There really are an infinite number of twin primes,
and other thoughts on the distribution of primes.

A proof by Jared Beal,
Orono Maine
Jared.Beal@maine.edu

Proof 1: There really are an infinite number of twin primes.

1.1) Arrange every integer greater than 5, in a table with 6 columns. In the form

	1	2	3	4	5	6
1	5	6	7	8	9	10
2	11	12	13	14	15	16
3	17	18	19	20	21	22
4	23	24	25	26	27	28
5	29	30	31	32	33	34
6	35	36	37	38	39	40
7	41	42	43	44	45	46
8	47	48	49	50	51	52
9	53	54	55	56	57	58
10	59	60	61	62	63	64
11	65	66	67	68	69	70
12	71	72	73	74	75	76
13	77	78	79	80	81	82
14	83	84	85	86	87	88
15	89	90	91	92	93	94
16	95	96	97	98	99	100
17	101	102	103	104	105	106
18	107	108	109	110	111	112
19	113	114	115	116	117	118
20	119	120	121	122	123	124

Every value in Column 2, 4, and 6 are even, they can’t be prime.
Every value in Column 5 is a multiple of 3, they can’t be prime.
We can see that EVERY value in columns 2,4,5,6 are NOT prime.
We can also see that no multiple of any number in Column 2,4,5,6 will ever be in column 1 or 3.

Every number that could possibly be prime falls in column 1 or 3.

1.2) Now we can isolate column 1 and 3. In the form

	C1	C2
1	5	7
2	11	13
3	17	19
4	23	25
5	29	31
6	35	37
7	41	43
8	47	49
9	53	55
10	59	61
11	65	67
12	71	73
13	77	79
14	83	85
15	89	91
16	95	97
17	101	103
18	107	109
19	113	115
20	119	121

We can see that every new row adds one C1 value and one C2 value. Each new value in C1 is 6 larger than the previous. And every new value in C2 is 6 larger than the previous. We can create an infinite number of rows in this form.

We can also see that every pair of numbers that could possibly be a twin prime pair, is on each row.

Let's create a set of all these possibly twin prime pairs and call it P(), such that P(1) is row 1 - (5,7), and P(2) is row 2 - (11,13), and P(3) is row 3 - (17,19) and so on.

The height of each column C1 and C2 is infinite, so we have an infinite number of rows.
So P() is an infinite set that contains every row, with every number pair that could possibly be a twin prime pair.

1.3) If either C1 of P(1) or C2 of P(1), is NOT prime, then P(1) is not a twin prime pair.
In order for a value in C1 or C2 to NOT be prime, it must be a multiple of a number in C1 or C2.

1.4) For each value in C1 and C2, we must create a function that removes all its multiple.

So for the number 5, we create a removal function R(5) that returns the row number of every row with a multiple of 5 in it.

If we look we can see that this function will be linear.
Pick any multiple of 5, anywhere in C1. Anyone, doesn't matter how high you go.
Let’s say you picked A which has 100 million digits. and its at row Z.

At row Z, C1 is equal to A.
so at row Z+1, C1 is equal to A+6. a multiple of 5, +6 is NOT a multiple of 5
so at row Z+2, C1 is equal to A+12. a multiple of 5, +12 is NOT a multiple of 5
so at row Z+3, C1 is equal to A+18. a multiple of 5, +18 is NOT a multiple of 5
so at row Z+4, C1 is equal to A+24. a multiple of 5, +24 is NOT a multiple of 5
so at row Z+5, C1 is equal to A+30. a multiple of 5+30 IS a multiple of 5
There will be another multiple of 5 in exactly 5 rows from the previous multiple of 5.

The same is true for every other number we need to find and remove the multiples of.

Pick any multiple of 13, anywhere in C1, call it A, and lets say its at row Z.
so at row Z+1, C1 is equal to A+6. a multiple of 13, +6 is NOT a multiple of 13
so at row Z+2, C1 is equal to A+12. a multiple of 13, +12 is NOT a multiple of 13
so at row Z+3, C1 is equal to A+18. a multiple of 13, +18 is NOT a multiple of 13
so at row Z+4, C1 is equal to A+24. a multiple of 13, +24 is NOT a multiple of 13
so at row Z+5, C1 is equal to A+30. a multiple of 13, +30 is NOT a multiple of 13
so at row Z+6, C1 is equal to A+36. a multiple of 13, +36 is NOT a multiple of 13
so at row Z+7, C1 is equal to A+42. a multiple of 13, +42 is NOT a multiple of 13
so at row Z+8, C1 is equal to A+48. a multiple of 13, +48 is NOT a multiple of 13
so at row Z+9, C1 is equal to A+54. a multiple of 13, +54 is NOT a multiple of 13
so at row Z+10, C1 is equal to A+60. a multiple of 13, +60 is NOT a multiple of 13
so at row Z+11, C1 is equal to A+66. a multiple of 13, +66 is NOT a multiple of 13
so at row Z+12, C1 is equal to A+72. a multiple of 13, +72 is NOT a multiple of 13
so at row Z+13, C1 is equal to A+78. a multiple of 13, +78 IS a multiple of 13
There will be another multiple of 13 in exactly 13 rows from the previous multiple of 13.

The removal function for finding all multiples of 5, let's call it R(5). takes the form

	Not prime in C1	Not prime in C2
R(5)	y=5x+1	y=5x-1

Where x is every integer from 1 to Infinity. And every Y value output is a row that contains a number that is NOT prime, meaning that row is NOT a twin prime pair.

If we look at R(5) we see

	X=1	X=2	X=3	X=4	X=5	X=6
R(5) #1	6	11	16	21	26	31
R(5) #2	4	9	14	19	24	29

Which is, for X 1 through 6, we identify that rows 4,6,9,11,14,16,19,21,24,26,29,31 contain a multiple of 5 and can not be a twin prime pair.

1.5) You can continue to make these removal functions for every number in C1 and C2, and they will form the following repeating pattern.

	1st	2nd
R(5)	y=5x+1	y=5x-1
R(7)	y=7x-1	y=7x+1
R(11)	y=11x+2	y=11x-2
R(13)	y=13x-2	y=13x+2
R(17)	y=17x+3	y=17x-3
R(19)	y=19x-3	y=19x+3
R(23)	y=23x+4	y=23x-4
R(25)	y=25x-4	y=25x+4
R(29)	y=29x+5	y=29x-5
R(31)	y=31x-5	y=31x+5
R(35)	y=35x+6	y=35x-6
R(37)	y=37x-6	y=37x+6
R(41)	y=41x+7	y=41x-7
R(43)	y=43x-7	y=43x+7

You can rearrange them to make the pattern more obvious

C1				C2		
R(5)	y=5x+1	y=5x-1		R(7)	y=7x-1	y=7x+1
R(11)	y=11x+2	y=11x-2		R(13)	y=13x-2	y=13x+2
R(17)	y=17x+3	y=17x-3		R(19)	y=19x-3	y=19x+3
R(23)	y=23x+4	y=23x-4		R(25)	y=25x-4	y=25x+4
R(29)	y=29x+5	y=29x-5		R(31)	y=31x-5	y=31x+5
R(35)	y=35x+6	y=35x-6		R(37)	y=37x-6	y=37x+6
R(41)	y=41x+7	y=41x-7		R(43)	y=43x-7	y=43x+7
R(N)	y=NX+Z	y=NX-Z		R(N)	y=NX-Z	y=NX+Z

Where the N value goes up by 6 each time, and the Z value goes up by 1 every time.

You can continue this pattern out to infinity, and create a removal function R(N) for every number in C1 and C2.

Every removal function will identify every row that contains a NOT prime number, and as such is NOT a twin prime pair. Every row that doesn't get removed, must contain two primes, and must be a twin prime pair.

If we take our infinite set of every possibly twin prime pair P(), consisting of all the rows of C1 and C2 out to infinity, and remove the rows identified in EVERY removal function, we are left with the set containing All the Twin Primes.

1.6) By looking at the R() functions we can see that each contains two simple equations.
The 1st function of R(5) will identify at most 1 row out of every 5, as NOT a twin prime row.
The 2nd function of R(5) will identify at most 1 row out of every 5, as NOT a twin prime row.
R(5) will therefore identify (2/5)ths of all rows, as not twin prime rows.

The 1st function of R(7) will identify at most 1 row out of every 7, as NOT a twin prime row.
The 2nd function of R(7) will identify at most 1 row out of every 7, as NOT a twin prime row.
R(7) will therefore identify (2/7)ths of all rows, as not twin prime rows.

The 1st function of R(11) will identify at most 1 row out of every 11, as NOT a twin prime row.
The 2nd function of R(11) will identify at most 1 row out of every 11, as NOT a twin prime row.
R(7) will therefore identify (2/11)ths of all rows, as not twin prime rows.

Every R(N) will therefore identify (2/N)ths of all rows, as not twin prime rows.

If we look at these functions and the Y values they return, we can see another pattern.

	x=1		x=2	
R(5)	4	6	9	11
R(7)	6	8	13	15
R(11)	9	13	20	24
R(13)	11	15	24	28
R(17)	14	20	31	37
R(19)	16	22	35	41
R(23)	19	27	42	50
R(25)	21	29	46	54
R(29)	24	34	53	63
R(31)	26	36	57	67
R(35)	29	41	64	76
R(37)	31	43	68	80
R(41)	34	48	75	89
R(43)	36	50	79	93

	x=1	small gap		large gap	x=2	small gap	
R(5)	4	1	6	2	9	1	11
R(7)	6	1	8	4	13	1	15
R(11)	9	3	13	6	20	3	24
R(13)	11	3	15	8	24	3	28
R(17)	14	5	20	10	31	5	37
R(19)	16	5	22	12	35	5	41
R(23)	19	7	27	14	42	7	50
R(25)	21	7	29	16	46	7	54
R(29)	24	9	34	18	53	9	63
R(31)	26	9	36	20	57	9	67
R(35)	29	11	41	22	64	11	76
R(37)	31	11	43	24	68	11	80
R(41)	34	13	48	26	75	13	89
R(43)	36	13	50	28	79	13	93

R(5) has a Not twin prime row (4), a small gap of 1 row (5), a NOT twin prime row (6), a large gap of 2 rows (7 and 8), and then repeats.

R(7) has a Not twin prime row, a small gap of 1 row, a NOT twin prime row, a large gap of 4 rows and then repeats

Every new removal function has a repeating pattern of small and large gaps.
The small gap size goes up by two every row, with Row 1 (5,7) having smalls gaps of size 1, Row 2 (11,13) having small gaps of size 3, and so on.
The large gap sizes go up by 2 for every new removal function.

1.7) Now for every R(N) imagine a wheel with N spokes. ***also see the “wheel notation of prime numbers” in Proof 2 to better visualize how these wheels work***

Now paint 2 red spokes on each wheel red, and the rest green. Arrange the sequence of red to green to correspond to the gaps identified above.

R(5) - will be a wheel with 5 spokes, 1 is red, 2 is green, 3 is red, 4 and 5 are green.
R(7) - will be a wheel with 7 spokes, 1 is red, 2 is green, 3 is red, 4, 5, 6, 7 are green.
R(11) - will be a wheel with 11 spokes, 1 is red, 2,3,4 are green, 5 is red, 6,7,8,9,10,11 are green.

This creates wheels in the form of our gap patterns.

Now if we place wheel R(5) on a table by itself, with any spoke pointing north.
We then rotate this wheel 1 spoke clockwise. This is equivalent to moving 1 row down. If we rotate this wheel 5 times it comes back to its original orientation. This is equivalent to going 5 rows down. This process of wheel rotations is going to identify the same rows as the R(5) removal function equation.

Every time a red spoke points north we know we have removed 1 possible twin prime row. So in 5 rotations, we have had 2 red spokes point north, and 3 green spikes point north. So in any 5 consecutive rows of P(), R(5) will remove at most 2/5ths of those rows as NOT twin prime rows, and at least 3/5ths will remain as possible twin prime rows. Over an infinite number of rows, R(5) will remove 2/5ths of them as not Twin prime rows, and 3/5ths will remain.

If we now place wheel R(7) on the table with R(5), also with 1 spoke pointing north. We now rotate each wheel 1 spoke clockwise. This is equivalent to moving 1 row down. R(5) will make a full revolution every 5 rotations, and R(7) will make a full revolution every 7 rotations. If we do 5*7 rotations, R(5) will revolve 7 times, and R(7) will revolve 5 times. This will give us every possible combination of R(5) and R(7) spokes pointing north. Of these 35 combinations, 20 instances will show at least one red spoke facing north, and 15 will show 2 green spokes facing north. So in any 35 consecutive rows of P(), R(5) and R(7) will remove at most 20/35ths rows as NOT twin prime rows, and at least 15/35ths will remain. 15/35th is the same as 3/5th times 5/7ths. Over an infinite number of rows, (3/5 * 5/7)ths of those rows will remain after R(5) and R(7).

If we now place wheel R(11) on the table with R(5) and R(7), also with 1 spoke pointing north. We now rotate each wheel 1 spoke clockwise. This is equivalent to moving 1 row down. R(5) will make a full revolution every 5 rotations, and R(7) will make a full revolution every 7 rotations. R(11) will make a full revolution every 11 rotations. If we do 5*7*11 rotations, R(5) will revolve 77 times, and R(7) will revolve 55 times, and R(11) will revolve 35 times. This will give us every possible combination of R(5), R(7), and R(11) spokes pointing north. Of these 385 combinations, 250 instances will show at least one red spoke facing north, and 135 will show 3 green spokes facing north. So in any 385 consecutive rows of P(), R(5), R(7), and R(11) will remove at most 250/385ths rows as NOT twin prime rows and at least 135/385ths will remain. 135/385ths is the same as 3/5th times 5/7ths times 9/11ths. Over an infinite number of rows, (3/5 * 5/7 * 9/11)ths of those rows will remain after R(5) and R(7) and R(11).

Adding more wheels, equivalent to adding more R() functions, will follow the same pattern with each new R(N) wheel changing the the ratio of twin primes to not twin primes in an infinite number of rows of P(), by (N-2)/N. Every wheel is removing a smaller and smaller portion of the rows.

If we add an infinite number of R() function wheels, one for every value in C1 and C2 we will then come to a formula that looks like.
(3/5)*(5/7)*(9/11)*(11/13)*(15/17)*(17/19)......*(Infinity-2)/infinity)
This value, lets call it Z, is the ratio of twin primes to not twin prime rows in P(). Z will always be greater than 0.

P() is an infinitely large set of possibly twin prime rows. Z is the ratio of twin prime rows to not twin prime rows in P(). So P() * Z gives you the size of the set of ALL Twin primes. P()*Z is infinitely large, so the set of ALL Twin primes is infinitely large.

Proof 2:
<u>Primes are NOT random, an algorithm for finding every prime number, and how they are distributed.</u>

2.1) If you take the 1st function of every removal function of every number in C1 and C2, Each of these removal functions will return the Y value (Row #), of every C1 value that is a multiple of a number in C1 or C2, and as such is NOT prime.

<u>For C1, the functions would look like this,</u>

For every Function 1 of every removal function,

R(5)	y=5x+1
R(11)	y=11x+2
R(17)	y=17x+3
R(23)	y=23x+4
R(29)	y=29x+5
R(35)	y=35x+6
R(41)	y=41x+7
[redacted]	
R(N)	y=NX+Z

R(7)	y=7x-1
R(13)	y=13x-2
R(19)	y=19x-3
R(25)	y=25x-4
R(31)	y=31x-5
R(37)	y=37x-6
R(43)	y=43x-7
[redacted]	
R(N)	y=NX-Z

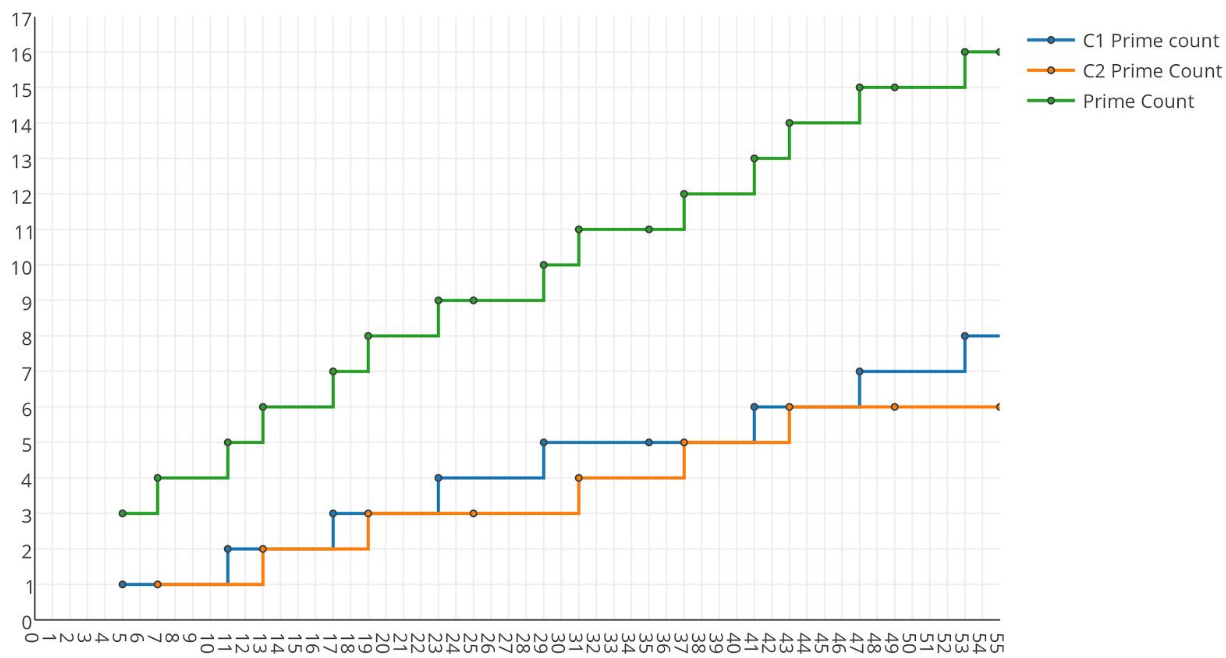
Where N goes up by 6 every time, and Z goes up by 1 every time.
And X is every integer from 1 to infinity.

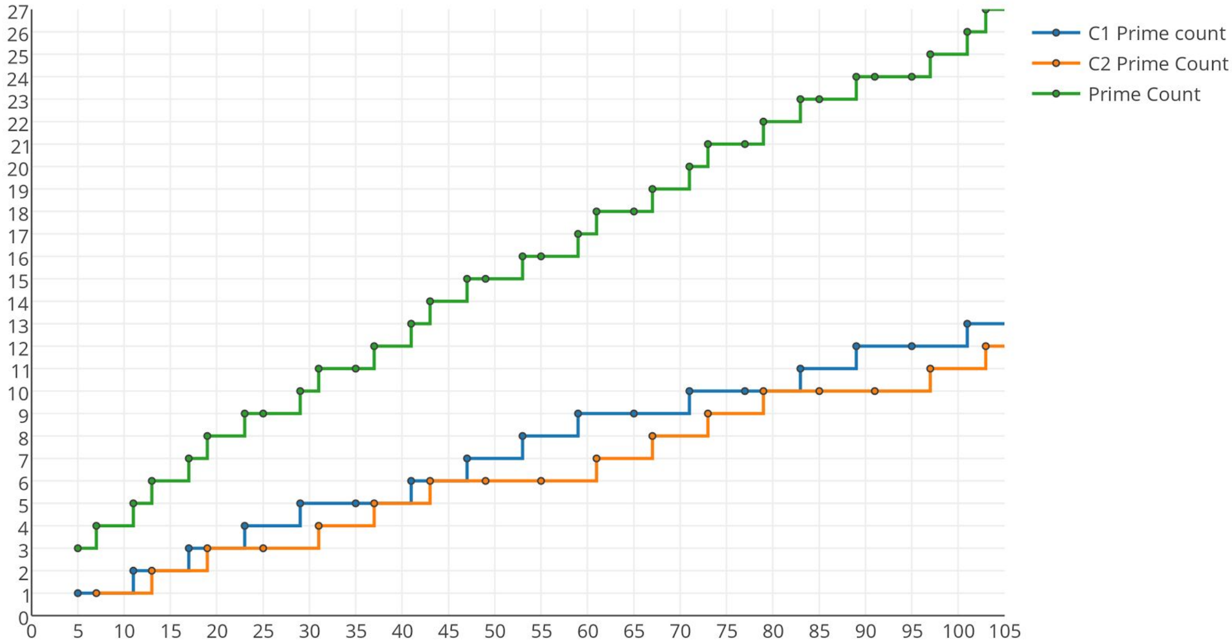
If you run the 1st function of every removal function for every number in C1 and C2, for an infinite x, you will remove every NOT prime number from C1.
The numbers left in C1 will all be prime.

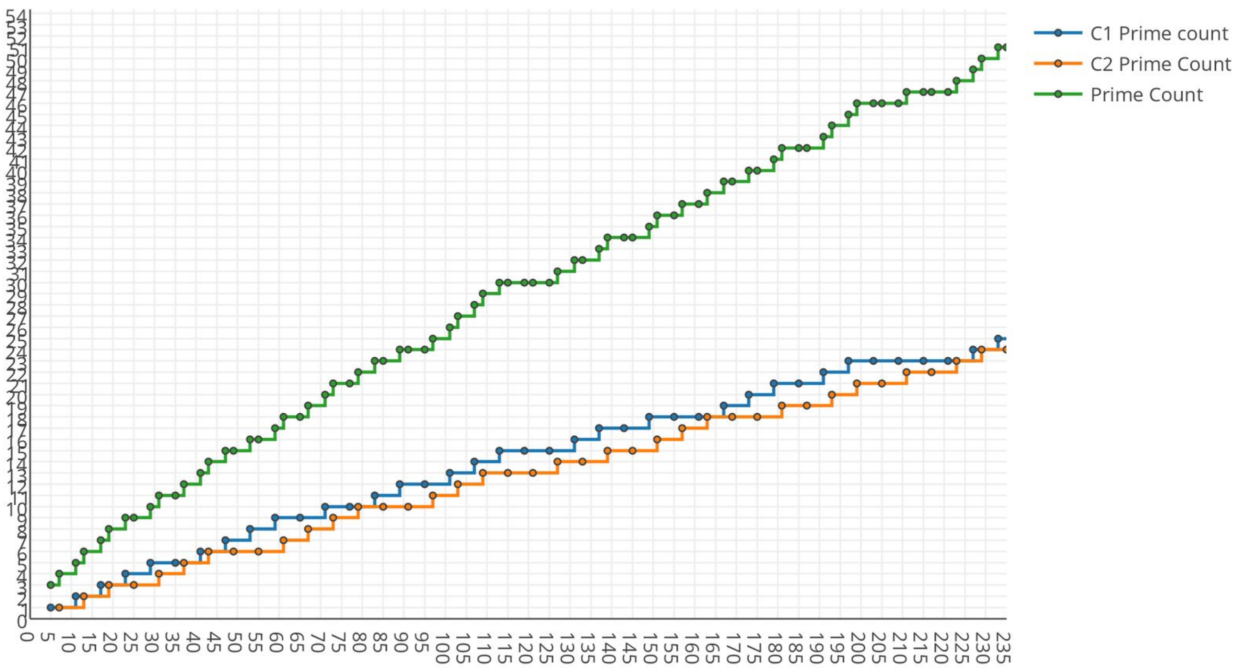
You can do the same with the 2nd function of every removal function for every number in C1 and C2, for an infinite x, and you will remove every NOT prime number in C2.
The numbers left in C2 will all be prime.

The numbers left over in C1 and those left over in C2, will be the set of ALL prime numbers.

2.2) These removal functions are what determine the distribution of prime numbers. If we look at the prime number step function, we can see that it is the sum of the primes in C1 + the primes in C2, plus 2 for the primes 2 and 3 which are not found in C1 or C2. Every time there is a “rise” in the prime step function, it is because of a rise in C1 or C2’s step function.







2.3) Going by the above proof 1 analogy to wheels, we can more easily see how this is working, and find every prime number.

Create a “removal wheel” for each R(N), with N spokes, with 1 painted red and the rest green. This algorithm starts with a blank table, and you then add removal wheels to it.

Row 1 - No wheels. Add wheel R(5), with the red spoke pointing north. Add wheel R(7),with the red spoke 2 to the right of north.

Row 2 - Rotate all wheels 1 spoke clockwise. If all spokes pointing north are green, Row 2 of C1 is Prime. Add wheel R(11), with the red spoke pointing north. Add wheel R(13),with the red spoke 4 to the right of north.

Row 3 - Rotate all wheels 1 spoke clockwise. If all spokes pointing north are green, Row 3 of C1 is Prime. Add wheel R(17), with the red spoke pointing north. Add wheel R(19),with the red spoke 6 to the right of north.

Row 4 -Rotate all wheels 1 spoke clockwise. If all spokes pointing north are green, Row 4 of C1 is Prime. Add wheel R(23), with the red spoke pointing north. Add wheel R(25),with the red spoke 8 to the right of north.

Row 5 - Rotate all wheels 1 spoke clockwise. If all spokes pointing north are green, Row 5 of C1 is Prime. Add wheel R(29), with the red spoke pointing north. Add wheel R(31),with the red spoke 10 to the right of north.

You can continue doing this for every row, with two new wheels being added at each row corresponding to the C1 and C2 values in that row. The Wheel for the C1 value in that row will be placed with the red spoke pointing north, and the wheel from C2 of the row having its red spoke pointing (2*row #) spokes to the right…..

If you continue this algorithm out to infinity you will find every prime number in C1.

<u>For C2, the function would look like this,</u>
You can can do the same things as above for every row, with two new wheels being added at each row corresponding to the C1 and C2 values in that row. The Wheel for the C2 value in that row will be placed with the red spoke pointing north, and the wheel from C1 of the row having its red spoke pointing (N-(2*row #)) spokes to the left of north…..

If you continue this algorithm out to infinity you will find every prime number in C2.

You can make this wheel algorithm more obvious using what I call the “wheel notation of prime numbers”. Where we label each spoke of each wheel with N spokes 1-N. The red spoke will be labeled 1, and all the green spokes will be labeled 2 through N. So for C1 we already know how to place our wheels from above, The Wheel for the C1 value in that row will be placed with the red spoke pointing north, and the wheel from C2 of the row having its red spoke pointing (2*row #) spokes to the right.

For C1 this would give us a notation where we have a “digit” for each removal function wheel, that looks like this.

Row #	C1	5	7	11	13	17	19	23	25	29	31	35	37	41	43	47	49	53	55	59	61	65	67
1	5	1	3																				
2	11	2	4	1	5																		
3	17	3	5	2	6	1	7																
4	23	4	6	3	7	2	8	1	9														
5	29	5	7	4	8	3	9	2	10	1	11												
6	35	1	1	5	9	4	10	3	11	2	12	1	13										
7	41	2	2	6	10	5	11	4	12	3	13	2	14	1	15								
8	47	3	3	7	11	6	12	5	13	4	14	3	15	2	16	1	17						
9	53	4	4	8	12	7	13	6	14	5	15	4	16	3	17	2	18	1	19				
10	59	5	5	9	13	8	14	7	15	6	16	5	17	4	18	3	19	2	20	1	21		
11	65	1	6	10	1	9	15	8	16	7	17	6	18	5	19	4	20	3	21	2	22	1	23

For C2 the notation would look like this.

Row #	C2	5	7	11	13	17	19	23	25	29	31	35	37	41	43	47	49	53	55	59	61	65	67
1	7	3	1																				
2	13	4	2	5	1																		
3	19	5	3	6	2	7	1																
4	25	1	4	7	3	8	2	9	1														
5	31	2	5	8	4	9	3	10	2	11	1												
6	37	3	6	9	5	10	4	11	3	12	2	13	1										
7	43	4	7	10	6	11	5	12	4	13	3	14	2	15	1								
8	49	5	1	11	7	12	6	13	5	14	4	15	3	16	2	17	1						
9	55	1	2	1	8	13	7	14	6	15	5	16	4	17	3	18	2	19	1				
10	61	2	3	2	9	14	8	15	7	16	6	17	5	18	4	19	3	20	2	21	1		
11	67	3	4	3	10	15	9	16	8	17	7	18	6	19	5	20	4	21	3	22	2	23	1

You can see the number of “digits” increase by 2 every time because we are adding two new wheels each row.

The R(5) wheel starts at spoke 1, moves to spoke 2, then 3 and so on. Every 5 rows it has 1 red spoke. Whenever a wheel comes back around to 1, red spoke, we know we have hit a NOT PRIME number. The same is true for every other wheel. When it comes around to 1, we have hit a NOT prime number. So the only time we will have a prime number, is when all “spoke values” go up by 1 without “coming around”. This happens only when every spoke pointing north is green.

2.4) We can see that for C1, R(5) will remove 1 C1 value as not prime for every 5 rows. R(7) will remove 1 C1 value as not prime for every 7 rows, and so on. R(1) removes 1/5th of rows, so 4/5ths remain. R(7) removes 1/7th of rows so 6/7ths remain. Each removal function R(N) is removing (1/N)th of rows from C1 as NOT prime, and ((N-1)/N) will remain.

If you take infinite set of rows for C1, and run all the removal functions for every value in C1 and C2, the ratio of primes to rows would be (4/5)(6/7)(10/11)(12/13)(16/17)......(Infinity-1)/infinity)

If you take infinite set of rows for C2, and run all the removal functions for every value in C1 and C2, the ratio of primes to rows would be (4/5)(6/7)(10/11)(12/13)(16/17)......(Infinity-1)/infinity)

The ratio of primes to rows, after running all the removal functions, is the same in C1 as it is in C2, (4/5)(6/7)(10/11)(12/13)(16/17)......(Infinity-1)/infinity). This means that, for an infinitely large set of C1 and C2 values, the number of prime numbers in C1 is equal to the number of primes in C2.

We can look at this in terms of integers instead of rows as well.
If we have X integers it would mean X/6 rows.
The ratio of primes to not primes in an infinite number of rows is
((4/5)(6/7)(10/11)(12/13)(16/17)......(Infinity-1)/infinity) in C1 and in C2

So the ratio of primes to not primes in a set of X integers for infinite X is,
(X/6) * ((4/5)(6/7)(10/11)(12/13)(16/17)......(Infinity-1)/infinity))* 2
Where X/6 is giving us the number of rows, times our prime to rows ratio, times 2 because we have to do it for C1 and C2, Which you can simplify to,

The number of primes in an infinite set of X integers, is
(X) * (2/6) * ((4/5)(6/7)(10/11)(12/13)(16/17)......(Infinity-1)/infinity)

This works on an infinite set of X integers, but breaks down on a finite set because the ratio of primes to rows is dependent on the number of wheels. For values of X less than infinity you would need to take into account only the removal function wheels that could remove a value less than X, and use only those wheels to find the ratio of primes to not primes. On a small enough scale you would also have to take into account that C1 starts at integer 5, so you would have to add primes 2 and 3 to the total as they don’t appear in C1 or C2.

(X) * (2/6) * ((4/5)(6/7)(10/11)(12/13)(16/17)......(Infinity-1)/infinity), will however, always be the MINIMUM number of primes in a set of X integers. As X gets closer to infinity it will become a more accurate representation of the distribution of prime numbers.

2.5) Since the number of primes in C1 is equal to the number of primes in C2, there must be a 1 to 1 relationship between every prime number in C1, to a prime in C2. Every number in C1 must have what I call a "companion prime" in C2. The first few companion primes are circled below. (5,7) (11,13) (17,19) (23,31) (29,37) (41,43)