

Chapter 1: Introduction to Discrete Markov Chain Monte Carlo

1.1 Overview

Although our topic belongs to mathematics, it has a number of applications, and so, to be concrete, I begin this description with an example of an application.

Applied problem 1: Galapagos Finches. Community ecologists study data like the matrix in Display 1.2, which shows the distribution of 13 species of finches among the 17 islands of the Galapagos chain. For simplicity's sake, consider here just a tiny version of the system, with only 3 species and 3 islands, distributed as follows:

Species	Island			Total
	1	2	3	
A	1	0	1	2
B	0	1	0	1
C	1	0	1	2
Total	2	1	2	5

Display 1.1 A simple co-occurrence matrix.

In the table, 1 records that a species is present, and 0 records absence. Thus Species A occurs on islands 1 and 3, but not on 2.¹

Notice that Species A and B never appear together on the same island: they form what island bio-geographers call a “checkerboard,” a species pair with no islands in common. Counting the number of checkerboards in a species-by-island matrix gives one way to measure the degree of competition between species. In the example, there are three possible species pairs, forming a total of 2 checkerboards (pairs A,B and B,C). In the finch data in Display 1.2, there are 78 species pairs, and 10 checkerboards.

How surprised should an ecologist be to find 10 checkerboards among 78 species pairs? Is 10 large enough to suggest that some biological cause is at work, or is 10 the sort of value you could easily get if the finch species has distributed themselves at random?

A mathematical question. Various versions of this question have been the subject of research by ecologists, statisticians, and mathematicians for more than 20 years. For the simple version with 3 species and 3 islands, there is a simple solution. If you write out all the tables of 0s and 1s that have the same row and column sums as the table above, you find there are only 5 of them. Of these 5, only the one shown above has 2 checkerboards; the other four tables have only 1 each. So if the species had distributed themselves at

¹ Although I've used an ecology context, with species as rows and islands as columns, there are many other applications outside of ecology. In educational testing, for example, rows could be test-takers, and columns could be multiple choice items, with a 1 to indicate a correct answer. In computer vision, the array of 1s and 0s might correspond to pixels.

random, all five tables would be equally likely, and the chance of 2 or more checkerboards would be $1/5$.

For any but the smallest tables, the questions turn out to be surprisingly difficult. For example, no one knows how many different versions of the finch data table there are with the same row and column sums, although researchers at both Harvard and Stanford are working on it. Current best estimates are in the neighborhood of 6.715×10^{17} tables. For other applications (educational testing or computer vision, for example), the data tables would have many more than 13 rows, many more than 17 columns, and the total number of tables would be astronomical.

With such a gigantic number of tables, trying to list them all is hopeless. Instead, ecologists have tried to generate random matrices through a long series of small steps. For example, one method still in use was first published by the ecologists Connor and Simberloff:² In the table above, there are four different sub-tables of the form

$$\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array} \quad \text{or} \quad \begin{array}{cc} 0 & 1 \\ 1 & 0 \end{array}$$

If you pick one of these four sub-tables at random, and swap the 0s and 1s, you create a new sub-table, and thus a new 3×3 data table, with the same row and column sums as the original. Similarly, with the finch data, you can create new tables by picking a pair of rows and a pair of columns at random to get a 2×2 sub-table; if your sub-table has one of the two diagonal forms above, you swap the 1s and 0s.

This method is an instance of Markov Chain Monte Carlo, an extremely general and powerful method of computer simulation that has become one of the most active research topics in statistics in the past decade or so. The sequence of random swaps defines a random path through the collection of all tables with the given row and column sums. Random paths of this sort have analogues in physics (Brownian motion) and molecular biology (3-dimensional structure of proteins). Trying to understand the mathematics of this method for generating random paths raises lots of questions: Does the method of random swaps give the right probability distribution? (No.) Can it be altered so that it does? (Yes.) How efficient is the method, i.e., how many random swaps do you need? (An area of ongoing research.) Can it be made more efficient, i.e., is there a way to get equally good results with far fewer swaps? (An area of ongoing research.)

This area of study offers many advantages as a topic for undergraduates. It is a topic under active study by statisticians and mathematicians, with many open questions. Some problems are simple enough that you can understand them and stand a good chance of making progress toward an answer without having taken a lot of advanced mathematics courses. At the same time, the study of these questions brings together very different

² E. F. Connor and D. Simberloff (1979). "The assembly of species communities: chance or competition?" *Ecology* **60**: 1132-40.

Galapagos Finches: Distribution of 13 finch species A - M among 17 islands

Species	Sites																	Total
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
A	0	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	14
B	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	0	0	13
C	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0	14
D	0	0	1	1	1	0	0	1	0	1	0	1	1	0	1	1	1	10
E	1	1	1	0	1	1	1	1	1	1	0	1	0	1	1	0	0	12
F	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	2
G	0	0	1	1	1	1	1	1	1	0	0	1	0	1	1	0	0	10
H	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1
I	0	0	1	1	1	1	1	1	1	1	0	1	0	0	1	0	0	10
J	0	0	1	1	1	1	1	1	1	1	0	1	0	1	1	0	0	11
K	0	0	1	1	1	0	1	1	0	1	0	0	0	0	0	0	0	6
L	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2
M	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	17
Total	4	4	11	10	10	8	9	10	8	9	3	10	4	7	9	3	3	122

The same data, with rows and columns in descending order of row and column totals

Species	Sites																	Total
	3	4	5	8	12	7	10	15	6	9	14	1	2	13	11	16	17	
M	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	17
A	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0	1	1	14
C	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	0	14
B	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	13
E	1	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	12
J	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	11
D	1	1	1	1	1	0	1	1	0	0	0	0	0	1	0	1	1	10
G	1	1	1	1	1	1	0	1	1	1	1	0	0	0	0	0	0	10
I	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	10
K	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	6
F	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	2
L	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
H	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1
Total	11	10	10	10	10	9	9	9	8	8	7	4	4	4	3	3	3	122

Density of 1s = 0.552

Checkerboards = 10 # species pairs = 78 Fraction = 0.128

A "checkerboard" is a species pair with no island inhabited by both.

2x2 sub-matrices = 10608 # CUs = 333 Fraction = 0.031

A "checkerboard unit" (CU) is a combination of species pair and island pair having the first species but not the second on one island, and the second species but not the first on the other island.

Data from <http://www.amsci.org/articles/00articles/sandersoncap3.html>

Display 1.2 Galapagos finch data

Rows correspond to species A – M; columns to islands 1 – 17.

A 1 indicates that the species is present on the island.

areas of mathematics, making connections that can be quite surprising and delightful. Among the areas of mathematics are probability, graph theory, combinatorics, linear algebra, and abstract algebra.

Discussion Questions: Random swaps for the finch data

1. Use the bottom version of the data matrix in Display 1.2.
 - a. Explain why the 2×2 sub-matrix in the lower left corner (rows L&H, columns 3&4) is not swappable.
 - b. Find a swappable 2×2 sub-matrix whose rows are next to each other in the bottom matrix, and whose columns are also physically adjacent.
 - c. Now find a 2×2 sub-matrix for which either rows or columns are not adjacent.
 - d. Find a swappable 2×2 sub-matrix for which the swap does not change the total number of checkerboards. (A checkerboard is a pair of species that never occur together.)
 - e. Now find a swappable 2×2 sub-matrix for which the swap does change the total number of checkerboards.

2. Some questions to keep in mind. Answer them now as best you can, and plan to revisit them later on.
 - a. If you swap one 2×2 sub-matrix, what fraction of the entries in the big matrix have you altered?
 - b. Based on (a), how many random 2×2 swaps would you say are needed before you would consider the resulting matrix “random”?
 - c. For the finch data, if you pick two rows at random, then pick two columns at random, what is the probability that the resulting 2×2 sub-matrix will be swappable? (In other words, what fraction of all possible 2×2 sub-matrices are swappable?)
 - d. Based on (c), how many random 2×2 s will you have to create, on average, until you get one that is swappable?
 - e. Based on (b) and (d), how many random 2×2 s will it take to produce a matrix that you consider “random”?

Activity: Random Swaps for the mini-finches

Each step (iteration) of the Connor/Simberloff method for creating a random matrix has two parts:

1. *Random*: Choose two rows of the matrix at random³; choose two columns at random.
2. *Swap*: Is the resulting 2×2 matrix swappable?
 - a. No. Return to (1). Do not count this as a step.
 - b. Yes. Make the swap.

³ “Random” here means **uniform**: all choices must be equally likely.

Refer to the scaled down data matrix from Display 1.1. Here is a way to choose random 2×2 sub-matrices using a table of random numbers:

- *Rows.* There are three pairs of rows: 1&2, 1&3, 2&3.

Assign random numbers to each pair:

Pair 1&2: 0, 1, 2 *Pair 1&3:* 3, 4, 5 *Pair 2&3:* 6, 7, 8

Draw a random digit from a table of random digits (e.g., Appendix 1.2) to get a random pair. If you get a 9, draw again.

- *Columns.* Use the same method (but with a different random number!) to get a random pair of columns.

Using this method, carry out a few steps of the Connor/Simberloff method: Choose a random 2×2 matrix; if it is swappable, swap; otherwise try again.

1. List all the different 3×3 matrices created by the class, together with their frequencies.
2. List all five of the 3×3 matrices of 0s and 1s with row sums 2, 1, 2 and column sums 2, 1, 2. Suppose you use the C/S method to generate random matrices. Would you predict that over the long run their method will produce the five different matrices with equal frequencies?
3. Label the five matrices a, b, c, d, e . Divide up the matrices among the members of the class. For each matrix, find all the swappable 2×2 sub-matrices. Then, for each possible swap, find the matrix that results. (Since the swaps leave row and column sums unchanged, the resulting matrix should always be one of $a - e$.)
4. Combine all the class results into a single diagram that shows all the moves that result from 2×2 swaps: List each matrix once, and show each move from one matrix to another as a line from the first matrix to the second. (Is it necessary to indicate the direction of the move?)

1.2 Markov chain Monte Carlo

The method of random swaps is an instance of a more general method, Markov chain Monte Carlo, or MCMC. Details and precise definitions will come later, but for now, consider a fanciful metaphor.

The tour of the shortsighted astronomer.

Imagine that you are an astronomer, charged with the task of choosing 1000 stars at random from the whole universe, in such a way that every star has the same chance of being picked. As if this challenge weren't hard enough already, you not only don't have a telescope, you don't even have very good vision, so you can only see a few hundred of the brightest stars that are closest to where you are. Fortunately, however, you have the

capacity for instantaneous space travel – a vehicle that can take you to any star you can see, the moment you decide to go there.

Not one to be outdone by a challenge, you devise an ingenious scheme. Start from where you are. Pick one of the stars you can see, and go there. Now, of course, you've changed your field of vision, and there's a new set of stars, a slightly different portion of the universe, that you can see. Pick one of these stars, at random, choosing with equal probabilities from among those you can see, and go there. Repeat the "pick and go" over and over. Each time you move, you see a new set of stars, from which you pick randomly. Step by step, you tour the universe, even though you never see more than a tiny fraction of it at any one time. In principle, if you take enough steps, the star where you are when you stop can be considered to have been chosen at random from all the stars in the universe.

Notice that for the astronomer's tour to do its job, two requirements must be met: (1) Tours must be long enough. If you stop after just a few steps, stars near your starting point will have an unfair advantage over very distant stars, because you haven't given your tour a chance to go very far. (2) Tours that do take enough steps must be just as likely to end up at one star as another: all stars in the universe must be equally likely. One essential condition for this to be possible is connectedness: Given any two stars a and b in the universe, there must be a path – a sequence of stars leading from a to b , such that the astronomer can always see the next star in the sequence.

The tour of the shortsighted astronomer is a metaphor for MCMC. There is an astronomically large but finite set of possible "states" (a finite universe of all the stars there are), and a transition rule that determines the probability of moving from any given state to any other state. The random "moves" take place in a sequence of steps, which generate a random path through the set of states. What makes the process a Markov chain⁴ is this: at each step, the probabilities for where you go next depend only on where you are, and not on anything else; not, for example, on where you may have been previously. A Markov chain is "memory-less" in this way. The future depends only on the present, not on the past.

You've probably already seen the correspondences between the star search and the finch swap. Each possible co-occurrence matrix is a state, and the universe of all such matrices with given row and column sums has upwards of 10^{17} elements. From any given finch matrix, you can only "see" the ones that differ from yours by a random swap. (From the actual finch data, you can see 333 other matrices.) You pick one at random and go there. From your new matrix, you can see a new, somewhat different set of neighboring matrices, and you pick one of them for the next step in your random tour.

And so it goes ...

⁴ After A. A. Markov, who was among the first to study and write about such processes.

1.3 Three recurring questions

The astronomer's tour raises three clusters of questions that will echo repeatedly throughout the book:

1. **Transition probabilities.** (a) Given any current state a , what are the possible destination states? (b) Given any current state a , and potential destination b , what is the probability p_{ab} of going from a to b in one step? (c) If I know all the one step transition probabilities, what can I say about the two-step, three-step, ... n -step transition probabilities? (d) Are the states **connected**, that is, given any pair of states a and b , is it possible to get from a to b in a finite number of steps?
2. **Limiting distribution.** If I continue long enough, will my chance $p^{(n)}[a]$ of being at a particular state a after n steps converge to a limit $p^\infty[a]$? If so, does the limit depend on the starting state? Are all states equally likely in the limit?
3. **Convergence rate.** How long does my tour have to be – how many steps must I take – in order to give all stars nearly equal chances of being the place where I end up after my last step?

For the Connor/Simberloff swap-walk on co-occurrence matrices, it turns out that the states are connected⁵ and the n -step transition probabilities $p^{(n)}[a]$ do converge, but the limiting probabilities are not equal.⁶ This means that using the swap walk to generate random matrices is biased: the method tends to favor some matrices over others, instead of treating them equally. Fortunately, there are straightforward ways to modify the walk to give equal probabilities.⁷

The question about convergence rates tends to be harder. As of this writing, the answer for the swap walk on binary matrices is not known. Trying to find answers to this question involves some lovely mathematics relating properties of graphs to eigenvalues of the matrix of transition probabilities.⁸

⁵ Proved by Richard Brualdi in 1980: "Matrices of zeros and ones with fixed row and column sum vectors," *Linear Algebra and Its Applications* **33**: 159-231. For other Markov chains, it may well happen that not all states are connected. See, e.g., Diaconis, Persi and Bernd Sturmfels (1998). "Algebraic algorithms for sampling from discrete exponential families," *Annals of Statistics* **26**: 363-397.

⁶ Simberloff, Daniel and Zaman (2000). "Random Binary Matrices in Biogeographical Ecology: Instituting a Good Neighbor Policy," *Environmental and Ecological Statistics* **9**: 405-421.

⁷ One method, called the Metropolis algorithm, is the topic of Chapter IV.

⁸ Diaconis, Persi and Daniel Stroock (1991). "Geometric Bounds for Eigenvalues of Markov Chains," *Ann. Applied Prob.* **1**: 36-61. Also Diaconis, Persi and Laurent Saloff-Coste (1993). "Comparison Theorems for Reversible Markov Chains," *Ann. Applied Prob.* **3**: 696-730.

1.4 Connectedness and the algebra of polynomials

Recall that the astronomer’s tour of the stars can’t work unless his random walk connects all the stars in his universe. If there are stars in one part of the universe that can’t be seen from any of the stars in some other portion of the universe, then his touring algorithm can never take him from one portion to the other. If the stars in one portion of the universe are “visually isolated” from all other stars in the universe, then either the tour remains trapped among these stars, or else can never visit any of them. Such a universe is not connected. The same can be true if you are working with data matrices rather than stars. Consider, for example, the two data tables in Display 1.3.

$$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \xleftrightarrow{???} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

Display 1.3. Two matrices that are not connected by any 2x2 swaps

There is no random swap of the Connor/Simberloff sort that can take you from the first matrix to the second. The two matrices have different sets of row totals, and so naturally any move that preserves (doesn’t change) row and column totals can’t take you from one table to the other.

For binary matrices with fixed “margins”⁹, connectedness is not an issue because it has been known since 1980 that the 2x2 swaps connect any such set.¹⁰ For other data sets – for example, multidimensional tables of counts – it may not be known, and may not be easy to tell, whether a proposed set of moves connects the universe of tables you need to sample from.

Fortunately, there is a way to test for connectedness, and the mathematical ideas involved are quite unexpected and lovely. The sketch that follows is incomplete, oversimplified, and intended only to give a glimpse of what is involved. The guiding strategy is to reduce questions about connectedness to questions about dividing polynomials, so that they can be answered using standard software for computational algebra. I’ve broken the ideas into eight clusters, but that division is somewhat arbitrary.

1. Introduce a set of indeterminates (variables), one for each cell of the table. For example, let variable x_{ij} correspond to the entry in row i , column j of the table. For our 2 x 3 tables, we have

$$\begin{array}{ccc} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \end{array}$$

⁹ Because row and column totals are usually shown in the right-most column and bottom row, the sets of totals are often collectively called the “margins” of the table.

¹⁰ See Brualdi, *op. cit.*

2. Represent each cell entry by the exponent of the corresponding variable. Thus if there is a 1 in cell i,j the exponent of x_{ij} is 1; if there is a 0 in cell i,j , the exponent is 0. By multiplying the terms for all the cells, we represent a data table by a monomial that has x_{ij} present for the cells with 1s in them, and x_{ij} absent for the cells that have 0s in them. For our example, we have representations

$$\begin{array}{ccc}
 x_{11} & x_{12} & x_{23} \\
 1 & 1 & 0 \\
 0 & 0 & 1
 \end{array}
 \quad \text{and} \quad
 \begin{array}{ccc}
 x_{11} & x_{13} & x_{22} \\
 1 & 0 & 1 \\
 0 & 1 & 0
 \end{array}$$

3. Represent a swap by a “monomial difference” of the form

$$x_{12} x_{23} - x_{13} x_{22}$$

where the first term, $x_{12} x_{23}$, tells where the two 1s are located in the swappable 2 x 2 sub-matrix, and the second term $x_{13} x_{22}$ tells where they will be after the swap.

4. It is straightforward to prove that a move connects two tables if and only if its monomial difference divides the difference of the monomials for the two tables:

Example:

$$\begin{array}{ccc}
 \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} & \xrightarrow{\begin{bmatrix} 0 & - & + \\ 0 & + & - \end{bmatrix}} & \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \\
 x_{11}x_{12}x_{23} & x_{12} x_{23} - x_{13} x_{22} & x_{11}x_{13}x_{22}
 \end{array}$$

$$\text{Check: } x_{11}x_{12}x_{23} - x_{11}x_{13}x_{22} = x_{11}(x_{12} x_{23} - x_{13} x_{22})$$

Table 1 Table 2 the move

Display 1.4. Translating a swap into polynomial algebra
 The monomial difference for the move divides the difference of the monomials for the two tables.

5. We can also represent the sets of row and column sums by monomials. Let r_1 and r_2 be variables that correspond to rows 1 and 2. Similarly, let $c_1, c_2,$ and c_3 correspond to the columns. Use exponents to show totals. For our example, the row totals are 2 and 1, and the column totals are 1, 1, and 1. Record these as $r_1^2 r_2^1 c_1^1 c_2^1 c_3^1 = r_1^2 r_2 c_1 c_2 c_3$.
6. We can now define a function ϕ that takes the monomial for a data set as input and gives back the monomial for its row and column totals as output. We first specify what ϕ does to x_{11} : $\phi(x_{11}) = r_1 c_1$. If x_{11} is present in a monomial, there is a 1 in cell (1,1), and that contributes 1 to the total for row 1 and 1 to the total for column 1. These contributions are recorded as exponents of the corresponding variables. Similarly, $\phi(x_{12}) = r_1 c_2, \phi(x_{13}) = r_1 c_3, \phi(x_{21}) = r_2 c_1, \phi(x_{22}) = r_2 c_2,$ and $\phi(x_{23}) = r_2 c_3$.

Finally, extend φ from single x_{ij} s to all polynomials¹¹ in the x_{ij} s by requiring that for any two monomials or polynomials u and v ,

$$\varphi(u+v) = \varphi(u) + \varphi(v) \quad \text{and} \quad \varphi(uv) = \varphi(u)\varphi(v).$$

7. We can now describe algebraically the set of all data tables with given margins¹²:

Two tables have the same margins \Leftrightarrow they have the same value of φ

8. The question “Do the moves connect all the tables with the same marginal row and column totals?” can be replaced by an algebraic question:

Do the monomial differences for the moves allow us to create, via adding and multiplication, all the monomial differences for which the value of φ is 0?¹³

If so, the moves connect the tables, as required. Fortunately, recent developments in computational algebra¹⁴ make it fairly easy to answer this question using standard computer software.

¹¹ If you have taken abstract algebra, you’ll recognize that the function φ is defined in a way that makes it a homomorphism. In linear algebra, there is a special class of functions, the linear transformations, from one vector space to another, that preserve the two operations of vector addition and scalar multiplication. In the algebra of polynomials, homomorphisms play a similar role. In particular, they preserve addition and multiplication of polynomials, as you can see from the final pair of requirements.

¹² In the language of algebra, two tables have the same marginal totals if and only if the difference of their monomials belongs to the kernel of φ , which is an ideal.

¹³ Equivalently, do the monomial differences for the moves generate the kernel of φ ? If these monomial differences constitute a basis for the ideal determined by the homomorphism φ , then the moves themselves connect the set of data tables.

¹⁴ The theory of Groebner bases – see Cox, David, John Little, and Donal O’Shea (1997). *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*, 2nd ed., New York: Springer-Verlag.

Appendix 1.1: Counting 0,1 matrices

Let $\mathbf{r} = (r_1, r_2, \dots, r_m)$ and $\mathbf{c} = (c_1, c_2, \dots, c_n)$ be vectors, and let $\mathbf{A}(\mathbf{r}, \mathbf{c})$ be the set of $m \times n$ matrices of 0s and 1s with row sums r_i and column sums c_j . Let $\#\mathbf{A}(\mathbf{r}, \mathbf{c})$, or, in another common notation, $\#A(\mathbf{r}, \mathbf{c})$, be the number of such matrices. A long-standing research problem is to find $\#\mathbf{A}(\mathbf{r}, \mathbf{c})$.

- An exact formula is known, but it is too complicated to be useful in practice for all but the simplest cases.
- A recent research problem has been to find estimates for $\#\mathbf{A}(\mathbf{r}, \mathbf{c})$, along with bounds that give the maximum possible distance from the estimate to the true value.¹⁵ (This problem is closely tied to the one described in this chapter, of finding a way to generate random matrices from $\mathbf{A}(\mathbf{r}, \mathbf{c})$ so that all the matrices are equally likely.)
- Some special cases are challenging but solvable. For example:
Let $\mathbf{A}_{n,k}$ be the set of $n \times n$ square matrices of 0s and 1s with all row and column sums equal to k . Find $\#\mathbf{A}_{n,k}$ for as many choices of n and k as you can.

Appendix 1.2: A small table of random digits

7 7 5 1 7	7 7 5 5 5	6 4 0 7 4	8 3 2 2 1	0 9 4 7 1
8 5 0 4 4	2 8 5 5 9	2 0 3 1 3	7 2 2 4 3	9 1 9 2 2
5 4 4 6 1	5 4 4 6 7	8 7 1 2 2	2 9 8 9 2	7 4 6 9 6
2 0 2 3 3	3 5 9 6 8	9 1 1 6 7	4 2 9 9 7	2 5 6 1 5
2 8 4 2 1	4 3 9 1 5	6 8 9 9 9	6 7 5 9 7	3 5 8 5 5
5 8 8 8 9	9 7 4 4 4	6 1 3 5 7	1 8 5 8 2	8 5 5 0 7
3 2 9 4 1	2 9 0 2 3	3 8 9 6 8	0 7 2 8 6	5 2 1 0 7
8 5 7 6 3	6 7 0 8 3	6 8 4 1 5	0 9 4 1 5	0 4 3 7 7
1 7 4 7 3	5 5 8 1 2	3 2 7 5 5	9 6 0 1 3	9 2 8 3 8
6 1 7 0 7	6 0 0 2 6	4 1 1 8 6	6 1 8 8 9	6 5 5 1 0
8 3 4 1 4	4 8 9 8 3	9 9 4 6 9	1 1 5 8 0	2 4 0 8 7
8 6 6 5 2	2 0 0 9 0	3 8 6 7 4	8 6 7 6 2	3 8 3 1 1
1 0 2 2 6	3 8 3 5 1	7 9 9 8 5	4 6 6 4 3	6 8 7 6 8
8 4 1 2 3	6 3 2 1 7	3 0 6 2 6	4 8 3 7 0	2 8 9 6 7
8 6 6 0 8	6 7 6 0 1	9 0 4 2 6	9 0 2 8 2	8 6 6 5 8
8 3 0 4 2	8 6 8 4 7	7 5 4 9 3	1 4 1 9 8	3 3 7 7 6
4 8 4 5 5	4 8 9 3 1	8 6 6 0 8	0 1 1 0 6	4 8 5 2 2
5 7 7 2 7	7 6 2 4 5	6 6 5 4 0	0 6 6 5 3	4 9 8 9 5
2 1 7 1 4	9 6 7 1 5	8 8 1 5 8	9 5 5 3 8	6 0 6 5 2

¹⁵ See Jun S. Liu (2001). *Monte Carlo Strategies in Scientific Computing*, New York: Springer-Verlag, Section 4.3.