TCU

COLLEGE OF SCIENCE & ENGINEERING

DEPARTMENT OF MATHEMATICS

Probabilities on Latin Squares

Anna Long, Advisor: Dr. Drew Tomlin

Department of Mathematics, Texas Christian University

Abstract

A Latin square is an *nxn* square that contains *n* different symbols, often numbers that are arranged so that each symbol appears exactly once in each row and column. In this project, we look at the probability of a random arrangement of symbols being a Latin square. I start with n number of n symbols, for example a 3x3 square will contain the numbers 1,1,1,2,2,2,3,3,3 in a random assortment. Using counting methods and statistical estimation through Python, we discover the proportion of Latin squares to total squares.

Example Latin Square

The following is a valid 4x4 Latin square:

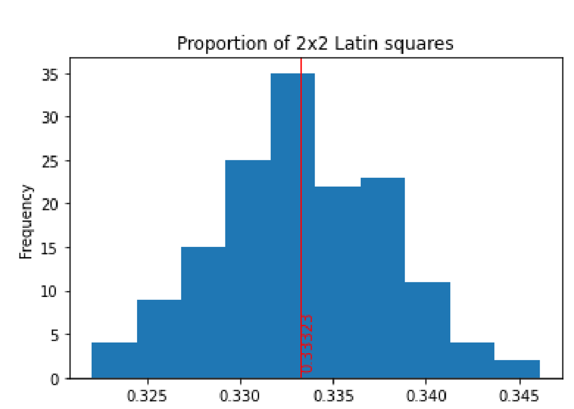
1	2	4	3
4	3	1	2
3	4	2	1
2	1	3	4

Estimation with Python

In order to check my calculations, I used Python to simulate random squares of sizes 2x2, 3x3, and 4x4. Due to limitations with technology, 4x4 was the largest size square that I attempted to create in Python. By listing the valid numbers that would be used in a square, randomizing, and reordering into a square shape, I created random squares that I could test for the quality that would make it a Latin square. In order to check for uniqueness of numbers in each row and column, I used lists and sets. In python, a list can contain any values, whereas a set can only contain unique values. With this information, I turned each row and column of the squares into a list and a set and then compared the lengths of each. If the list and the set had the same number of elements, then I knew that that row or column contained only unique values. If every row and column passed this test, then I knew the square must be a Latin square. Counting the squares that met the criteria and the total randomly generated squares, I created the proportion of Latin squares to total squares for the 3 different sizes of squares. By simulating this process hundreds of times, I visualized this estimation through histograms with a mean that is approximately equal to the calculated proportion.

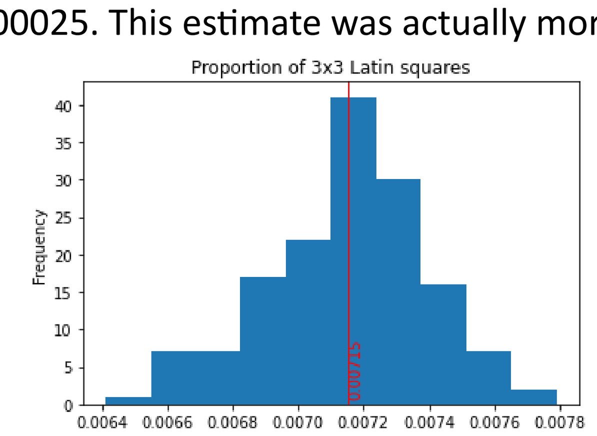
2x2 Latin Squares

The 2x2 case of Latin squares is the simplest variety of Latin Squares. There are only two possible variations of Latin squares and six possible variations of squares with the given symbols; an overall proportion of 1/3 or 0.33333. My code was able to estimate this proportion as 0.33323 with a standard deviation of 0.00461.



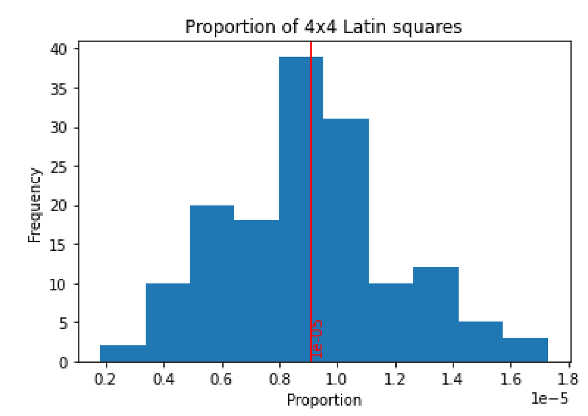
3x3 Latin Squares

Using the same technique, I discovered the estimated proportion of 3x3 Latin squares to be 0.00715 with a standard deviation of 0.00025. This estimate was actually more precise than the 2x2 estimation.



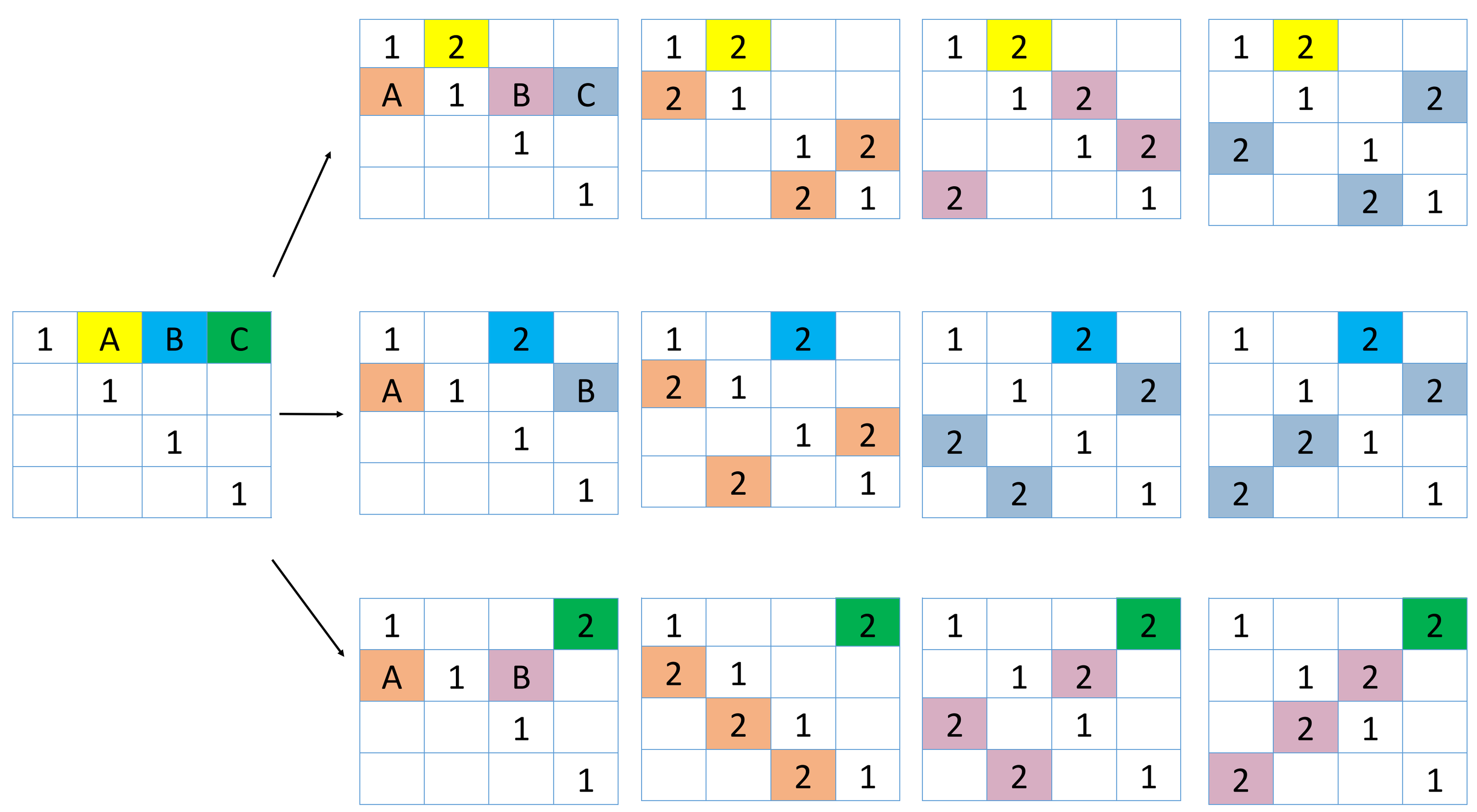
4x4 Latin Squares

I discovered the estimated proportion of 4x4 Latin squares to be 9.06×10^{-6} with a standard deviation of 2.9×10^{-6}. This estimate was again more precise than the previous estimates, due to the fact that the average is a much smaller proportion than the smaller squares. The average printed on the graph is rounded to 5 decimal places, therefore it rounded up to 1×10^{-5}.



Counting Techniques

In order to count the Latin squares of various sizes by hand, I counted the different patterns that were possible. Beginning with the small squares and then working my way to larger and larger squares, I counted the patterns of 1's then the patterns of 2's and so on. Below is an example of the counting of the 4x4 patterns of 2's.



Patterns of 1's

The count of patterns of 1's is simply $n!$ or $n(n-1)(n-2)\ldots(2)(1)$. I began with a 1 in the top right corner of every square to simplify my counting, but this does not change my proportion due to symmetry. For example, you take a Latin square with a specific pattern of 1's with a 1 in the top left corner, and swap all the 1's and 2's in that square and you essentially have the same Latin square, and so on by swapping all possible values. Therefore, I'm only working with $(n-1)!$ patterns of 1's for each size square, which is significantly less than $n!$ patterns of 1's when dealing with larger sizes.

Patterns of 2's

Counting the patterns of 2's consisted of drawing a bunch of examples of various sizes and looking for a relationship between the values. I was again able to use symmetry to my advantage for the larger squares and begin with a 2 placed in the top row then multiplying by $(n-1)$ at the end to account for the choices of that first placement. I finally noticed a pattern after finding the patterns of 2's for the Latin squares up to 6x6. I discovered that by adding the previous 2 counts of patterns of 2's and then multiplying by $(n-1)$ I would get the count of patterns of 2's for each pattern of 1's for the current sized square.

Equations

Total random *nxn* squares:

$$n\prod_{i=0}^{n-2}\binom{n^2-in-1}{n}$$

Patterns of 1's:

$$n!$$

Patterns of 2's:

$$t_2 = 1 \qquad t_3 = 2$$

$$t_n : n \geq 4$$

$$t_n = (n-1)(t_{n-2} + t_{n-1})$$

Conclusion

There are 6 possible 2x2 configurations with 2 of those being Latin squares with two different patterns of 1's. For 3x3 squares, the random configurations go up to 1,680 with only 12 of these being Latin squares, consisting of 2 patterns of 2's for each of the 6 patterns of 1's. There are 576 different 4x4 Latin squares out of the 63,063,000 random squares that are made up by 24 patterns of 1's, 9 patterns of 2's for each pattern of 1's, and 1/3 of those resulting in 4 patterns of 3's and the remaining 2/3 resulting in 2 patterns of 3's. There are 623,360,750,000,000 random 5x5 squares with 161,280 possible Latin squares with 120 patterns of 1's, 44 patterns of 2's for each, 5/11 of those patterns of 2's result in 12 patterns of 3's and 6/11 result in 13. When the pattern of 2's results in 12 patterns of 3's, then each of those make 2 Latin squares, but the ones that create 13 patterns of 3's then 5/13 make 4 Latin squares and the remaining ones make 2. The following table presents the same information where P1 refers to the patterns of 1's, etc. and LS refers to Latin squares.

n	P1	P2	P3	P4	P5	LS	Total
2	2	1	-	-	-	2	6
3	6	2	1	-	-	12	1,680
4	24	9	$\frac{2}{3}(2)+\frac{1}{3}(4)$	1	-	576	63 Million
5	120	44	$\frac{5}{11}(12)+\frac{6}{11}(13)$	$\frac{336}{11}$	1	161,280	623 Trillion

I noticed that beginning with the 4x4 Latin squares, two rows and two columns could pair up for the pattern of $(n-2)$'s, and this is what created the 4 resulting Latin squares rather than 2. Bringing this into higher dimensions would allow for even more resulting Latin squares from each pattern of the $(n-2)$ number. A 6x6 Latin square can have 3 of these pairings, so some patterns of 4's could create 8, or 2^3 Latin squares from a single pattern. Generally, the largest number of Latin squares that could arise from a single pattern of $(n-2)$'s would be the floor function of $n/2$ as an exponent of base 2.

Future Work

The next steps in this research would be finding the generic formula for counting the patterns of 3's for an *nxn* Latin square. I would continue looking at these patterns until a generic formula for counting Latin squares of a certain size is found. I believe that looking at the patterns within Latin squares will tell a lot about how they are formed, such as the pairing technique I mentioned. I ran into problems with computing power due to the large number of squares I had to generate due to the proportions being so small. Given more computing power, I would also estimate these proportions of larger Latin squares through statistical estimation on Python. I might also create a program that can look at the patterns within the Latin squares so that trends might be able to be drawn for certain patterns.