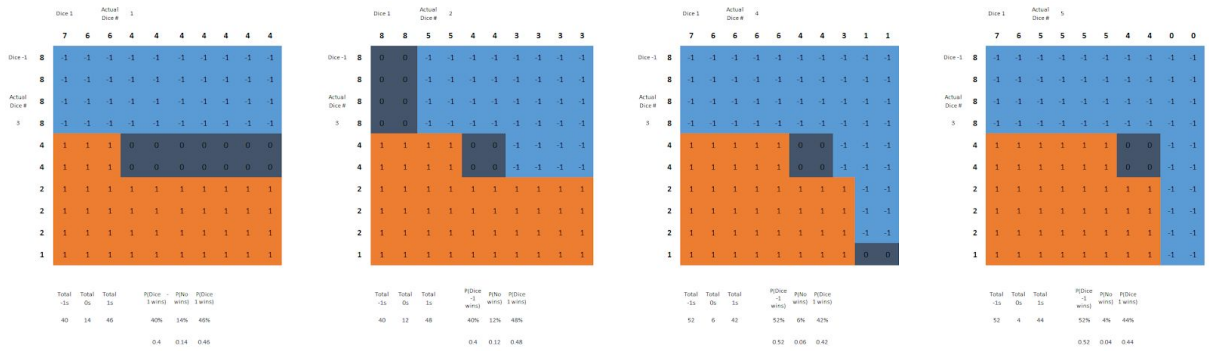


Figure 3: Win rates of dice 3

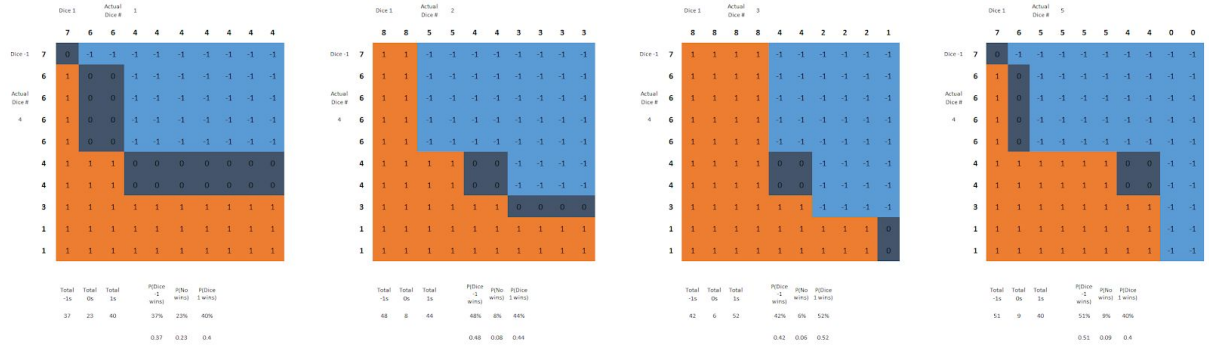


Figure 4: Win rates of dice 4

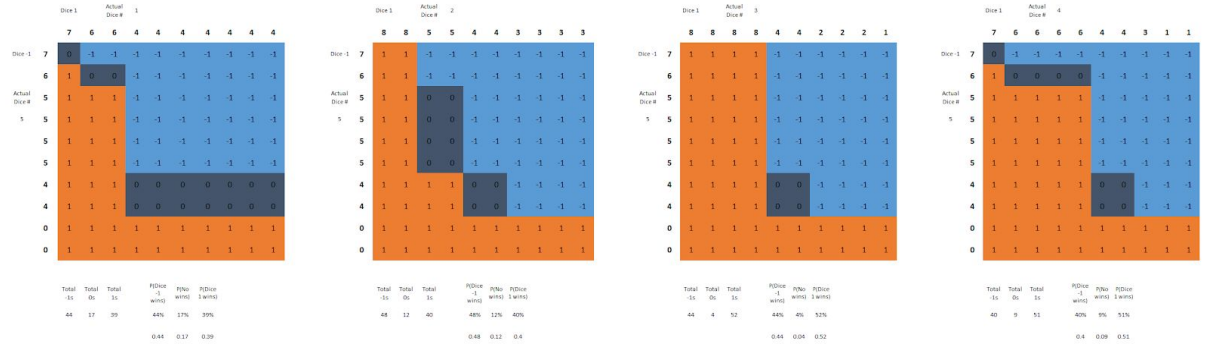


Figure 5: Win rates of dice 5

Figure 6: Win rates of Dice X against Dice Y

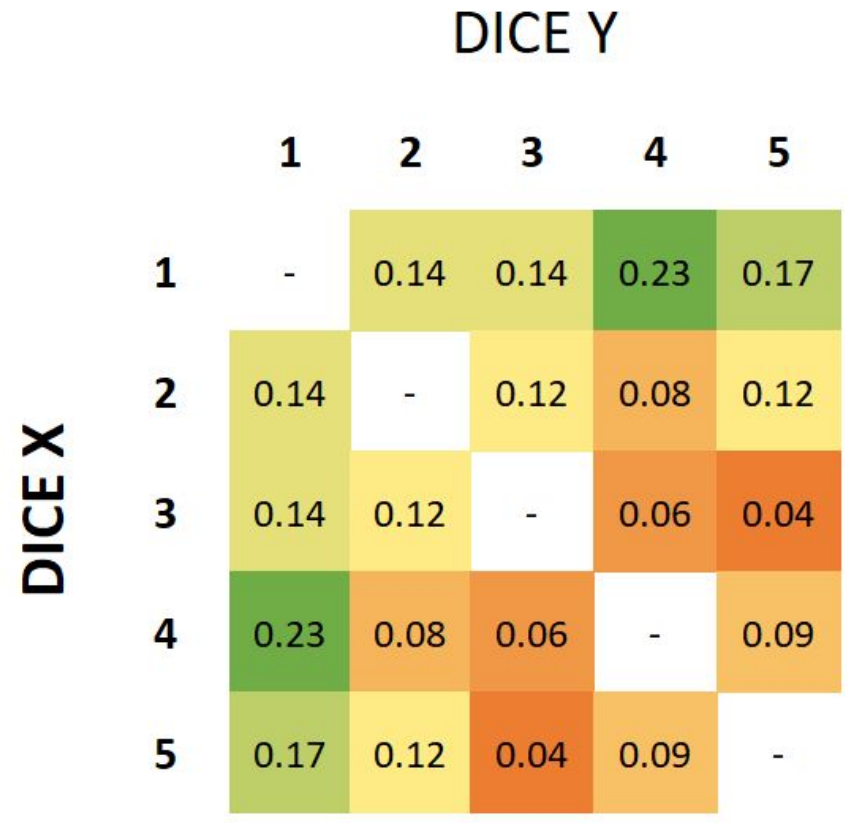


Figure 7: Draw rates of Dice X against Dice Y

QUESTION 6

Modelling Life Expectancy

By Jordan Stirzaker, Ravon Chew and Mohit Singh

Abstract

Life expectancy is an important indicator for a population's health. The current standard for life expectancy - life expectancy at birth - considers only a narrow range of factors and is centered around mortality. Thus, it is important to investigate other life expectancy models, which may be able to provide a better metric.

First, the initial problem scope was broadened from the life expectancy of an individual to the life expectancy of a country. In preparation for building a model, several parameters theorised to influence life expectancy were investigated. The associated data sets of each parameter were sourced, and the R-Square value of each parameter was calculated using the Matlab Curve Fitting Toolbox.

Out of the R-Squared values, the 5 best fitting parameters were chosen, and fitted onto a polynomial model using Matlab. Two models were created - one using a smaller data set from Gapminder and another using a larger data set from the World Bank. The second model's data was sorted and organised using a custom python program.

Model 2 is as shown:

$$\text{Life Expectancy} = 48.1606 + 16.6802A + 0.3208B + 0.0568C + 0.6301D + 9.0513E$$

The two models were tested against new sets of data in order to determine validity. The median deviance of predicted versus observed life expectancies of Model 1 was 3.8%, while the median deviance of Model 2 was 4.8%. Although the two differ in accuracies across a relatively accurate error rate.

The model identifies which aspects of a country contribute to the average citizen life expectancy, as well as whether the impact is positive or negative. Thus, the model provides a greater range of applications when compared to life expectancy at birth. The model can be used by countries to predict future growth patterns and determine areas of concern.

However, the model also contains some flaws. Inexperience with high-level statistic programs such as Matlab contributed to issues with creating an exponential model. The

availability of accurate and consistent data was an issue, as well as a consideration of only a limited number of parameters.

Ultimately, the proposed life expectancy model is a promising start but still has room for improvement.

Table of Contents

1.0 Problem Statement **1**

2.0 Introduction **1**

2.1 Assumptions 2

2.2 Statistical Terminology 2

3.0 Analysis **3**

3.1 Data Organisation 4

3.2 Data Fitting 4

3.3 Creating the Model 6

3.4 Model 1 7

3.41 Testing of Validity 8

3.42 Evaluation 10

3.5 Model 2 10

3.51 Testing of Validity 11

3.52 Evaluation 11

4.0 Discussion **11**

4.1 Strengths 11

4.2 Weaknesses 12

4.3 Further Study 12

6.0 Appendix **14**

7.0 References **21**

1.0 Problem Statement

Life expectancy from birth is a frequently utilized and analysed component of demographic data for the countries of the world. It represents the average lifespan of a newborn and is an indicator of the overall health of a country. Use real data, freely available from sources such as Gapminder, to investigate and model how to predict life expectancy.

2.0 Introduction

Life expectancy is a statistical measure of the average time an individual is expected to live.

Life expectancy can be a useful tool to indicate patterns that span a community, region, or country, and can help point communities towards factors they need to improve. Currently, the most common metric for life expectancy is life expectancy from birth (LEB). LEB is typically calculated by analysing and normalising the ace-specific death rates of a population.

There are many benefits of adopting the LEB system as an indicator of population health. The calculation methods behind LEB are generally non-controversial and well understood. Death registration data is also widely available for most countries and regularly updated.

However, there are also many disadvantages of the LEB system.

Calculations of LEB are often restrictive. Generally, cohort data is incomplete, as not all newborns born in the same year have completed their full livespans. In addition, LEB is calculated from mortality rates - are a product of multiple variable factors a person experiences throughout their lives.

A crucial distinction so between an average and a probability. LEB is a form of statistical average, and is not a metric that can be used to *predict* life expectancy patterns in the future. According to ONS data, the average life expectancy for a 65 year old man is 86, but he has a 1 in 4 chance of living to 94. For a 65 year old woman the average life expectancy is 89, but she has a 1 in 4 chance of living to 962. However, predicting the age that someone is going to live to is not quite as simple as that.

A different approach to the life expectancy problem concerns the mechanistic perspective. An alternate interpretation of the problem concerns: How can a country predict its own life expectancy based on measurable and controllable factors?

This approach broadens the scope from the individual to the country. If a holistic model were to be developed for a country, then the country would be able to analyse which areas funding should be directed towards, in order to ensure life expectancy growth.

2.1 Assumptions

- Life expectancy calculations found on gapminder are accurate
- Data found on gapminder is unbiased and reliable
- Data reported by countries is unbiased and reliable
- Each country is weighted equally

2.2 Statistical Terminology

- **SSE** is the **sum of squares due to error**. It is a measure of the total deviation of response values to the fit. The formula for SSE is:

$$\sum_{i=1}^{n}(X_i - \overline{X})^2$$

Where:

$X_i =$ the i^{th} observed value

$\overline{X} =$ the mean of all observed values

- The **R-Squared** value is the square of the variation between observed and predicted values. It measures the proportion of variance for dependent variables as explained by the independent variables. R-Squared is used to quantify how successful the fit is in explaining the variation of the data. The formula for R-Squared is:

$$R^2 = 1 - \frac{SS_{Regression}}{SS_{Total}}$$

$$= 1 - \frac{\sum(X_i - \hat{X}_i)^2}{\sum(X_i - \overline{X})^2}$$

Where:

$X_i =$ the i^{th} observed value

$\hat{X}_i =$ the i^{th} predicted value

$\overline{X} =$ the mean of all observed values

- The **Adjusted R-Squared** modifies the R-squared value by the residual degrees of freedom, which can be found by subtracting the number of estimated fitted coefficients m from the number of response values n. It compensates for the addition of variables by subtracting those that do not contribute towards predicting the dependent variable. The formula for Adjusted R-Squared is:

$$\bar{R}^2 = 1 - \frac{\sum(X_i - \hat{X}_i)^2/(n-p-1)}{\sum(X_i - \bar{X})^2/(n-1)}$$

$$= 1 - \frac{(1 - R^2)(n - 1)}{(n - p - 1)}$$

Where:

R^2 = Sample R-Square

$p =$ number of predictors

$n =$ total sample size

- **RMSE** refers to the **Root Mean Square Error**, and is also known as the standard error of the regression. It is an estimate of the standard deviation of the random component in the data, through the normalised distance between predicted and observed values. The closer the RMSE is to 0, the better of a fit. The formula for RMSE is:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (\hat{X}_i - X_i)^2}{n}}$$

Where:

$X_i =$ the i^{th} observed value

$\hat{X}_i =$ the i^{th} predicted value

3.0 Analysis

3.1 Data Organisation

First, factors that are likely to influence life expectancy were determined. Due to time restraints, the choices of which parameters to test were largely based on intuition. This allowed the team to narrow the field of search, and invest more time towards creating mechanisms to fit the data.

Possible factors that could affect life expectancy that were considered include:

- Alcohol Consumption (per capita in litres)
- Basic Sanitation (% of population)
- BMI Average
- Government Health Spending (USD$)
- Government Health Spending (% of budget)
- Doctors per 1000 People
- GDP per Capita (USD$, inflation-adjusted)
- Gini Coefficient (measure of inequality, higher means more inequality)
- Primary School Completion Rate (% of population)

Next, data from the aforementioned categories were downloaded from the website ‘gapminder.com’. Care was taken in order to ensure the factors covered a range of different areas. However, most of the factors identified were related to health and economics.

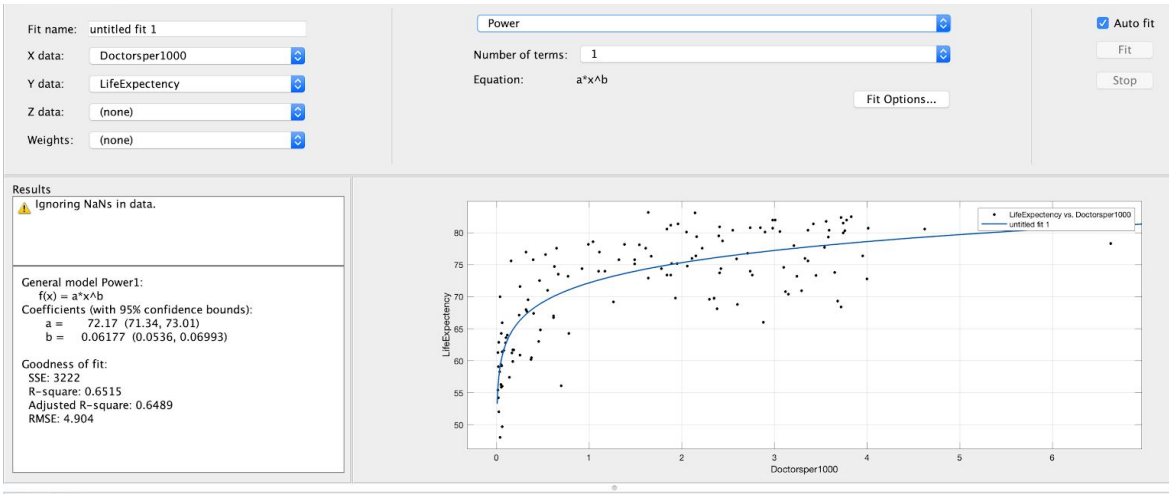
In total, Gapminder provided the data of over 180 countries, encompassing a range of years from 1800 to the present day. However, gaps in the data would sometimes arise - either because the country in question did not exist before a certain time period, or because no data for that category had been collected.

Hence, all data we used was taken from 2008, which was the most recent year that provided data for all of the target parameters. Consistency in the selection of data was important - i.e. to take data all from the same year -, as data from different timeframes can not be compared effectively. Ultimately, a tradeoff between consistency of data and lacking data from the most recent years was made, instead collecting complete information from a single year and several countries.

3.2 Data Fitting

In the Data Fitting stage, each identified parameter was fitted individually against life expectancy, testing for goodness of fit. This process was achieved through the Matlab Curve Fitting Toolbox, which uses regression analysis to fit the data to a variety of models, providing optimisation of fitting. Both linear and exponential models were used, depending on which provided a more accurate fit.

A sample of the resulting fit is as shown:



Above: Doctors per 1000 against Life Expectancy (fig.4)

Basic sanitation and primary school completion both had a linear line of best fit when compared to the life expectancy data individually while GDP per capita, government health spending per capita and doctors per thousand all had exponential lines of best fit.

The program was then able to test for goodness of fit, represented in the forms of SSE, R-Squared, Adjusted R-Squared, and RMSE. Although all the aforementioned statistical tests were valid, the R-Squared Value was chosen as it was the most familiar to the

team. Additionally, adjusted R-Squared could have been used, but were deemed largely unnecessary, as all the models used only had one term.

The R-Squared value for each tested parameter are shown below:

Tested Parameter	R-Squared Value
Alcohol Consumption (per capita in litres)	0.07
Basic Sanitation (% of population)	**0.70**
BMI Average	0.21
Government Health Spending (USD$)	**0.53**
Government Health Spending (% of budget)	0.06
Doctors per 1000 People	**0.65**
GDP per Capita (USD$, inflation-adjusted)	**0.43**
Gini Coefficient (a measure of inequality, higher means more inequality)	0.17
Primary School Completion Rate (% of population)	**0.53**

Note: Bolded factors were chosen for the final model.

The closer the R-Squared value was to 1, the greater the case for causality. Based on the R-Squared value, the 5 best parameters were chosen to be included in the final model.

Each of the 5 chosen parameters had R-Squared values of at least 0.43. This meant that all parameters chosen had a Pearson's Correlation Coefficient (square root of R-Squared) of at least 0.65, which is indicative of a fairly significant correlation. Although the threshold of 5 parameters was chosen somewhat arbitrarily, the analysis of Pearson's correlation coefficient demonstrated each had a notable level of correlation. Intuitively one would also expect the chosen parameters to have some correlation with life expectancy.

This was also necessary, as life expectancy is difficult to model due to influence from many different factors (more than those that can be feasibly tested), all of which would affect life expectancy differently and to various degrees.

3.3 Creating the Model

In accordance with the previous step, 5 factors that were determined to have a significant enough correlation towards life expectancy were identified. The sets of data that corresponded to the 5 chosen parameters were organised, removing countries that lacked sufficient data for all 5 parameters. After reorganisation of the data, a total of 89 countries with sufficient data were left.

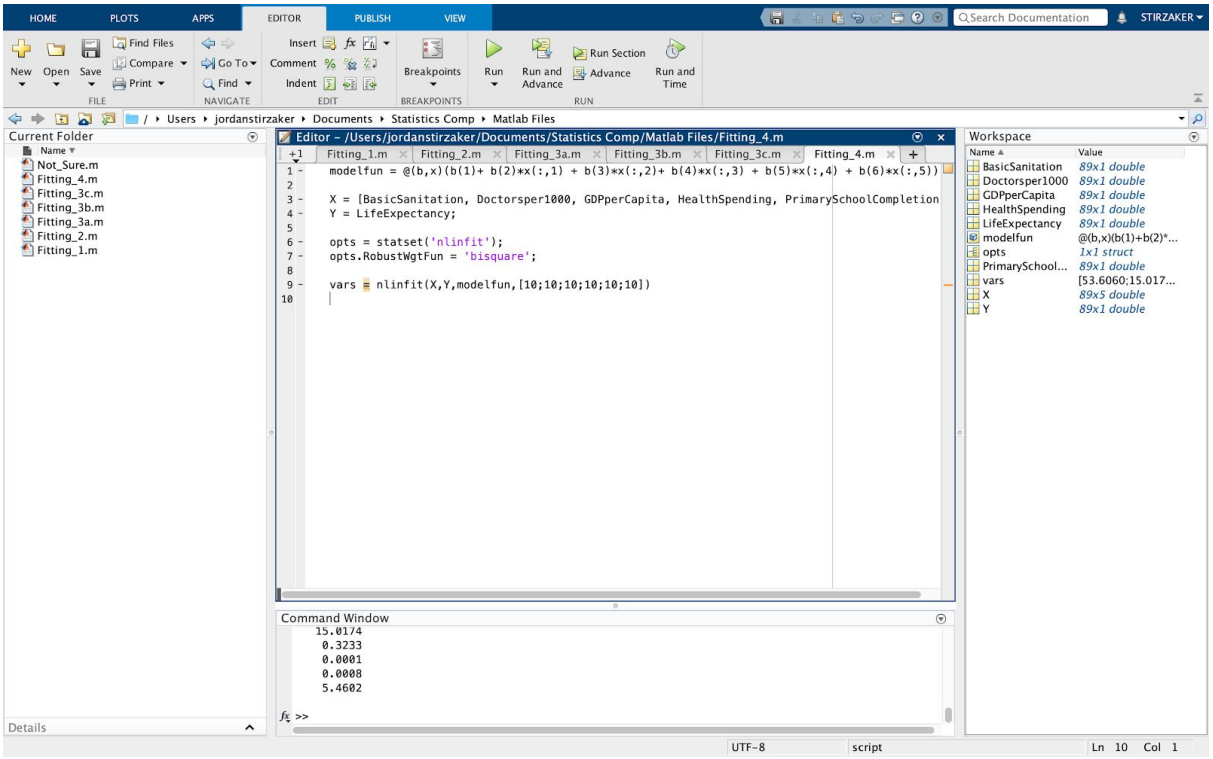
Next, Matlab was again used in an attempt to create a model using all 5 sets of data.

Unfortunately, attempting to fit the data using a combination of both linear and exponential factors did not succeed. If there was more than one exponential factor in the model, Matlab tended to return strange results. When starting values returned an actual result, that result usually ended up including imaginary numbers or negative coefficients to a parameter that had strong positive correlation when tested independently.

With limited data fitting knowledge, the team were unable to find a solution to this issue, and instead decided to only include linear factors in the model. Although this was not ideal and could result in a poorer fit to the experimental data, when later tested, the model turned out to predict life expectancy relatively accurately.

To create the model, the nlinfit function on Matlab was used, which returns estimated coefficients for a nonlinear regression model. In particular, it returns the least squares parameter estimates. Several different combinations of the factors, as well as multiple varieties of polynomial expressions were tried. Eventually, they were evaluated against each other, and the most suitable fit was chosen.

A sample of the code used is as shown below:



3.4 Model 1

Using the method outlined above, the first model was created:

$$\text{Life Expectancy} = 53.6060 + 15.0174A + 0.3233B + 0.0532C + 0.8430D + 5.4602E$$

Where:

A = Percentage of Population with Basic Sanitation (from 0 to 1)
B = Doctors per 1000 People
C = GDP per Capita (USD$)
D = Government Health Spending per Capita (USD$)
E = Percentage Primary School Completion Rate (from 0 to 1)

3.41 Testing of Validity

Like many statistical models, the accuracy of the predictions from the model can not be fully confirmed until its predicted events have occurred. However, several initial tests of validity can be performed.

One such test is to use the model to predict the life expectancy of a country that was not used in the initial regression. For example, the difference between predicted and observed life expectancies for Australia was calculated. Australia was not included in

the initial data, as it lacked data for primary school completion rate. However, the primary school completion rate can be estimated from countries with similar cultures and GDPs to Australia. The most recent available data about Australia was sourced:

$$A = 1,\ B = 3.7,\ C = 55.306,\ D = 3.272,\ E = 0.995 \textit{ (estimated)}$$

$$\begin{aligned}\text{Life Expectancy} &= 53.6060 + 15.0174 * 1 + 0.3233 * 3.7 + 0.0532 * 55.306 + \\ &\quad 0.8430 * 3.272 + 5.4602 * 0.995 \\ &\approx 80.953\end{aligned}$$

Data from the World Bank states that the average Australian life expectancy in 2018 was 82.749 years, differing from the model's prediction by only 1.796 years or 2.22%.

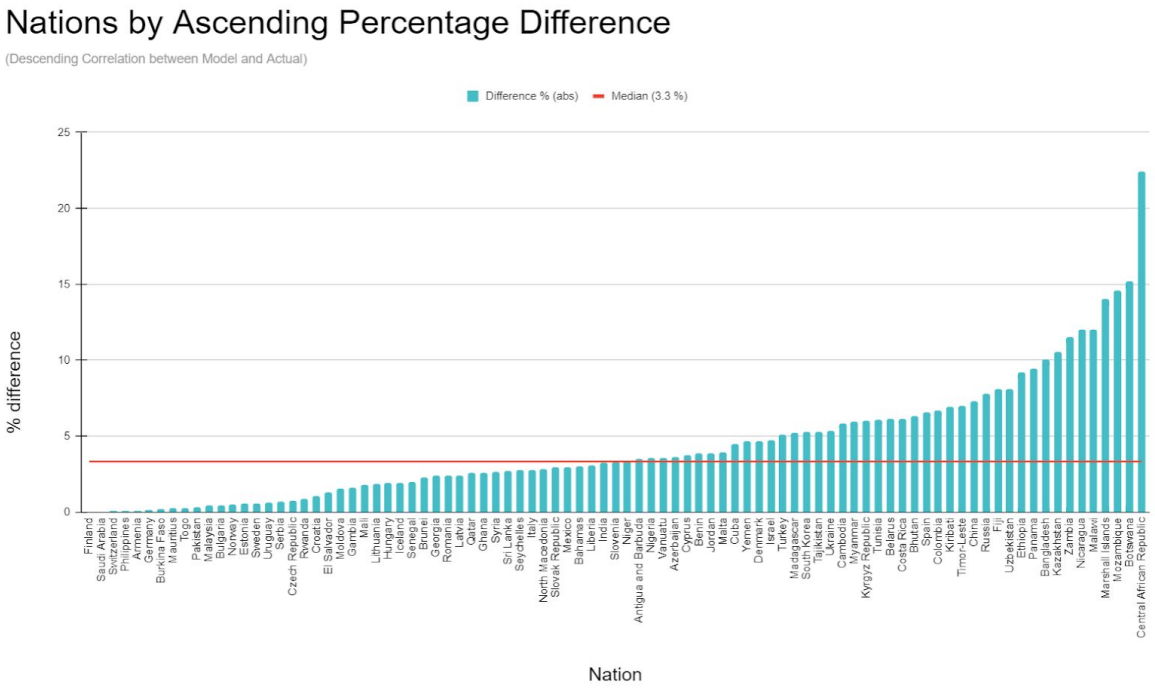
Furthermore, the predicted life expectancies from the model were compared against actual observed life expectancy data. This was done in order to ensure the predicted life expectancies as created from the model were within an acceptable range of error.

For each, the difference in terms of years and difference in terms of percent was calculated. The results were formatted in a spreadsheet and colour coded.

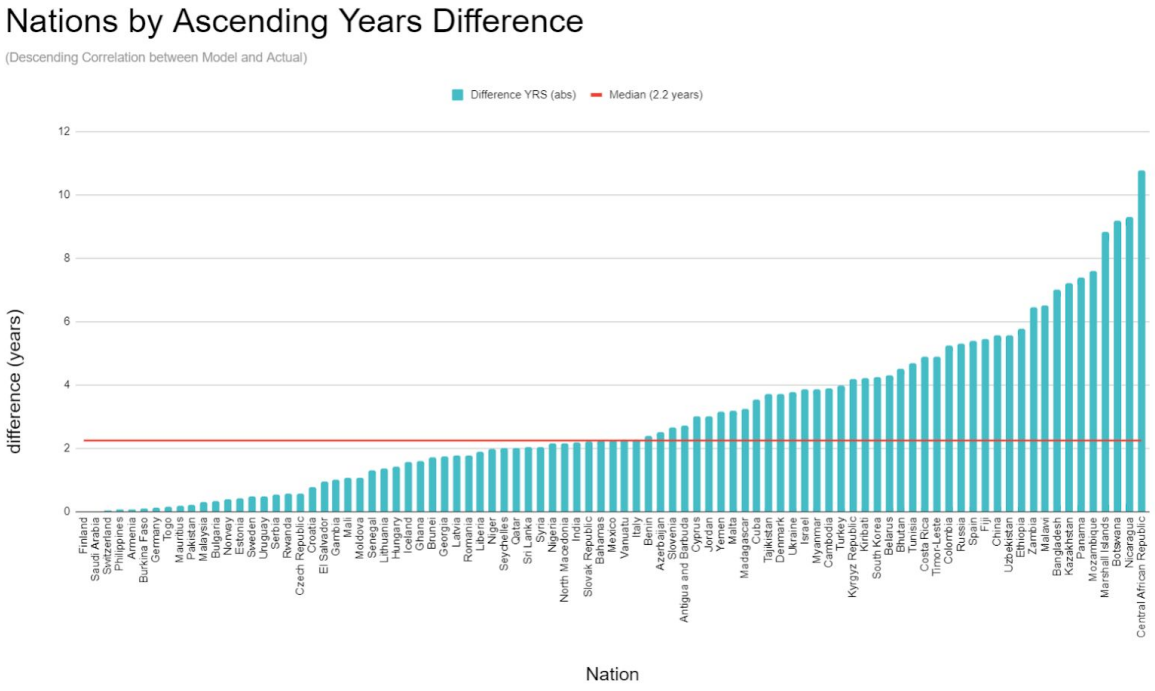
Nation	Life Expectancy	Basic Sanitation	Doctors per 1000	GDP per Capita	Health Spending	Primary School Completion	Expected Life Expectancy	Difference YRS	Difference %
Switzerland	82.5	0.999	3.83	75800	4160	0.944	85.9030097	3.4030097	4.124860242
Iceland	82.4	0.988	3.72	47900	4120	0.948	82.9208424	0.5208424	0.632092913
Italy	82	0.988	3.78	37700	2690	1.06	81.6155476	0.3844524	0.468843902
Spain	82	0.999	2.98	32100	31.7	0.973	78.1507276	3.8492724	4.694234634
Israel	81.8	1	3.56	30000	1250	1.01	79.408285	2.391715	2.923856968
Sweden	81.5	0.993	3.74	52800	3970	0.943	83.3293784	1.8293784	2.244636074
Norway	80.9	0.981	2.41	19200	6800	0.992	81.9703659	1.0703659	1.323072806
South Korea	80.6	1	1.84	20800	692	1	77.366976	3.233024	4.01119603
Germany	80.4	0.992	3.59	42100	3610	1.02	82.4812194	2.0812194	2.588581343
Malta	80.4	1	3.36	21200	1040	1.01	78.290505	2.109495	2.62375
Finland	80.2	0.994	3.06	49400	3170	0.988	82.4825226	2.2825226	2.821100499
Cyprus	80.1	0.996	2.05	32700	796	1	78.6582782	1.4417218	1.799902372
Slovenia	79.5	0.991	2.4	16800	1660	0.97	77.5882317	1.9117683	2.40474
Costa Rica	79.4	0.951	2.16	8030	413	0.945	74.8912252	4.5087748	5.678557683
Denmark	79.3	0.996	3.58	60500	5390	1.01	85.7220962	6.4220962	8.098481967
Cuba	78.3	0.893	6.63	5510	557	0.89	75.1026841	3.1973159	4.083417497
Panama	78.2	0.715	1.38	7810	341	0.977	71.5240052	6.6759948	8.537077749
Colombia	78.1	0.802	1.54	6120	266	1.13	73.6951624	4.4048376	5.639996927
Qatar	78	1	3.21	65000	1300	1.06	83.200175	5.200175	6.666891026
Czech Republic	77.7	0.991	3.54	20500	1220	0.985	78.1161792	0.4161792	0.535623166
Jordan	77.6	0.98	2.22	3790	204	0.947	74.7358787	2.8641213	3.690877964
Turkey	77.6	0.901	1.61	3950	455	0.987	73.9495094	3.6504906	4.704240484
Nicaragua	77.6	0.664	0.649	1530	57.7	0.802	68.4019251	9.1980749	11.8531893
Tunisia	77	0.843	1.11	10600	133	0.922	72.8963343	4.1036657	5.329435974
Croatia	76.8	0.963	2.71	15200	1040	0.987	76.7813848	0.0186152	0.02423854167
Antigua and Ba	76.6	0.855	0.527	16600	409	1.11	75.0854848	1.5145152	1.97717389
Uruguay	76.4	0.948	3.95	10700	460	1.07	76.7139936	0.3139936	0.4109863874
Estonia	76	0.994	3.33	16800	836	0.955	77.1826823	1.1826823	1.556160921
North Macedor	75.9	0.904	2.59	90900	223	0.918	82.3260796	6.4260796	8.466508037
Syria	75.8	0.928	1.49	707	41.8	1.04	74.0216026	1.7783974	2.346170712
China	75.8	0.701	1.32	3800	78.2	0.993	70.818429	4.981571	6.571993404
Slovak Republic	75.6	0.979	3.36	43200	948	0.974	79.8579957	4.2579957	5.632269444
Mexico	75.2	0.83	1.95	9590	281	1.01	73.685767	1.514233	2.013607713
Brunei	75.1	0.963	1.49	35800	754	1.05	78.6581731	3.5581731	4.737913582

Above: Unsorted difference between expected and observed life expectancies (fig. 12)

This was then sorted in ascending order and graphed.



Above: Nations by ascending percentage difference (fig. 10). The median value (red line) is a difference of 3.8%.



Above: Nations by ascending years of difference (fig. 11).

3.42 Evaluation

Despite the promising deviances of 2.2 years and 3.8%, it is important to recognise that Model 1 was trained solely on data from 2008. This meant that any extrapolations from the model could possibly not be representative of life expectancy for future years.

Therefore, a second model that would be trained on data from several years was proposed.

The first step was to source and reorganise a larger set of data. Instead of Gapminder, data (from range) was downloaded from the World Bank. A python program was then created in order to sort the set of data, which organised and filtered the original data to only include data from years where a certain country had data for all 5 needed parameters.

This was achieved by importing the data into a spreadsheet and assigning each spreadsheet cell a coordinate on a cartesian plane. The program would iterate between all six (one per parameter) graphs and save any points which existed on all six graphs. The implementation of this custom data-sorting algorithm meant that the model would train on data from years other than and including 2008.

```
out_writer = csv.writer(out_file, delimiter=',', quotechar='"', quoting=csv.QUOTE_MINIMAL)

LIFEDATA = list(CSVLIFE)
BASICDATA = list(CSVBASIC)
DOCDATA = list(CSVDOC)
GDPDATA = list(CSVGDP)
EXPENDATA = list(CSVEXPEN)
PRIMDATA = list(CSVPRIM)
out_writer.writerow(["YEAR", "NATION","ACTUAL LE", "BASIC SANITATION", "DOCTORS PER 1000", "GDP PER CAPITA", "HEALTH EXPENDITURE", "PRIMARY COMPLETION", "DIFFERENCE (YEARS)"])

for y in range(5,269):
    for x in range(4,64):
        if LIFEDATA[y][x]!="" and BASICDATA[y][x]!="" and DOCDATA[y][x]!="" and GDPDATA[y][x]!="" and EXPENDATA[y][x]!="" and PRIMDATA[y][x]!="":
            output = model(LIFEDATA[y][x],BASICDATA[y][x],DOCDATA[y][x],GDPDATA[y][x],EXPENDATA[y][x],PRIMDATA[y][x])
            out_writer.writerow([BASICDATA[4][x],BASICDATA[y][0],LIFEDATA[y][x],str(float(BASICDATA[y][x])/100),DOCDATA[y][x],str(float(GDPDATA[y][x])/1000),
                                 str(float(EXPENDATA[y][x])/1000),str(float(PRIMDATA[y][x])/100),str(output-float(LIFEDATA[y][x]))])
            a+=1
            b +=(output-float(LIFEDATA[y][x])) #/float(LIFEDATA[y][x])

            print(BASICDATA[4][x],BASICDATA[y][0],"Actual: "+str(LIFEDATA[y][x]),"Model: "+str(output),"Diff (yrs): " + str(output-float(LIFEDATA[y][x])),b/a)
```

Above: Sample of the python sorting code

3.5 Model 2

The resulting model is as follows:

$$\text{Life Expectancy} = 48.1606 + 16.6802A + 0.3208B + 0.0568C + 0.6301D + 9.0513E$$

Interestingly, the coefficients of B and C did not change significantly. D decreased while both A and E increased. This was likely as A and E both had an individual linear line of best fit, while the others were exponential.

3.51 Testing of Validity

Similar to before, the new model was tested by iterating it through a large sample of data. The differences between the predicted and observed life expectancies was recorded and used to calculate the median deviance.

3.52 Evaluation

Despite the larger sample of training data, there was no significant improvement in the accuracy of the model.

When tested against all the larger data set, the median difference of model 2 was 2.850 years and 4.330%, compared to model 1's median deviance of 2.887 years and 4.489%. Therefore, the newer model outperformed the older model by ~0.027 years in the larger dataset.

	Model 1		Model 2	
	Years	Percentage	Years	Percentage
Small Dataset	2.2	3.3%	2.761	4.067%
Large Dataset	2.887	4.489%	2.850	4.330%

However, this is not to say that model 2 is of no use.

Interestingly, it is important to consider the contexts in which the two models perform. Model 1 performed better when tested with the smaller data set, whereas model 2 performed better with the larger data set. Overall, this can determine which model to use, depending on what set of data is needed.

4.0 Discussion

For the rest of the discussion, both models will be considered together. This is as there is not a large difference between the two's performance.

4.1 Strengths

Strengths of the model include:

- The model directly relates measurable parameters to life expectancy, offering a greater level of specificity in potential applications then the conventional life expectancy at birth model.
    - In particular, the model is based upon 5 different parameters, which can account for a greater range of influences than the life expectancy at birth model.
- The model can be used to offer direction for countries' future growth, as it targets several quantifiable parameters.
- The model has been fitted and tested with a large amount of data, comprising a span of several different years and countries. Therefore, the model can be considered to be relatively accurate.

4.2 Weaknesses

Due to limitations in the modelling procedure, several flaws in the model can be identified.

A main flaw of our model was the limitations of available data. One aspect of this was the assumption that all data collected was consistent and accurate. Moreover, the model used data solely from one year (2008) and from one source (Gapminder).

Additionally, other flaws include:

- Inability to fit factors that do not correlate linearly with life expectancy. This was due to low familiarity with the MATLAB environment and language. Attempts to include factors with an exponential fit into the model resulted in a computational error - including unreasonable outputs of NaN and imaginary numbers.
- Inclusion of a limited range of data. The initial model only considered data from 2008 since it was the most recent year containing all the necessary data. This was appended in the second model, but the improvement was minimal.
- Consideration of a limited number of parameters. The choice to include only 5 factors in the final model was an arbitrary one limited by the amount of time, data, and computational power available. Ideally, all parameters available in the form of data should be tested to determine the strongest fitting factors.

4.3 Further Study

As per the weaknesses identified above, several of the limitations on the model could be improved upon if given the sufficient resources.

1) Larger variety of data

A better model would be able to train using data collected over a larger period of time. The larger training sample would help improve the robustness of the model, as well as significantly improve the model's accuracy due without there being a possibility of significant interference from 2008 specific events/oddities in certain nations.

Furthermore, data from multiple sources could be used instead of the single source used in this paper. By using multiple sources, the impact of bias in data from one source would be reduced (but not completely eliminated). This would also allow easier identification and better understanding of outliers and potential discrepancies in data collection by comparing and contrasting multiple sources.

2) Consideration of more factors

If the range of considered factors was broadened, the model would be able to allow for greater accuracy. As the number of factors increase, the closer the weighting of each factor will move compared to its real world significance. This would also solve the problematic issue of manually choosing only a few factors in the model and disregarding the rest.

However, the increase in computation power required as the number of factors increase will have to be considered. The computation power and time needed would likely increase at a rate much higher than linear - possibly quadratic, polynomial, or by a factor of n! factorial (where n is number of factors).

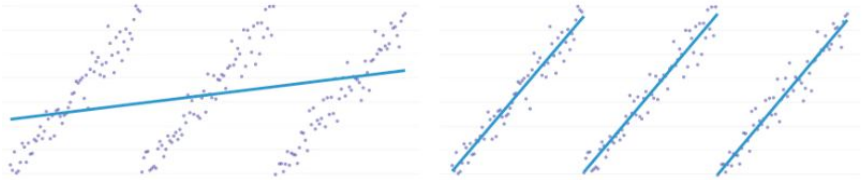
3) Increased computing power

If access to a greater arsenal of computing power was available along with a larger set of data, models with greater complexity could be computed. For example, the data could have been fed into an unsupervised deep learning model which could perfect the model weightings through many iterations.

4) Improved regression methods

If access to a greater amount of computing power was available as well as a longer period of time, several other forms of regression methods apart from the currently implemented linear regression could have been investigated.

For example, a common non-linear regression technique is the use of segmented or piecewise regression. This is a form of data fitting that breaks the curve of best fit into several separate domains, allowing for greater accuracy in modelling non-linear data sets.



Above: A representation of piecewise regression (source 6)

Piecewise/segmented regression could also allow for a dynamic model which can exhibit a variety of behaviours with different weights for different inputs. Instead of modelling all countries on one linear line, 2 or more separate curves could provide a better fit for outliers in the current model - such as the island/3rd world/war ridden nations.

Furthermore, another available type of non-linear regression is exponential regression. This type of regression is used to model situations which tend to accelerate or decay rapidly.

By using exponential regression, a significant reduction in the model's R-Squared value can be expected. This will allow for significantly improved weighting and increase the model's chances of working in situations where a factor doesn't affect life expectancy in a linear manner (a weakness of the current model).

5.0 Conclusion

Through the statistical analysis and fitting of data gathered from Gapmider and World Bank, a polynomial based model was created. The model, created using linear regression analysis, takes into account five different factors deemed to be significant: basic sanitation rate, doctors per thousand people, GDP per capita, government health spending per capita, and primary school completion rate.

Although several flaws exist with the current model, its focus on parameters other than mortality rate allow it to be applied as a useful indicator for population health. In the future, the model may be used to identify areas of concern for nations, and help those nations work towards a more equitable society.

6.0 Appendix

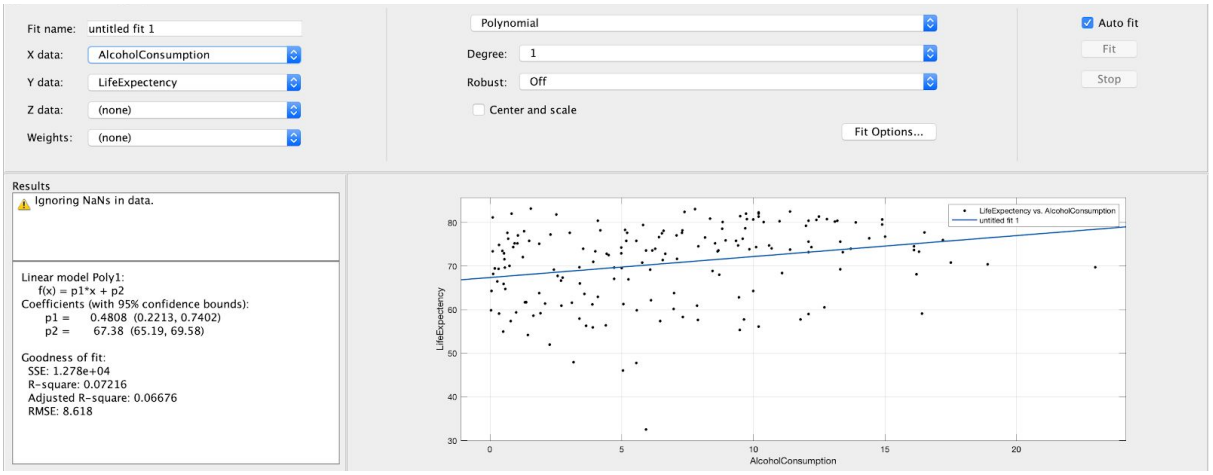


Figure 1: Alcohol Consumption against Life Expectancy

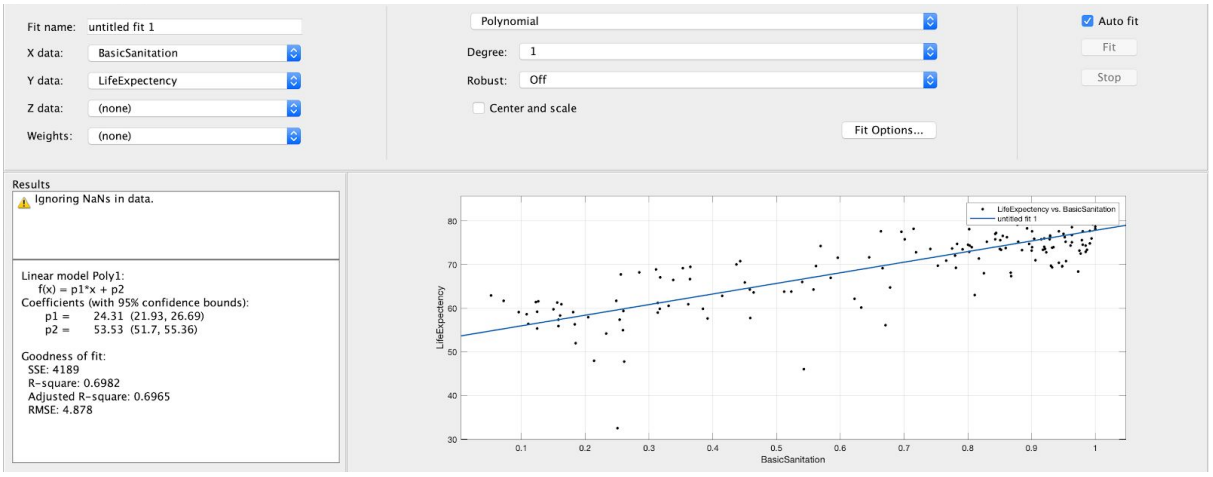


Figure 2: Basic Sanitation against Life Expectancy

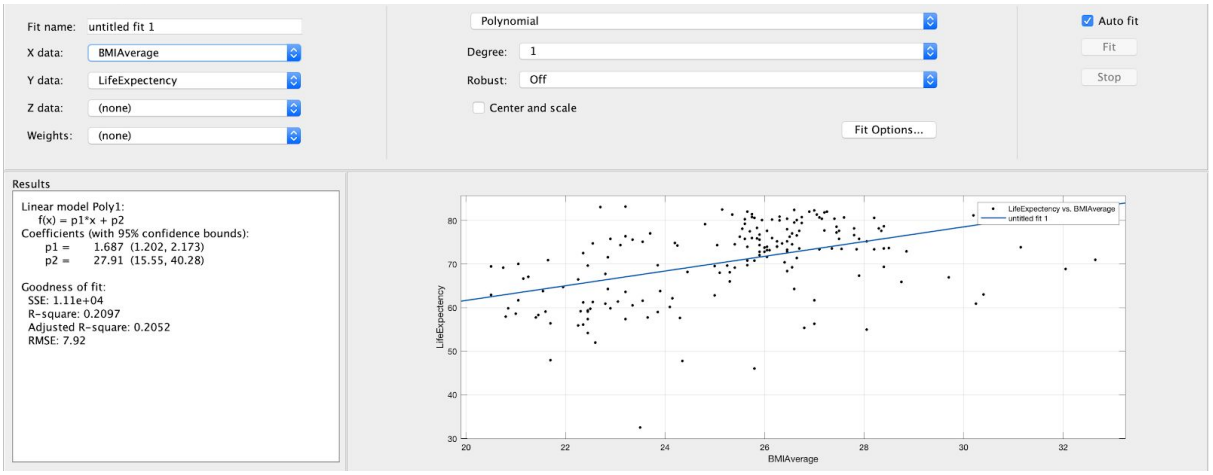


Figure 3: BMI against Life Expectancy

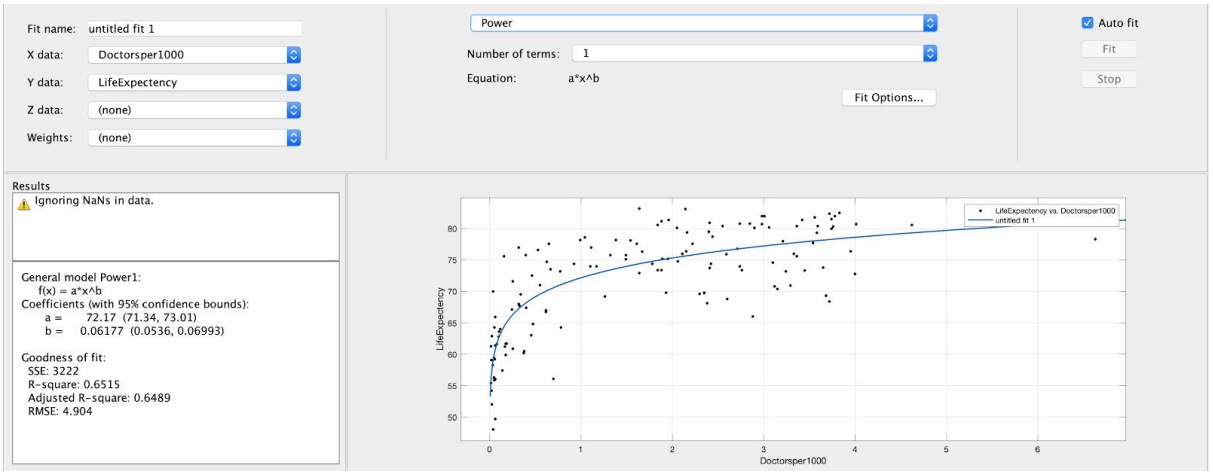


Figure 4: Doctors per 1000 against Life Expectancy

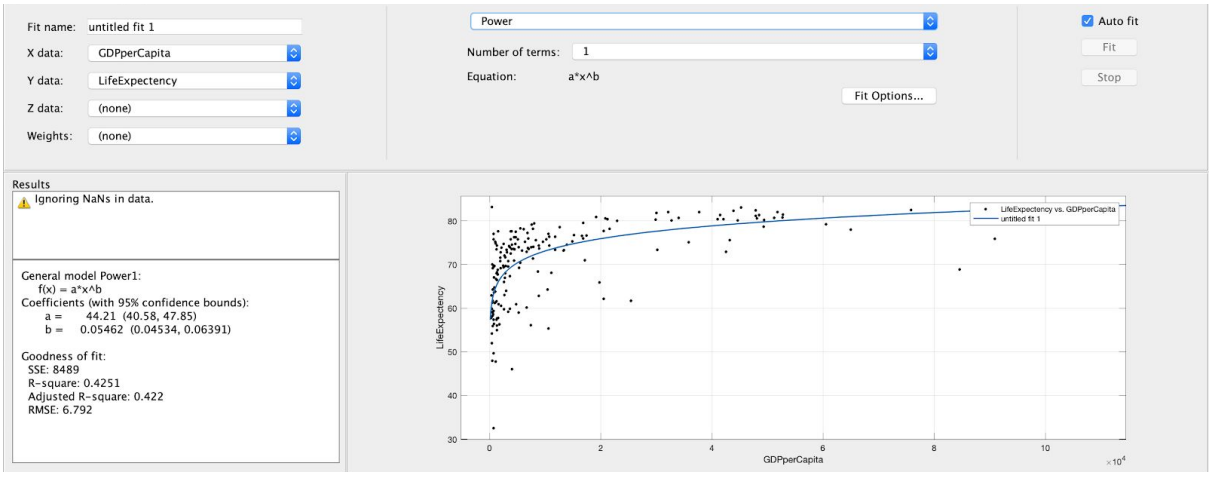


Figure 5: GDP per capita USD against Life Expectancy

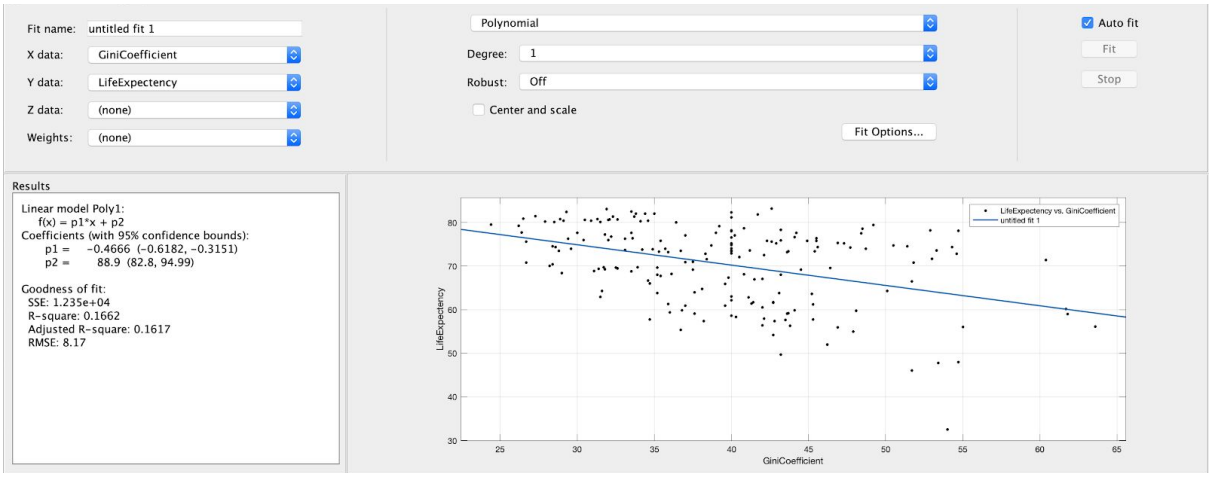


Figure 6: Gini Coefficient against Life Expectancy

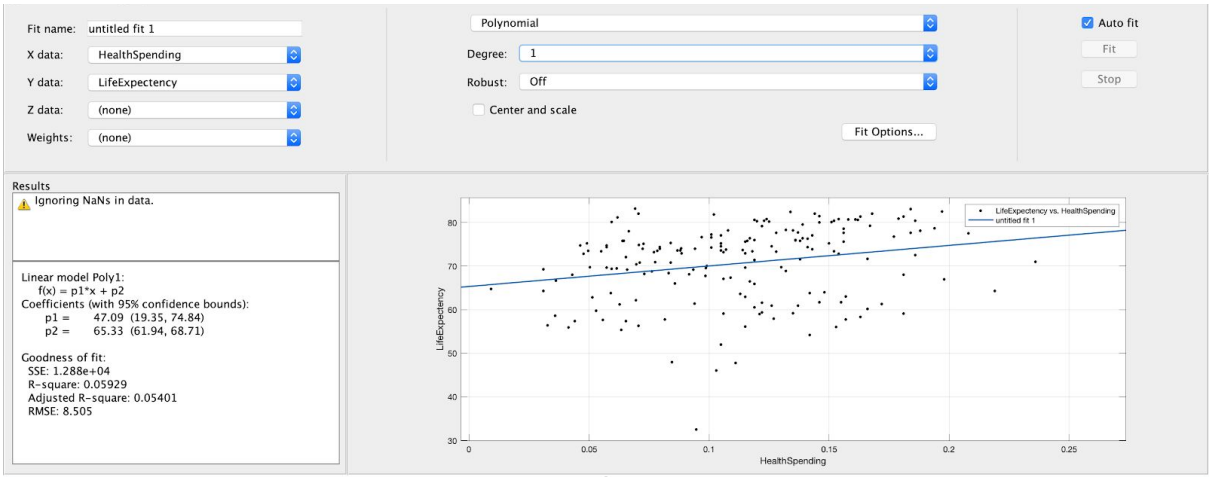


Figure 7: Health Spending against Life Expectancy

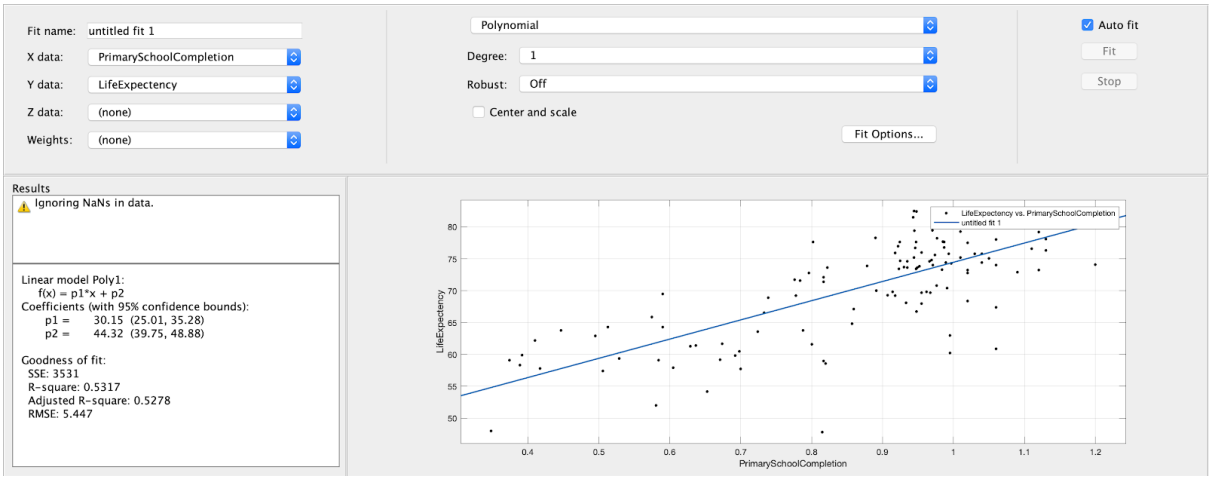


Figure 8: Primary School Completion Rate (%) against Life Expectancy

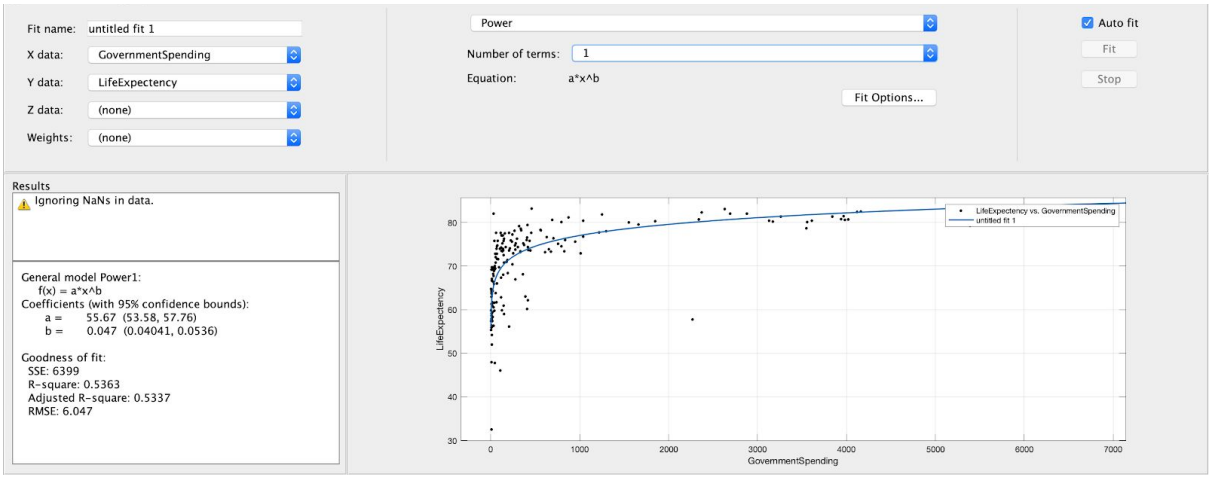


Figure 9: Government Spending against Life Expectancy

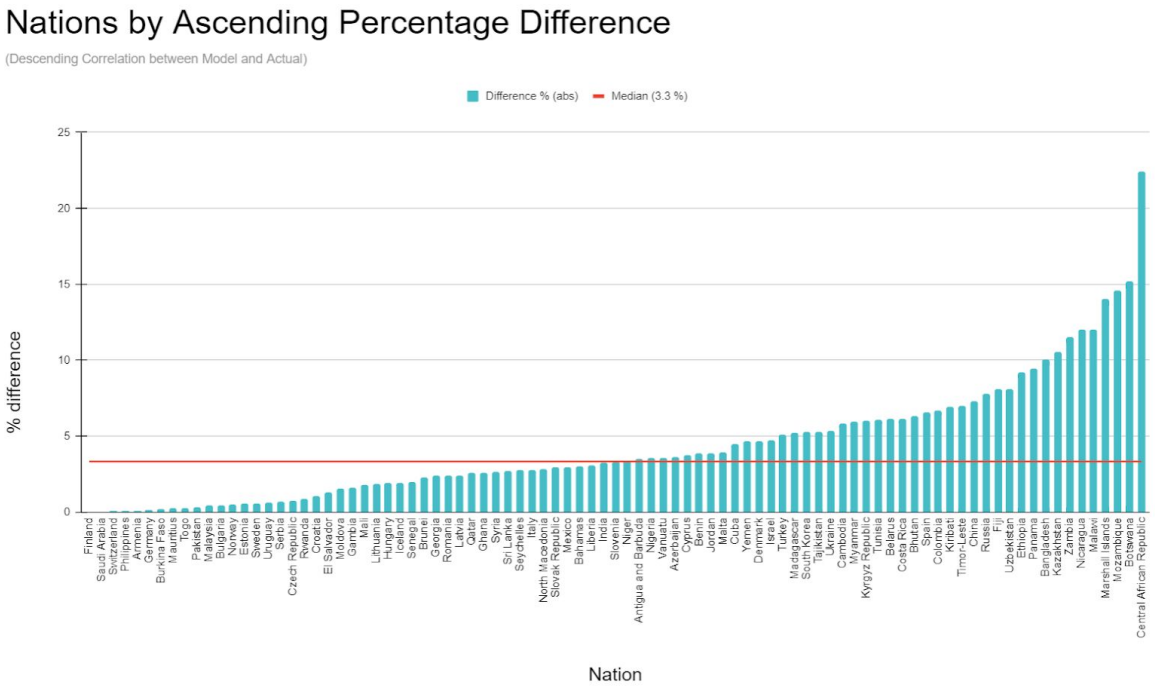


Figure 10: Nations sorted by Ascending Percentage Difference

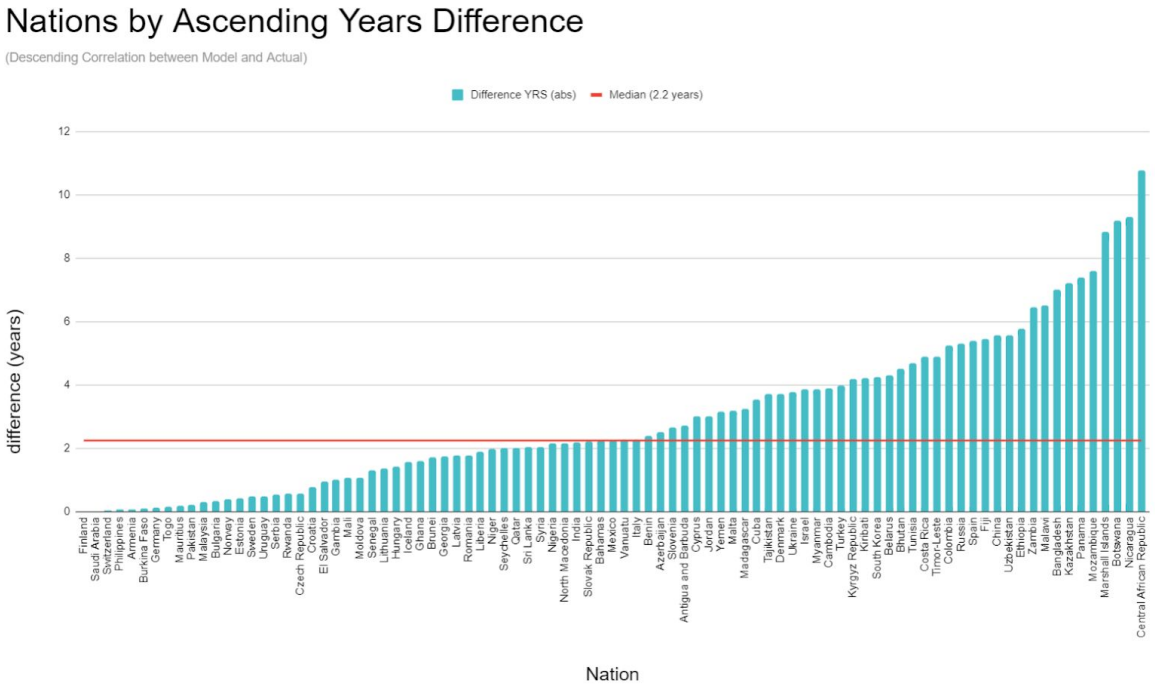


Figure 11: Nations sorted by Ascending Years Difference

Nation	Life Expectancy	Basic Sanitation	Doctors per 1000	GDP per Capita	Health Spending	Primary School Completion	Expected Life Expectancy	Difference YRS	Difference %
Switzerland	82.5	0.999	3.83	75800	4160	0.944	85.9030097	3.4030097	4.124860242
Iceland	82.4	0.988	3.72	47900	4120	0.948	82.9208424	0.5208424	0.6320902913
Italy	82	0.988	3.78	37700	2690	1.06	81.6155476	0.3844524	0.4688443902
Spain	82	0.999	2.98	32100	31.7	0.973	78.1507276	3.8492724	4.694234634
Israel	81.8	1	3.56	30000	1250	1.01	79.408285	2.391715	2.923856968
Sweden	81.5	0.993	3.74	52800	3970	0.943	83.3293784	1.8293784	2.244636074
Norway	80.9	0.981	2.41	19200	6800	0.992	81.9703659	1.0703659	1.323072806
South Korea	80.6	1	1.84	20800	692	1	77.366976	3.233024	4.01119603
Germany	80.4	0.992	3.59	42100	3610	1.02	82.4812194	2.0812194	2.588581343
Malta	80.4	1	3.36	21200	1040	1.01	78.290505	2.109495	2.62375
Finland	80.2	0.994	3.06	49400	3170	0.988	82.4625226	2.2625226	2.821100499
Cyprus	80.1	0.996	2.05	32700	796	1	78.6582782	1.4417218	1.799902372
Slovenia	79.5	0.991	2.4	16800	1660	0.97	77.5882317	1.9117683	2.40474
Costa Rica	79.4	0.951	2.16	8030	413	0.945	74.8912252	4.5087748	5.678557683
Denmark	79.3	0.996	3.58	60500	5390	1.01	85.7220962	6.4220962	8.098481967
Cuba	78.3	0.893	6.63	5510	557	0.89	75.1026841	3.1973159	4.083417497
Panama	78.2	0.715	1.38	7810	341	0.977	71.5240052	6.6759948	8.537077749
Colombia	78.1	0.802	1.54	6120	266	1.13	73.6951624	4.4048376	5.639996927
Qatar	78	1	3.21	65000	1300	1.06	83.200175	5.200175	6.666891026
Czech Republic	77.7	0.991	3.54	20500	1220	0.985	78.1161792	0.4161792	0.535623166
Jordan	77.6	0.98	2.22	3790	204	0.947	74.7358787	2.8641213	3.690877964
Turkey	77.6	0.901	1.61	3950	455	0.987	73.9495094	3.6504906	4.704240464
Nicaragua	77.6	0.664	0.649	1530	57.7	0.802	68.4019251	9.1980749	11.8531893
Tunisia	77	0.843	1.11	10600	133	0.922	72.8963343	4.1036657	5.329435974
Croatia	76.8	0.963	2.71	15200	1040	0.987	76.7813848	0.0186152	0.02423854167
Antigua and Ba	76.6	0.855	0.527	16600	409	1.11	75.0854848	1.5145152	1.97717389
Uruguay	76.4	0.948	3.95	10700	460	1.07	76.7139936	0.3139936	0.4109863874
Estonia	76	0.994	3.33	16800	836	0.955	77.1826823	1.1826823	1.556160921
North Macedor	75.9	0.904	2.59	90900	223	0.918	82.3260796	6.4260796	8.466508037
Syria	75.8	0.928	1.49	707	41.8	1.04	74.0216026	1.7783974	2.346170712
China	75.8	0.701	1.32	3800	78.2	0.993	70.818429	4.981571	6.571993404
Slovak Republic	75.6	0.979	3.36	43200	948	0.974	79.8579957	4.2579957	5.632269444
Mexico	75.2	0.83	1.95	9590	281	1.01	73.685767	1.514233	2.013607713
Brunei	75.1	0.963	1.49	35800	754	1.05	78.6581731	3.5581731	4.737913582

Figure 12: Sorted difference between expected and observed life expectancies

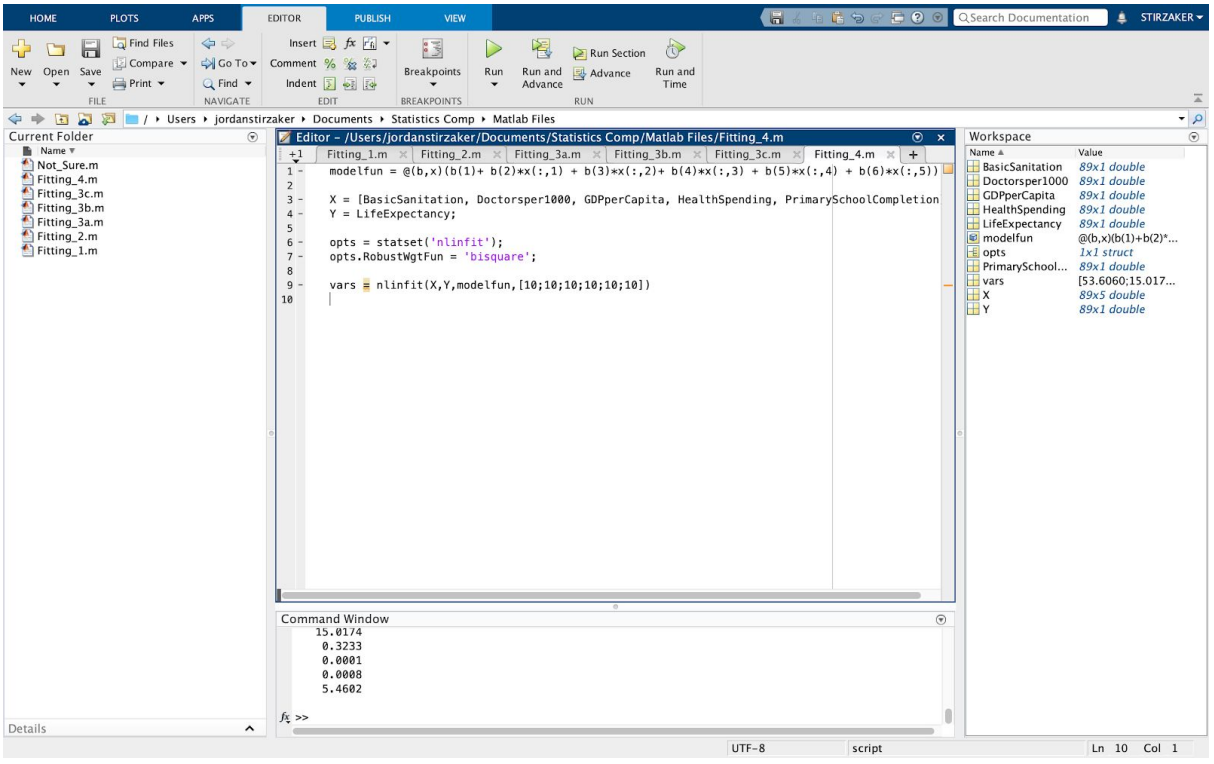


Figure 13:

7.0 References

1) Data. Retrieved 17 July 2020, from https://www.gapminder.org/data/Roser, M., Ortiz-Ospina, E., & Ritchie, H. (2019). Life Expectancy. Retrieved 17 July 2020, from https://ourworldindata.org/life-expectancy
2) Goodness of Fit Statistics. Retrieved 17 July 2020, from https://web.maths.unsw.edu.au/~adelle/Garvan/Assays/GoodnessOfFit.html#
3) An overarching health indicator for the Post-2015 Development Agenda. (2014). Retrieved 17 July 2020, from https://www.who.int/healthinfo/indicators/hsi_indicators_SDG_TechnicalMeeting_December2015_BackgroundPaper.pdf
4) Kappel, S. (2017). Piecewise regression: when one line simply isn't enough. Retrieved 23 July 2020, from https://www.datadoghq.com/blog/engineering/piecewise-regression/

PROBLEM 7; TRANSFORMATIONS

Leigh Greville and Oliver Kalicin
Albert Park College

ALBERT PARK COLLEGE

Introduction

In this poster, 'a mapping' should typically be taken to mean 'a mapping using some finite combination of the allowed transformations'. Similarly, 'to map' should typically be taken to mean 'to map using some finite combination of the allowed transformations'

Lemma 1

It is possible to map any rational number $\frac{a}{b}$ to $\frac{a}{b}+n$ for $n \in \mathbb{Z}$.
Proof by construction:
Let $T_3(x) = (T_2 \circ T_1^{-1})(x)$. Therefore,

$$T_3(x) = 1 - \frac{1}{\frac{1}{2-x}} = 1 - (2-x) = x - 1 \tag{1}$$

Similarly, let $T_4(x) = (T_1 \circ T_2^{-1})(x)$. Then,

$$T_4(x) = 2 - \frac{1}{\frac{1}{1-x}} = 2 - (1-x) = x + 1 \tag{2}$$

Clearly, repeated applications of either T_3 or T_4 allow one to add any integer to a rational number

Lemma 2

It is possible to map any rational number $\frac{a}{b}$ to a rational number $\frac{a^*}{b}$, where $0 \leq a^* < b$.
Proof:
Let $a = k * b + c$, where $k, c \in \mathbb{Z}$ and $0 \leq c < b$ (in other words, $c \equiv a \mod b$). Therefore,

$$\frac{a}{b} = \frac{k*b+c}{b} = k + \frac{c}{b} \tag{3}$$

By lemma 1, it is possible to map any rational number $\frac{a}{b}$ to $\frac{a}{b}+n$ for $n \in \mathbb{Z}$. Taking $n = -k$, $k + \frac{c}{b}$ can be mapped to $\frac{c}{b}$. Therefore, $\frac{a}{b}$ can be mapped to $\frac{c}{b}$. By the condition described above, $0 \leq c < b$, so a^* can be taken to equal c and therefore lemma 2 is true.

Lemma 3

It is possible to map any rational number $\frac{a}{b}$ to a rational number $\frac{b^*}{a}$.
Proof by construction:

$$T_2(\frac{a}{b}) = 1 - \frac{1}{\frac{a}{b}} = 1 - \frac{b}{a} = \frac{a-b}{a} \tag{4}$$

Taking $b^* = a - b$, lemma 3 is true.

An inductive proof that all rational numbers can be mapped to 0

If P is true for $n \Rightarrow P$ is true for $n+1$:
Consider a number $n \in \mathbb{N}$ for which the following statement (P) is true: for all $\phi \in \mathbb{N}$ where $0 < \phi < n$, there is a mapping of any rational number of the form $\frac{a}{\phi}$ to 0. We will show that this statement entails that a mapping exists for all rational numbers of the form $\frac{a}{n}$, which is equivalent to saying that P is true for $n+1$. By lemma 2, it is possible to map any rational number $\frac{a}{n}$ to a rational number $\frac{a^*}{n}$, where $0 \leq a^* < n$. Then, by lemma 3, one can map $\frac{a^*}{n}$ to $\frac{n^*}{a^*}$, as long as $a^* \neq 0$. Importantly, we need not consider $a^* = 0$ as if $a^* = 0$, $\frac{a^*}{n} = 0$, so a mapping to 0 has already been achieved. This means that $0 < a^* < n$, so it is possible to map $\frac{n^*}{a^*}$ to 0, by P. Therefore, if P is true for n, it is also true for $n+1$.

P is true for the base case:
The base case in this induction argument is $n = 1$, as for $n = 0$ the fraction $\frac{a}{n}$ is undefined. For $n = 1$, $\frac{a}{n} = a$. By lemma 1, all fractions of this form can be mapped to 0 by adding $-a$ to them. Therefore, the base case is true.

On the completeness of this induction argument:
This inductive process reaches all rational numbers as it proves that for all rationals with denominators within $\mathbb{N}$, there is a mapping that takes these numbers to 0. The numerator of these fractions is allowed to be any number within $\mathbb{Z}$, so this spans the entire rational field. Therefore, for all rational numbers $\frac{a}{b}$, there is a mapping that takes them to 0.

Solution

By following the proof that a mapping to 0 exists for all rational numbers, a method for finding such a mapping quickly emerges. Namely:

1. If $\frac{a}{b} = 0$, finished
2. If $\frac{a}{b} > 0$, apply T_3 until the numerator of the resulting fraction is between 0 and b
3. Else, apply T_4 until the same condition is met
4. Apply T_2
5. Repeat steps 1-4 until the denominator of the fraction is 1 (until the fraction has been mapped to an integer)
6. If the integer is greater than 0 then apply T_3 until the integer has been mapped to 0
7. Else, apply T_4 until the same condition is met

Other sets of transformations

Following the proof, it can be seen that for any set of transformations where lemmas 1, 2 and 3 are true, then one can use this set of transformations to map any rational number to 0. Furthermore, as lemma 1 entails lemma 2, as long as a set of transformations obeys these properties:

1. It is possible to map any rational number $\frac{a}{b}$ to $\frac{a}{b}+n$ for $n \in \mathbb{Z}$ (lemma 1)
2. It is possible to map any rational number $\frac{a}{b}$ to a rational number $\frac{b^*}{a}$ (lemma 3)

then this set of transformations can map any rational number to 0. However, this condition is merely sufficient, not necessary, so it is possible that there are sets of transformations on the rational numbers for which lemma 1 and lemma 3 are not true that can still map every rational number to 0.

Infinite mappings

Suppose:

$$T_1^n(x) = \frac{(n+1)x - n}{nx - (n-1)} \text{ for some } n \in \mathbb{N} \tag{5}$$

Then:

$$\begin{aligned} T_1^{n+1}(x) &= T_1(\frac{(n+1)x-n}{nx-(n-1)}) \\ &= 2 - \frac{1}{\frac{(n+1)x-n}{nx-(n-1)}} \\ &= \frac{2*((n+1)x-n) - (nx-(n-1))}{(n+1)x-n} \\ &= \frac{(n+2)x-(n+1)}{(n+1)x-n} \end{aligned} \tag{6}$$

The base case, $n = 1$, is clearly true as,

$$\begin{aligned} T_1(x) &= 2 - \frac{1}{x} \\ &= \frac{2x-1}{x} \end{aligned} \tag{7}$$

Therefore,

$$T_1^n(x) = \frac{(n+1)x-n}{nx-(n-1)} \text{ for all } n \in \mathbb{N} \tag{8}$$

And moreover,

$$T_1^n(x) = 1 + \frac{x-1}{n(x-1)+1} \tag{9}$$

From (9), it can be seen that,

$$\lim_{n\to\infty} T_1^n(x) = 1 \text{ for all } x \in \mathbb{R} \tag{10}$$

And as $T_2(1) = 0$,

$$T_2(\lim_{n\to\infty} T_1^n(x)) = 0 \text{ for all } x \in \mathbb{R} \tag{11}$$

Hence, for all real numbers, there is an mapping of infinite length to 0.

Complex numbers

Consider the result when one applies the supplied transformations to a general complex number $a + bi$,

$$T_1(a+bi) = \frac{2(a^2+b^2)-1}{a^2+b^2} + \frac{b}{a^2+b^2}i \tag{12}$$

$$T_2(a+bi) = \frac{a^2+b^2-1}{a^2+b^2} + \frac{b}{a^2+b^2}i \tag{13}$$

$$T_1^{-1}(a+bi) = \frac{2-a}{(2-a)^2+b^2} + \frac{b}{(2-a)^2+b^2}i \tag{14}$$

$$T_2^{-1}(a+bi) = \frac{1-a}{(1-a)^2+b^2} + \frac{b}{(1-a)^2+b^2}i \tag{15}$$

This shows that none of these transformations could map a complex number with non-zero imaginary component to a real number, and hence could never (in a finite number of steps) map all complex numbers to 0. However, the above argument 'infinite transformations' can be generalised to the complex numbers, as none of its steps are false for $x \in \mathbb{C}$. Consequently, any complex number can still be reduced to 0, but only using infinite transformations.

CONWAY'S RATIONAL TANGLES AND RATIONAL NUMBERS

AARON DO AND TOM HE

Scotch College, Melbourne

Abstract. The reduction of rational numbers through a pair of functions and their inverses is analogous to the untangling of Conway's rational tangles. The representation of rational numbers as corresponding tangles through continued fractions yields a method to reduce rational numbers to 0: alternating expressions of inverse tangle operations with the functions examined form the basis of this approach. A follow up study aimed at determining a universal method for a family of functions in similar forms is also suggested.

1. Introduction

Functions, as presented on a high school level, are often studied in relation to algebra and calculus. Function composition, however, is often used to construct faithful representations of groups in abstract algebra, where the functions form right actions on corresponding groups.

The problem to be examined here was posed by the University of Melbourne, and involves the mapping of rational numbers to 0 through two functions and their inverses; we define them as follows.

Definition 1.1. Let T_1, T_2, and their inverses be functions on $\mathbb{Q}^* \to \mathbb{Q}^*$, where $\mathbb{Q}^* = \mathbb{Q} \cup \{\infty\}$, $\frac{1}{\infty} \mapsto 0$, $\frac{1}{0} \mapsto \infty$.

$$T_1(x) = 2 - \frac{1}{x} \qquad T_2(x) = 1 - \frac{1}{x}$$
$$T_1^{-1}(x) = \frac{1}{2 - x} \qquad T_2^{-1}(x) = \frac{1}{1 - x}$$

The prominence of function composition in the reduction of a rational number suggests a connection to abstract algebra. A closer observation of the problem yields an interesting resemblance to Conway's rational tangles; in particular, the functional representation of geometric tangle operations allows us to draw parallels between the untangling process of tangles and the reduction of rational numbers. This forms the basis of our approach: under such interpretation, we produce a method of reducing rationals that is analogous to the untangling of tangles.

2. Conway's Rational Tangles

Before we examine the applications of rational tangles to our problem, we provide a brief outline of rational tangles.

Key words and phrases. Rational numbers, Conway's rational tangles, transformations.

2.1. **A Tale of T and R.** By slicing apart knots and pinning down the cut ends, we obtain a class of mathematical objects known as *tangles*. We begin by considering two geometric operations, *Twist* and *Rotate*, that act on rational tangles.

Definition 2.1 ([2])**.** A rational tangle is any tangle which can be obtained by performing the following operations on an empty tangle, where an empty tangle (denoted by [0]) describes two horizontal, parallel strands.

(1) **"Twisting"** bottom right over top right. (T)
(2) **"Rotating"** the whole tangle 90° clockwise. (R)

Definition 2.2 (Simple Rational Tangles [1])**.**

(1) An infinity tangle, denoted by $[\infty]$, describes two vertical, parallel strands.
(2) An integer tangle, denoted by $[n]$, is made up of n horizontal twists. ($n \in \mathbb{Z}$)
(3) A vertical tangle, denoted by $\frac{1}{[n]}$, is made up of n vertical twists. ($n \in \mathbb{Z}$)

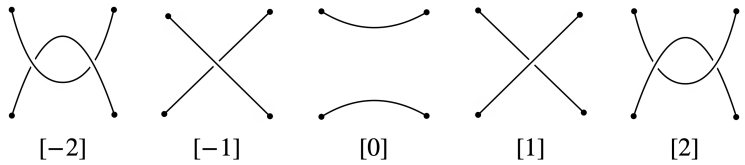


FIGURE 1. Integer tangles.

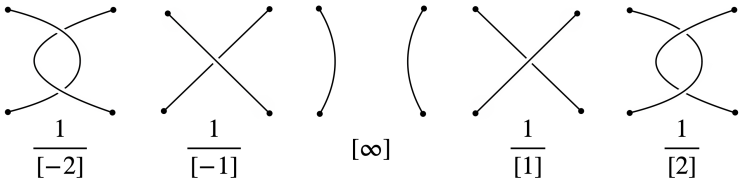


FIGURE 2. Vertical tangles.

If we consider the set $\mathfrak{T}$ to include all rational tangles, then the operations T and R are functions from $\mathfrak{T} \to \mathfrak{T}$. We then introduce a group, $(\Gamma, \circ)$, which includes the collection of all such functions under composition.

Theorem 2.3 ([2])**.** *The collection of all finite combinations of T and R forms a group $(\Gamma, \circ)$ under composition. The group has the presentation:*

$$\Gamma = \langle\, T, R \,|\, R^2 = I = (TR)^3 \,\rangle.$$

From these properties, we can form a right action of Γ on $\mathbb{Q}^*$ by defining functions t and r such that

$$t(x) = x + 1 \qquad r(x) = -\frac{1}{x}.$$

2.2. **Tangle Arithmetic.** A more systematic way of characterising and manipulating tangles can be derived from T and R by defining additive and multiplicative operations on tangles.

Definition 2.4 ([1])**.** Let G be a rational tangle with tangle number x. Then,

(1) Making n twists on the right of a tangle is expressed as:

$$G + [n] \rightsquigarrow x + n.$$

(2) Making n twists on the bottom of a tangle is expressed as:

$$G * \frac{1}{[n]} \rightsquigarrow \frac{1}{n + \frac{1}{x}}.$$

We also consider Lemma 2 from Kauffman and Lambropoulou's paper [1].

Lemma 2.5 (KL Flipping Lemma [1])**.** *We denote T^{hflip} as the tangle obtained from T by a $180°$-rotation around a horizontal axis on the plane of T, and T^{vflip} as the tangle obtained from T by a $180°$-rotation around a vertical axis on the plane of T. If T is rational, then:*

$$(i)\ T \sim T^{hflip} \text{ and } (ii)\ T \sim T^{vflip}.$$

A consequence of this lemma is the commutativity of tangle addition and multiplication, as summarised in the following corollary. Importantly, this implies that all rational tangles can be constructed using only bottom and right twists, as top and left twists can be flipped to the other side with no effects on the tangle number, as shown.

Corollary 2.6 ([1])**.** *For $m, n \in \mathbb{Z}$ and rational tangle T:*

$$[m] + T + [n] \sim T + [m + n],\ \frac{1}{[m]} * T * \frac{1}{[n]} \sim T * \frac{1}{[m + n]}.$$

This is key to tangle arithmetic: it allows us to assign every rational tangle a tangle number.

Theorem 2.7 (Continued Fraction Theorem [1])**.** *Let G be a rational tangle and x be its tangle number. Then G is equivalent to a tangle with a continued fraction representation, where $a_1, \ldots, a_n \in \mathbb{Z}^+ \cup 0$ or $\mathbb{Z}^- \cup 0$, and n is an odd positive integer:*

$$G = [[a_1], [a_2], [a_3], \ldots, [a_n]] = [a_1] + \cfrac{1}{[a_2] + \cfrac{1}{[a_3] + \cdots + \cfrac{1}{\ddots_{+\frac{1}{[a_n]}}}}},$$

where its tangle number can be expressed as:

$$x = [a_1, a_2, a_3, \ldots, a_n] = a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{\ddots_{+\frac{1}{a_n}}}}}.$$

3. From Rational Tangles to Rational Numbers

The faithful representation of the tangle group using rational numbers and infinity ($\mathbb{Q}^*$) allows us to establish a relationship between the mapping of rational numbers and geometric transformations on rational tangles. We begin by drawing algebraic parallels between the two.

3.1. **A Geometric Interpretation.** As established, all rational tangles can be constructed using a series of right and bottom twists. Through algebraic manipulation, we express the inverses of the right and bottom twist operations in terms of T_1, T_2, and their inverses.

Lemma 3.1 (The Inverse Right Twist Lemma)**.** $T_2 \circ T_1^{-1}(x) = t^{-1}(x) = x - 1$.

Proof. The functional representation of a right twist is defined as $t(x) = x + 1$. Because this is an injective function, there exists an inverse $t^{-1}(x)$ such that $t^{-1} \circ t(x) = x$. Since

$$T_2 \circ T_1^{-1}(x) = T_2(\frac{1}{2-x}) = x - 1,$$

substituting $x = t(x)$ yields

$$T_2 \circ T_1^{-1}(t(x)) = t(x) - 1 = x.$$

□

Lemma 3.2 (The Inverse Bottom Twist Lemma)**.** $T_2^{-1} \circ T_1(x) = b^{-1}(x) = \frac{1}{-1+\frac{1}{x}}$

Proof. The addition of bottom twists to a rational tangle T with tangle number x is defined as

$$T * \frac{1}{[n]} \rightsquigarrow \frac{1}{n + \frac{1}{x}}.$$

Letting $n = 1$, we form a functional representation of the addition of 1 bottom twist:

$$b : \mathbb{Q}^* \to \mathbb{Q}^*, b(x) = \frac{1}{1 + \frac{1}{x}}.$$

Since this is an injective function, there exists an inverse $b^{-1}(x)$ such that $b^{-1} \circ b(x) = x$. We consider the following:

$$T_2^{-1} \circ T_1(x) = T_2^{-1}(2 - \frac{1}{x}) = \frac{1}{-1 + \frac{1}{x}}.$$

Substituting $x = b(x)$ yields

$$T_2^{-1} \circ T_1(b(x)) = \frac{1}{-1 + \frac{1}{b(x)}} = x.$$

□

Since every rational number corresponds to a class of rational tangles with the same tangle number, and every rational number can be written in the form of a continued fraction, the reduction of rational numbers is a faithful representation of the untangling process of a rational tangle, where the operations introduced previously can be used to undo positive right and bottom twists.

3.2. **Reduction of Rational Numbers.** The invertibility of functions warrants a method of untangling a tangle through the inverses of b and t. To determine this, we first present a method of expressing a rational number as a continued fraction.

Proposition 3.3 (Continued Fraction Form of Positive Rational Numbers)**.** Every positive rational number can be expressed as a continued fraction through the Euclidean algorithm.

Proof. Consider the positive rational number $\frac{a}{b}, a \perp b, a, b \in \mathbb{Z}^+$. We can apply the Euclidean algorithm as follows:

$$\begin{aligned} a &= q_1 b + r_1 & \frac{a}{b} &= q_1 + \frac{r_1}{b} \\ b &= q_2 r_1 + r_2 & \frac{b}{r_1} &= q_2 + \frac{r_2}{r_1} \\ &\vdots & &\vdots \\ r_{n-3} &= q_{n-1} r_{n-2} + r_{n-1} & \frac{r_{n-3}}{r_{n-2}} &= q_{n-1} + \frac{r_{n-1}}{r_{n-2}} \\ r_{n-2} &= q_n r_{n-1} & \frac{r_{n-2}}{r_{n-1}} &= q_n, \end{aligned}$$

where $q_1 \in \mathbb{Z}^+ \cup \{0\}$, $n, r_1, \ldots, r_{n-1}, q_2, \ldots, q_n \in \mathbb{Z}^+, b > r_1 > r_2 > \cdots > r_{n-1}$. We note that $q_n \geq 2$ ($r_{n-2} > r_{n-1}$ and $r_{n-1} \mid r_{n-2}$) and that the algorithm is a finite process (the terms are decreasing). As such, we can express $\frac{a}{b}$ as a continued fraction with finite elements:

$$\frac{a}{b} = q_1 + \frac{1}{\frac{b}{r_1}} = q_1 + \frac{1}{q_2 + \frac{r_1}{r_2}} = q_1 + \frac{1}{q_2 + \frac{1}{\ddots + \frac{1}{q_n}}} = [q_1, q_2, \ldots, q_n]$$

□

This systematic method of converting rational numbers gives rise to our method.

Theorem 3.4 (Rational Reduction Theorem)**.** *All rational numbers can be reduced to 0 using T_1, T_2, and their inverses.*

Proof. For negative rationals, we first map it to a positive rational number using T_2. Consider the positive rational number $\frac{a}{b}$, which can be expressed as

$$\frac{a}{b} = [q_1, q_2, \ldots, q_n] \rightsquigarrow [[q_1], [q_2], \ldots, [q_n]] = ([q_n] * \frac{1}{[q_{n-1}]} + [q_{n-2}]) * \cdots + [q_1].$$

(If n is even, the expression is changed to $[q_1, q_2, \ldots, q_{n-1}, 1]$ to satisfy the canonical form of tangles.) Knowing the construction of the tangle that corresponds to the number enables us to use the inverse functions of right and bottom twist to untwist the tangle to the [0] tangle. The reduction sequence can be expressed as

$$(T_2 \circ T_1^{-1})^{q_n} \circ \cdots \circ (T_2^{-1} \circ T_1)^{q_2} \circ (T_2 \circ T_1^{-1})^{q_1}(x).$$

□

We illustrate this in the following example.

Example 3.5. Consider the rational number $\frac{7}{5}$.

$$\begin{aligned} 7 &= 1 \times 5 + 2 \\ 5 &= 2 \times 2 + 1 \\ 2 &= 2 \times 1 + 0 \end{aligned}$$

Therefore $\frac{7}{5} = [1, 2, 2]$. The reduction sequence yields:

$$\begin{aligned}(T_2 \circ T_1^{-1})^2 \circ (T_2^{-1} \circ T_1)^2 \circ (T_2 \circ T_1^{-1})(\frac{7}{5}) &= (T_2 \circ T_1^{-1})^2 \circ (T_2^{-1} \circ T_1)^2(\frac{2}{5}) \\ &= (T_2 \circ T_1^{-1})^2(2) \\ &= T_2 \circ T_1^{-1} \circ T_2(\infty) \\ &= T_2 \circ T_1^{-1}(1) \\ &= 0\end{aligned}$$

3.3. **Construction of Rational Numbers.** Mirroring the inverse right and bottom twist operations, we express the right and bottom twist functions in terms of T_1, T_2, and their inverses.

Lemma 3.6 (The Right Twist Lemma)**.** $T_1 \circ T_2^{-1}(x) = t(x) = x + 1$

Proof. $T_1 \circ T_2^{-1}(x) = T_1(\frac{1}{1-x}) = x + 1 = t(x)$ □

Lemma 3.7 (The Bottom Twist Lemma)**.** $T_1^{-1} \circ T_2(x) = b(x) = \frac{1}{1+\frac{1}{x}}$

Proof. $T_1^{-1} \circ T_2(x) = T_1^{-1}(1 - \frac{1}{x}) = \frac{1}{1+\frac{1}{x}} = b(x)$

□

As such, we can generalise the problem to include the construction of any rational number from any other rational number.

Corollary 3.8 (Generalised Rational Reduction)**.** *Any rational number can be mapped to any other rational number through T_1, T_2, and their inverses.*

Proof. Suppose we would like to map the rational number $x = [q_1, \ldots, q_n]$ to $y = [p_1, \ldots, p_m]$. We first reduce the rational using the sequence

$$(T_2 \circ T_1^{-1})^{q_n} \circ \cdots \circ (T_2^{-1} \circ T_1)^{q_2} \circ (T_2 \circ T_1^{-1})^{q_1}(x).$$

Then, we can construct the rational y using the right and bottom twist operations:

$$(T_1 \circ T_2^{-1})^{p_m} \circ \cdots \circ (T_1^{-1} \circ T_2)^{p_2} \circ (T_1 \circ T_2^{-1})^{p_1}(0) = y.$$

□

Example 3.9. Suppose we would like to map $\frac{3}{8}$ to $\frac{5}{2}$. We first express them in continued fraction form.

$$\begin{aligned}3 &= 0 \times 8 + 3 & 5 &= 1 \times 3 + 2 \\ 8 &= 2 \times 3 + 2 & 3 &= 1 \times 2 + 1 \\ 3 &= 1 \times 2 + 1 & 2 &= 2 \times 1 + 0 \\ 2 &= 2 \times 1 + 0 & \frac{5}{2} &= [1, 1, 2] \\ \frac{3}{8} &= [0, 2, 1, 2] = [0, 2, 1, 1, 1]\end{aligned}$$

This yields the sequence

$$\begin{aligned}&(T_1 \circ T_2^{-1})^2 \circ (T_1^{-1} \circ T_2) \circ (T_1 \circ T_2^{-1}) \circ \\ &(T_2 \circ T_1^{-1}) \circ (T_2^{-1} \circ T_1) \circ (T_2 \circ T_1^{-1}) \circ (T_2^{-1} \circ T_1)^2 \circ (T_2 \circ T_1^{-1})^0(\frac{3}{8}) = \frac{5}{2}.\end{aligned}$$

4. Final Remarks

The faithful representation of tangles through the functions introduced in the problem enables the geometric interpretation of rational numbers as rational tangles constructed from a series of right and bottom twists. Such an approach provides physical meaning to the domain of the functions – the inclusion of infinity is not a "mathematical hack"; rather, it refers to the well-defined infinity tangle. The mappings, $\frac{1}{\infty} \mapsto 0, \frac{1}{0} \mapsto \infty$, stem from this: the rotation operation ($r(x) = -\frac{1}{x}$) transforms the [0] and [∞] tangles into each other, which yields the mappings in their respective tangle numbers.

A useful generalisation of the problem is to explore pairs of functions $T_1(x) = m - \frac{1}{x}$ and $T_2(x) = k - \frac{1}{x}$ for $m, k \in \mathbb{Z}$ to investigate the conditions for which a general method of reducing fractions exists and whether they relate to Conway's rational tangles.

Acknowledgements

The authors thank Dr Ainsworth and Dr Coutis for providing helpful suggestions to earlier versions of the manuscript.

References

[1] Kauffman, L & Lombropoulou, S 2004, On the classification of rational tangles, pdf, viewed 11 July 2020, <https://arxiv.org/pdf/math/0311499.pdf>.

[2] Prof. Salomone, M 2018, OAK Rational Tangles, Open Algebra and Knots, viewed 3 June 2020, <http://matthematics.com/oak/chapter-tangles.html>.

Date: 22 July 2020.

Aaron Do, Scotch College
Email address: AD5609@scotchmel.vic.edu.au

Tom He, Scotch College
Email address: TH5894@scotchmel.vic.edu.au

Finding Maximal Prime Gaps

Oliver Tan Lucas Teoh Andrew Wang

July 20, 2020

1 Introduction

A prime number, p, is a positive integer that is only divisible by 1 or p (itself). In this paper, we examine a method, using Python code, of finding the maximal prime gap, that is, the largest difference between two consecutive prime numbers p_n and p_{n+1}, such that $p_n, p_{n+1} < 10^x, x \in N$. Section 2 explores the basic outline of such a function and the fundamental steps required in developing an algorithm to find prime gaps. Section 3 explores the Miller-Rabin primality test and demonstrates how it can be implemented to increase efficiency. Section 4 explores how the Logarithmic Integral function is used in conjunction with various other functions to simplify the search for a maximal prime gap. Section 5 displays the final algorithm and discusses further potential optimisations and limitations.

2 Development of the Algorithm

In this section we demonstrate the development of the base outline of a Prime Gap finding algorithm, as well as the further alterations that were made to increase its versatility and efficiency.

2.1 Constructing a base formula

To begin, the root steps in finding a prime gap less than 10^x needed to be outlined, and they were as follows:

1. Generate a list of primes less than 10^x
2. Check the difference between each successive pair of primes and record the largest difference
3. Return the largest difference

The premise of this initial method was not to immediately generate the perfect method, but rather to set a base algorithm to build from. From these three steps, we constructed, using Python code, the first algorithm.

```
def GetPrime(n):
    primeList = [2]
    num = 3
    while len(primeList) < n:
        for p in primeList:
            if num % p == 0:
                break
        else:
            primeList.append(num)
        num += 2
    return primeList[-1]

def PrimeGap(n):
    return(GetPrime(n+1)-GetPrime(n))

q = 1
t = 0
while GetPrime(q) < 10**x:
    if PrimeGap(q)>PrimeGap(t):
        t = q
    q += 1
```

Despite the inefficiency of this initial method, it provided a solid foundation on which to build on. As there were only two main steps - the prime finding and the prime gap finding, these were to be the main focus for optimisation and improvements.

2.2 Improving prime finding

The immediate shortcomings of this method lay within the fact that as x increased, the number checks would increase exponentially, resulting in a very quick growth in computing time.

```
def GetPrime(n):
    primeList = [2]
    num = 3
    while len(primeList) < n:
        for p in primeList:
            if num % p == 0:
                break
        else:
            primeList.append(num)
        num += 2
    return primeList[-1]
```

This initial method of obtaining prime numbers involved shovelling through every odd number and searching for a divisor in the list of all primes. This

involved largely numerous checks per number, and so it was an incredibly slow method that was only able to efficiently produce a list of primes up to 10^5. As such, a significantly better method was needed.

This method came to light as the Miller-Rabin primality test, which in simple terms, is more complex yet optimised prime determiner. The Miller-Rabin test will be explored in further detail in Section 3, however its most notable and beneficial trait was that the number of checks required to determine whether any given number was prime or not, remained constant. This allowed for a more improved algorithm, capable of determining the primality of even larger given numbers without increasing the number of checks required. Having now found a time-effective method of obtaining prime numbers, the next step was to improve the method for finding prime gaps.

2.3 Improving prime gap finding

In the same way that proved a hindrance in obtaining the initial set of primes, the number of primes still remained fairly substantial for higher values of x.

```
def PrimeGap(n):
    return(GetPrime(n+1)-GetPrime(n))

q = 1
t = 0
while GetPrime(q) < 10**x:
    if PrimeGap(q)>PrimeGap(t):
        t = q
    q += 1
```

This code was incredibly inefficient and wasteful, as not only did it have to comb through every single pair of primes to find the maximal gap, but with each pair of primes, it would generate a completely new list to pull the primes from. This long and arduous process could however, be removed, with the introduction of a new and significantly improved prime finding algorithm allowing for more options to be easily implemented in conjunction with the Miller-Rabin test.

In order to avoid the drawback of having to check every consecutive pair in the potentially hugely long list of primes, rough boundaries had to be made around areas where the maximal prime gap was most likely to be found. This is explored in further detail in Section 4, where a formula suggested by Marek Wolf is discussed. This formula utilises the Logarithmic Integral function and is able to produce an approximate number around which to look for such a gap, greatly reducing the range of numbers required to check.

3 The Miller-Rabin Primality Test

The Miller-Rabin primality test [5], constructed by Gary L. Miller in 1976 and refined by Michael O. Rabin in 1980, is an probabilistic algorithm used to approximate if a given number is prime or not. Using specific data points, this algorithm becomes deterministic below a certain threshold, one that is above any of x that we would need to equate.

3.1 Explaining the test

Lemma 3.1.1. *No square roots of 1 modulo p exist, such that p is prime and $p > 2$, other than those congruent to either 1 or -1 mod p.*

Proof. Suppose:

$$x^2 \equiv 1 \pmod{p}$$

It then follows, by the difference of perfect squares,

$$(x-1)(x+1) \equiv 0 \pmod{p}$$

By Euclid's Lemma, p must divide either $x-1$ or $x+1$, and so it follows that x is congruent to either 1 or -1 modulo p.

Now, for any prime number $n > 2$, it follows that $n-1$ is even. By taking the largest power of 2 from $n-1$, we can now say $n-1 = 2^s d$, where d and s are both positive integers, and d is odd. We say that for any a in Z/n, either

$$a^d \equiv 1 \pmod{n}$$
$$or$$
$$a^{2^r d} \equiv -1 \pmod{n}$$

where $0 \leq r \leq s-1$.
Fermat's Little Theorem states

$$a^{n-1} \equiv 1 \pmod{n}$$

for some prime number n. By Lemma 3.1.1., if we continually take square roots from a^{n-1}, we will end up with either 1 or -1. If we get -1, then we see that the second equality holds. If we do not get -1, then we are left with a^d, which will hold true to the first equality if n is indeed prime.

However, rather than checking every value of a to see if this holds for all, the Miller-Rabin test works with the opposite idea, that is, if we can find a value of a, such that

$$a^d \not\equiv 1 \pmod{n}$$
$$\text{and}$$
$$a^{2^r d} \not\equiv -1 \pmod{n}$$

then n is not prime.

3.2 Reliability

The overall correctness of this test on a large scale is reliant on the Riemann Hypothesis, which is at the time of writing, unproven. This is largely in part due to the existence of *strong liars*, which are values of a with which the equations hold, despite n being composite. However, there exist deterministic variants, which allow for only a specific set of a values to be tested, when n is within a certain limit, and these have been proven for n values up to $n < 3,317,044,064,679,887,385,961,981$, a 25 digit number. Given that the largest known prime gap is 20 digits long, using the Miller-Rabin algorithm for this specific task is acceptable, as the numbers being tested will not exceed its current known deterministic bounds.

3.3 Implementation into code

```
a_list = [2,3,5,7,11,13,17,19,23]

def MillerRabin(n):
    q = n-1
    r = 0
    while True:
       q = q>>1
       r += 1
       if q&1 == 1:
           break
    for a in a_list:
        check = True
        m = pow(a,q,n)
        if m == 1 or m == n-1:
            check = False
        else:
            for _ in range(c):
                m = pow(m,2,n)
                if m == n-1:
                    check = False
            if check:
                return False
    return True
```

The code begins with assigning q to the value $n-1$. As it moves into the `while` loop, q is converted into binary, and all powers of 2 are removed, leaving only the odd number, d. For each square root, the variable `r` (the r value) also increases. From here the code references a list, `a_list`. This list can alter based on the n value that needs to be checked, but since we will not be exceeding 10^{19} with any prime numbers, we can settle for the list shown, which has been proven to be enough to test $n < 3,825,123,056,546,413,051$. As the Miller-Rabin test looks for the inequalities, if the check variable is found to be false, for all a values in the set, the algorithm can output that n is indeed prime, but if the check variable remains true for just one a value, then n is known to be composite.

4 Estimating Prime Gaps

This section mainly explores some conjectures by Marek Wolf [14] and Daniel Shank which allow us to find an approximate range in which the maximal prime gap less than 10^x can be found. We will also explore the numerous other well established mathematical functions utilised in this approximation, and we will demonstrate how these are then implemented into the algorithm as code.

4.1 Wolf's conjecture

Wolf conjectures [2] that the first occurring largest prime gap $G(n)$ such that both primes are less than n occurs at approximately

$$G(n) \sim \frac{n}{\pi(n)}(2\ln(\pi(n)) - \ln(n) + \ln(2c_2))$$

where $\pi(n)$ is the prime counting function, and $c_2 \approx 0.660$ is the twin primes constant.

4.2 Shank's conjecture

Let $p(n)$ be the first prime of a pair of consecutive primes with a gap n. Shank conjectures [13] that

$$p(n) \sim exp(\sqrt{n})$$

where $exp(n) = e^n$. In combination with Wolf's conjecture, Shank's conjecture can be used to find the approximate location of the maximal prime gap found using Wolf's conjecture.

4.3 Mathematical Functions of Note

4.3.1 The Prime Counting Function

The Prime Counting Function[6][12], $\pi(n)$, is a function that gives the number of primes that exist less than or equal to n. It has many approximated formulas, however one of its most significant approximations is stated by the prime counting theorem:

$$\pi(n) \sim li(x)$$

where li(x) is the logarithmic integral function.

4.3.2 The Logarithmic Integral Function

The Logarithmic Integral Function [4][11], $li(x)$, is defined as

$$li(x) = \int_0^x \frac{dt}{\ln t}$$

for all $\{x|R_+^* \backslash 1\}$.

There exists a unique constant, $\mu = 1.4513692348...$ called Soldner's constant [11] where $li(x) = 0$. This allows us to rewrite the function as

$$li(x) = \int_{\mu}^{x} \frac{dt}{\ln t}$$

for $x > \mu$. It is this function that we will apply into our Python code of Wolf's conjecture, instead of the prime counting function, as this still works within out range of numbers, and is a simpler, less programmatically expensive system that produces a very similar result.

4.4 Implementation into code

```
def li(x):
    n=10000
    t=0
    d=0
    dx=(x-2)/n
    for k in range(n):
        d=2+k*dx
        t+=dx*(1/math.log(d))
    return t + 1.05

po = pow(10,x)
G = math.floor(po/li(po)*(2*math.log(li(po))-math.log(po)+0.277))

g = math.floor(math.exp(math.sqrt(G)))
```

To calculate the area of an integral, we know to add numerous slices of minimal width. By assigning the variable n to be 10000, we have now set the number of rectangular slices we will use to calculate this area. We also see that the value dx, which would usually be calculated as simply x/n, has now been changed to $(x-2)/n$. This is done in order to avoid encountering any asymptotes or singularities which would otherwise disrupt the code. Ideally, Soldner's constant would be subtracted, however it would be unnecessarily exact so we round up to 2. Later, within the `for` loop, we see that we add 2 onto the variable d. As can be seen in the line of code following this, we will be taking the log of d, and so we add the 2 back on in order to avoid taking the log of ∞ or 1. The `for` loop allows us to take the sum of the areas of all the slices, and so we end the function by returning this value. We also add on $1.05 \approx li(2)$ to make up for the area lost initially when we subtracted 2 from x in order to avoid asymptotes.

We now translate Wolf's conjecture into code, using the logarithmic integral function as a substitute for the prime counting function. We use 0.277 as an approximation for $ln(2c_2)$, which is acceptable as we are not looking for an exact value from this test. We also apply the floor function to round to a whole number.

Finally we apply Shank's conjecture, and end up with an approximate number to which we can start searching. Because Shank's conjecture is an underestimate, there is no chance of us skipping over a large prime gap.

5 The Final Algorithm

```
import math
def li(x):
    n=10000
    t=0
    d=0
    dx=(x-2)/n
    for k in range(n):
        d=2+k*dx
        s+=dx*(1/math.log(d+dx))
        t+=dx*(1/math.log(d))
    return t + 1.05
a_list = [2,3,5,7,11,13,17,19,23]
def MillerRabin(n):
    q = n-1
    c = 0
    while q&1 != 1:
       q = q>>1
       c += 1
    for a in a_list:
        check = True
        m = pow(a,q,n)
        if m == 1 or m == n-1:
            check = False
        else:
            for _ in range(c):
                m = pow(m,2,n)
                if m == n-1:
                    check = False
            if check:
                return False
    return True
def nextPrime(u):
    y = u
    isPrime1, isPrime2 = False, False
    while y%6 != 0:
        y -= 1
    while not isPrime1 and not isPrime2:
        y+=6
        if y-1 != u:
            isPrime1 = MillerRabin(y-1)
        isPrime2 = MillerRabin(y+1)
    if isPrime1:
        return(y-1,y-1-u)
    return(y+1,y+1-u)
x = 6
po = pow(10,x)
if x == 1:
    prevHigh = 2
    placement = 5
else:
    G = math.floor(po/li(po)*(2*math.log(li(po))-math.log(po)+0.277))
    if G&1 == 1:
        G += 1
    g = math.floor(math.exp(math.sqrt(G)))
    while not MillerRabin(g):
        g += 1
    initPrime = g
    prevHigh = 0
    placement = 0
    while initPrime < po:
        d = nextPrime(initPrime)
        if d[1] > prevHigh:
            prevHigh = d[1]
            placement = d[0]
        initPrime = d[0]
print(prevHigh)
print(placement)
print(placement-prevHigh)
```

5.1 Putting it all together

Displayed at the beginning of Section 5 is the full and final version of our prime gap finding algorithm (an annotated version will be attached on the end of this document). It begins by defining the various functions discussed previously, but as we approach the latter half of the code, we begin to see how these functions are all utilised in conjunction with one another. There is also another function introduced, the `nextPrime(u)` function. This function operates on the idea that every prime is either 1 more or 1 less than a multiple of 6. It takes the number u, and checks numbers 1 greater than or 1 less than all multiples of 6 greater than u. The purpose of this function is simply to output the next prime after u, and so as soon as it finds a prime, it will return the value of that prime, as well as the difference between u and that prime.

The immediate first line of code after defining all of the functions is to choose our x value, that is, the number that defines the range of primes in which to look, 10^x. In this example we have used 10^6, as this is the highest value of x for which the algorithm can run with maximal efficiency. We then set the variable, `po`, to be 10^x. Following this, we take into account the case of $x = 1$, which fails to work with the rest of the code as it is designed to compute larger sets of primes with larger numbers in them. As the $x = 1$ case can be easily calculated, the values are simply assigned, and the remainder of the code is ignored.

With the case of $x = 1$ dealt with, we can now explore the remainder of the code, which begins by applying Wolf's conjecture. We then add 1 to the value G if it is odd, as the prime gap can only be even. With this value, we now apply Shank's conjecture to arrive at an estimated location for the prime at which to start looking.

This can be applied in combination with the function, `nextPrime(u)`, as it offers a lower bound to the prime gap by a considerable margin, and one which has been tested and proven for all $10^x, x < 15$ so that the maximal gap will not occur at a number less than g. However in order for this to work effectively, the input u for the `nextPrime(u)` function should be prime, and so we increase the value of g until we reach a prime number. From here we set the value of `initPrime` to g, and create the variables `prevHigh` and `placement`, which will track the length of the gap itself, and its position.

With the setup complete, the code can now run its course as it continues to loop until it exceeds the upper limit, checking every subsequent prime number following the initial g value. The final output of this algorithm will be the length of the gap, and the pair of consecutive primes that form the gap.

5.2 Results

Our final code was able to produce results up to $x = 9$, however, this took several minutes. $x = 6$ was where the program took less than a second to run.

All results were checked and verified to those given by the Wolfram Mathworld page on prime gaps (excluding $x = 1$, as while Wolfram describes the prime gap beneath a number to be beneath if the beginning of the gap is underneath, the document which outlined problem 5 described the whole gap as being underneath the number, leading to a slight inconsistency in values in that instance). Although we did not manage to make it to $x \geq 10$, the gaps from $x \geq 12$ were achieved by others using more optimised algorithms, more time on their hands, and supercomputers; overall, we did not too badly. Consequentially, the challenge set for the generation of prime gaps from $x > 15$ would be either require immensely different code from what we have currently, or be extraordinarily time consuming.

5.3 Limitations and Further Optimisations

Though significant improvements to the algorithm were made during the process of its construction and development, there still remained some key flaws.

The most outstanding flaw of this code lies with its limited range, as it can only output values up to $x = 8$ within a reasonable time. While this could potentially be increased if it was run on a computer with higher processing power, it would still take far too long to reach any numbers within the range of $15 \leq x \leq 20$. The code itself, while greatly optimised, still likely has room for further improvements and adjustments that could reduce computing time. For example, other similar tests have used some form of Euler's sieve, an old method of producing a list of primes quickly. It has been shown that there are faster prime determining functions available to quantum computers, however, efficient and commercially available ones will likely not be around for some time yet. The mathematics in this area is rich and developing, with world famous mathematicians like Terence Tao being highly involved. We have no doubt that in years to come, there will be further progression in this area, and consequently, further optimisation of the finding of maximal prime gaps.

References

[1] Marek Wolf. "First occurrence of a given gap between consecutive primes". In: (Apr. 1997).

[2] Marek Wolf. "Some Conjectures on the Gaps Between Consecutive Primes". In: (Sept. 1998).

[3] *"Prime gap"*. Wikipedia. URL: `https://en.wikipedia.org/wiki/Prime_gap`.

[4] *Logarithmic integral function*. Wikipedia. URL: `https://en.wikipedia.org/wiki/Logarithmic_integral_function`.

[5] *Miller–Rabin primality test*. Wikipedia. URL: `https://en.wikipedia.org/wiki/Miller-Rabin_primality_test`.

[6] *Prime-counting function*. Wikipedia. URL: https://en.wikipedia.org/wiki/Prime-counting_function.

[7] *Table of Known Maximal Gaps*. PrimePages. URL: https://primes.utm.edu/notes/GapsTable.html.

[8] Terence Tao. *Small and large gaps in the primes*. URL: https://terrytao.files.wordpress.com/2015/07/lat.pdf.

[9] *The Gaps Between Primes*. PrimePages. URL: https://primes.utm.edu/notes/gaps.html?id=research&month=primes&day=notes&year=gaps.

[10] Uni. SoloLearn. URL: https://code.sololearn.com/ciMlRPgrK8DH/#py.

[11] Eric Weisstein. *Logarithmic Integral*. URL: https://mathworld.wolfram.com/LogarithmicIntegral.html.

[12] Eric Weisstein. *Prime Counting Function*. URL: https://mathworld.wolfram.com/PrimeCountingFunction.html.

[13] Eric Weisstein. *Prime Gaps*. URL: https://mathworld.wolfram.com/PrimeGaps.html.

[14] Marek Wolf. URL: http://pracownicy.uksw.edu.pl/mwolf/.

```
#this is a fully documented version of our final code for problem 5, essentially step by step

import math #importing in one of python's inbuilt libraries, in order to avoid outsourcing one

def li(x): #li(x) is the logarithmic integral function, approximately equal to the prime counting function which we will use later
    #we will solve this integral by adding together small rectangles of the area underneath
    n=10000 #this is the number of rectangles we will use
    t=0 #this will be our output
    d=0 #this is the number which will be taken log of
    dx=(x-2)/n #this is the number which will be multiplied by to give the area. note that it is reduced by 2. this is in order to avoid the asymptotes in li(x)
    for k in range(n): #repeating n times...
        d=2+k*dx #defining d based on x. we add the 2 back on here to avoid taking the log of infinity or 1
        t+=dx*(1/math.log(d)) #here we add the area of the rectangle onto t
    return t + 1.05 #once all the rectangles are added on, we have t, then we add on ~li(2) to make up for the area we did not use in order to avoid the asymptotes

a_list = [2,3,5,7,11,13,17,19,23] #here we have the list of numbers used in the Miller-Rabin prime check

def MillerRabin(n): #here we begin to define the Miller-Rabin prime check
    q = n-1 #q becomes one less than n
    c = 0 #set c
    while q&1 != 1: #while the last bit of q is 0
        q = q>>1 #bitshift q by 1 to the right
        c += 1 #each time this happens add 1 to c
        #this produces n-1 = 2^c*q
    for a in a_list: #now all the numbers in a_list are checked
        check = True #used to check if conditions for primality are met
        m = pow(a,q,n) #now we set m to be a^q mod n
        if m == 1 or m == n-1: #if m meets the conditions outlied further on the document
            check = False #check is set to false; it is not composite yet
        else: #in the case that it could be composite
            for _ in range(c): #repeating c times
                m = pow(m,2,n) #m = m^2 mod n
                if m == n-1: #again, m is checked for potential primality, outlined further in the document
                    check = False #check is set to false; it is not composite yet
            if check: #if check was unchanged, this would come into effect
                return False #and return that it was composite
    return True #after all the cycles are done and it was never proven composite, return True

def nextPrime(u): #define a function that spits out the next prime, and the gap between them
    y = u #we set y to be u
    isPrime1, isPrime2 = False, False #both variables used to detect a prime are set to False
    while y%6 != 0: #while y is not divisible by 6
        y -= 1 #subtract 1
        #y is now a lower multiple of 6
    while not isPrime1 and not isPrime2: #while a prime has not been found
        y+=6 #6 is added to y
        if y-1 != u: #as long as y-1 was not the original number
            isPrime1 = MillerRabin(y-1) #check if y-1 is prime
        isPrime2 = MillerRabin(y+1) #check if y-2 is prime
    if isPrime1: #once a prime is found, if isPrime1 was the prime number
        return(y-1,y-1-u) #print the number and the gap
    return(y+1,y+1-u) #otherwise do the same for isPrime2

x = 5 #here we define x, as an example, 5
po = pow(10,x) #here we set 10^x
if x == 1: #because nextPrime(u) is not compatible with small numbers, we have to define x = 1 elsewhere
    prevHigh = 2 #as such we define the numbers that would work for x = 1
    placement = 5
else:
    G = math.floor(po/li(po)*(2*math.log(li(po))-math.log(po)+0.277)) #here we use Wolf's conjecture to estimate the gap of the prime we are looking for
    if G&1 == 1: #if it is an odd number (which a gap cannot be)
        G += 1 #add 1
    g = math.floor(math.exp(math.sqrt(G))) #here we use Shank's Conjecture to estimate the number of which to start on by using
```

```
our estimated prime gap. we are able to combine this with nextPrime(u) because it offers a lower bound to the prime gap by a fair margin, one which has been tested for all 10^x x<15
   while not MillerRabin(g): #in order to use nextPrime(u) effectively, u should be a prime number. here we repeat until g is prime
      g += 1 #add 1 to g
   initPrime = g #here we initialise initPrime to our estimate of g
   prevHigh = 0 #this racks our previous high gap
   placement = 0 #this tracks the number the gap finished at
   while initPrime < po: #while initPrime is less than 10^x
      d = nextPrime(initPrime) #find the next prime after initPrime
      if d[1] > prevHigh: #if the gap between the two primes is now the largest gap
         prevHigh = d[1] #set the new highest gap to prevHigh
         placement = d[0] #get the placement of the gap
      initPrime = d[0] #initPrime is now the new prime

print(prevHigh) #once this is all done, print the highest gap
print(placement) #print the placement
print(placement-prevHigh) #print the placement the gap started at

#done!
```