Chapter 3

Combinatorics

3.1 Permutations

Many problems in probability theory require that we count the number of ways that a particular event can occur. For this, we study the topics of *permutations* and *combinations.* We consider permutations in this section and combinations in the next section.

Before discussing permutations, it is useful to introduce a general counting technique that will enable us to solve a variety of counting problems, including the problem of counting the number of possible permutations of n objects.

Counting Problems

Consider an experiment that takes place in several stages and is such that the number of outcomes m at the nth stage is independent of the outcomes of the previous stages. The number m may be different for different stages. We want to count the number of ways that the entire experiment can be carried out.

Example 3.1 You are eating at Émile's restaurant and the waiter informs you that you have (a) two choices for appetizers: soup or juice; (b) three for the main course: a meat, fish, or vegetable dish; and (c) two for dessert: ice cream or cake. How many possible choices do you have for your complete meal? We illustrate the possible meals by a tree diagram shown in Figure 3.1. Your menu is decided in three stages—at each stage the number of possible choices does not depend on what is chosen in the previous stages: two choices at the first stage, three at the second, and two at the third. From the tree diagram we see that the total number of choices is the product of the number of choices at each stage. In this examples we have $2 \cdot 3 \cdot 2 = 12$ possible menus. Our menu example is an example of the following general counting technique. □

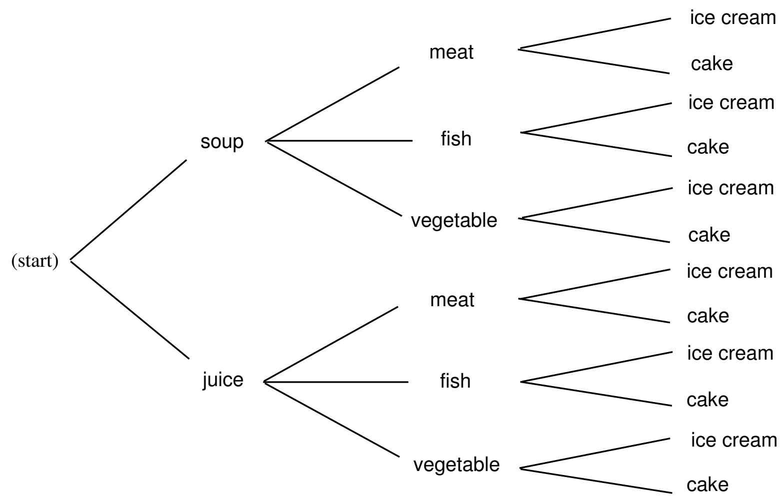

Figure 3.1: Tree for your menu.

A Counting Technique

A task is to be carried out in a sequence of r stages. There are n_1 ways to carry out the first stage; for each of these n_1 ways, there are n_2 ways to carry out the second stage; for each of these n_2 ways, there are n_3 ways to carry out the third stage, and so forth. Then the total number of ways in which the entire task can be accomplished is given by the product $N = n_1 \cdot n_2 \cdot \ldots \cdot n_r$.

Tree Diagrams

It will often be useful to use a tree diagram when studying probabilities of events relating to experiments that take place in stages and for which we are given the probabilities for the outcomes at each stage. For example, assume that the owner of Émile's restaurant has observed that 80 percent of his customers choose the soup for an appetizer and 20 percent choose juice. Of those who choose soup, 50 percent choose meat, 30 percent choose fish, and 20 percent choose the vegetable dish. Of those who choose juice for an appetizer, 30 percent choose meat, 40 percent choose fish, and 30 percent choose the vegetable dish. We can use this to estimate the probabilities at the first two stages as indicated on the tree diagram of Figure 3.2.

We choose for our sample space the set Ω of all possible paths $\omega = \omega_1$, ω_2, $\ldots$, ω_6 through the tree. How should we assign our probability distribution? For example, what probability should we assign to the customer choosing soup and then the meat? If 8/10 of the customers choose soup and then 1/2 of these choose meat, a proportion $8/10 \cdot 1/2 = 4/10$ of the customers choose soup and then meat. This suggests choosing our probability distribution for each path through the tree to be the *product* of the probabilities at each of the stages along the path. This results in the probability distribution for the sample points ω indicated in Figure 3.2. (Note that $m(\omega_1) + \cdots + m(\omega_6) = 1$.) From this we see, for example, that the probability

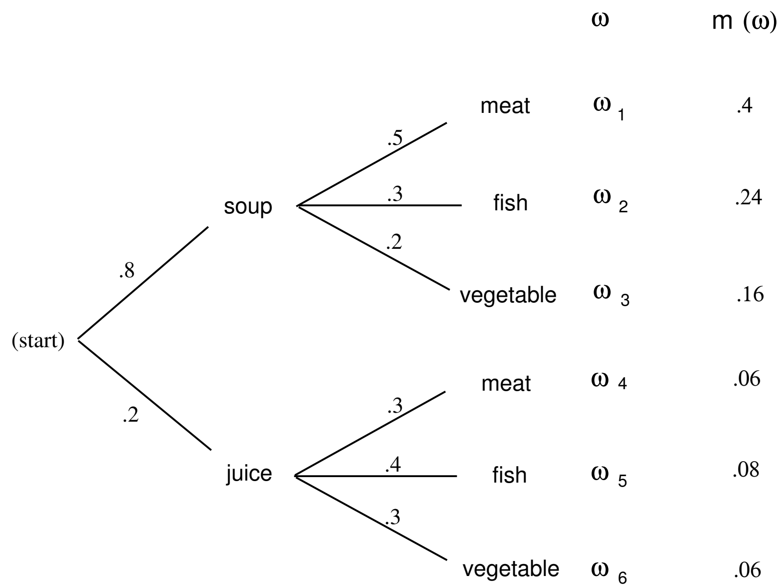

Figure 3.2: Two-stage probability assignment.

that a customer chooses meat is $m(\omega_1) + m(\omega_4) = .46$.

We shall say more about these tree measures when we discuss the concept of conditional probability in Chapter 4. We return now to more counting problems.

Example 3.2 We can show that there are at least two people in Columbus, Ohio, who have the same three initials. Assuming that each person has three initials, there are 26 possibilities for a person's first initial, 26 for the second, and 26 for the third. Therefore, there are $26^3 = 17{,}576$ possible sets of initials. This number is smaller than the number of people living in Columbus, Ohio; hence, there must be at least two people with the same three initials. □

We consider next the celebrated birthday problem—often used to show that naive intuition cannot always be trusted in probability.

Birthday Problem

Example 3.3 How many people do we need to have in a room to make it a favorable bet (probability of success greater than 1/2) that two people in the room will have the same birthday?

Since there are 365 possible birthdays, it is tempting to guess that we would need about 1/2 this number, or 183. You would surely win this bet. In fact, the number required for a favorable bet is only 23. To show this, we find the probability p_r that, in a room with r people, there is no duplication of birthdays; we will have a favorable bet if this probability is less than one half.

Number of people	Probability that all birthdays are different
20	.5885616
21	.5563117
22	.5243047
23	.4927028
24	.4616557
25	.4313003

Table 3.1: Birthday problem.

Assume that there are 365 possible birthdays for each person (we ignore leap years). Order the people from 1 to r. For a sample point ω, we choose a possible sequence of length r of birthdays each chosen as one of the 365 possible dates. There are 365 possibilities for the first element of the sequence, and for each of these choices there are 365 for the second, and so forth, making 365^r possible sequences of birthdays. We must find the number of these sequences that have no duplication of birthdays. For such a sequence, we can choose any of the 365 days for the first element, then any of the remaining 364 for the second, 363 for the third, and so forth, until we make r choices. For the rth choice, there will be $365 - r + 1$ possibilities. Hence, the total number of sequences with no duplications is

$$365 \cdot 364 \cdot 363 \cdot \ldots \cdot (365 - r + 1) .$$

Thus, assuming that each sequence is equally likely,

$$p_r = \frac{365 \cdot 364 \cdot \ldots \cdot (365 - r + 1)}{365^r} .$$

We denote the product

$$(n)(n-1)\cdots(n-r+1)$$

by $(n)_r$ (read "n down r," or "n lower r"). Thus,

$$p_r = \frac{(365)_r}{(365)^r} .$$

The program **Birthday** carries out this computation and prints the probabilities for $r = 20$ to 25. Running this program, we get the results shown in Table 3.1. As we asserted above, the probability for no duplication changes from greater than one half to less than one half as we move from 22 to 23 people. To see how unlikely it is that we would lose our bet for larger numbers of people, we have run the program again, printing out values from $r = 10$ to $r = 100$ in steps of 10. We see that in a room of 40 people the odds already heavily favor a duplication, and in a room of 100 the odds are overwhelmingly in favor of a duplication. We have assumed that birthdays are equally likely to fall on any particular day. Statistical evidence suggests that this is not true. However, it is intuitively clear (but not easy to prove) that this makes it even more likely to have a duplication with a group of 23 people. (See Exercise 19 to find out what happens on planets with more or fewer than 365 days per year.) □

Number of people	Probability that all birthdays are different
10	.8830518
20	.5885616
30	.2936838
40	.1087682
50	.0296264
60	.0058773
70	.0008404
80	.0000857
90	.0000062
100	.0000003

Table 3.2: Birthday problem.

We now turn to the topic of permutations.

Permutations

Definition 3.1 Let A be any finite set. A *permutation of* A is a one-to-one mapping of A onto itself. □

To specify a particular permutation we list the elements of A and, under them, show where each element is sent by the one-to-one mapping. For example, if $A = \{a, b, c\}$ a possible permutation σ would be

$$\sigma = \begin{pmatrix} a & b & c \\ b & c & a \end{pmatrix}.$$

By the permutation σ, a is sent to b, b is sent to c, and c is sent to a. The condition that the mapping be one-to-one means that no two elements of A are sent, by the mapping, into the same element of A.

We can put the elements of our set in some order and rename them 1, 2, …, n. Then, a typical permutation of the set $A = \{a_1, a_2, a_3, a_4\}$ can be written in the form

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \end{pmatrix},$$

indicating that a_1 went to a_2, a_2 to a_1, a_3 to a_4, and a_4 to a_3.

If we always choose the top row to be 1 2 3 4 then, to prescribe the permutation, we need only give the bottom row, with the understanding that this tells us where 1 goes, 2 goes, and so forth, under the mapping. When this is done, the permutation is often called a *rearrangement* of the n objects 1, 2, 3, …, n. For example, all possible permutations, or rearrangements, of the numbers $A = \{1, 2, 3\}$ are:

$$123,\ 132,\ 213,\ 231,\ 312,\ 321\ .$$

It is an easy matter to count the number of possible permutations of n objects. By our general counting principle, there are n ways to assign the first element, for

n	$n!$
0	1
1	1
2	2
3	6
4	24
5	120
6	720
7	5040
8	40320
9	362880
10	3628800

Table 3.3: Values of the factorial function.

each of these we have $n-1$ ways to assign the second object, $n-2$ for the third, and so forth. This proves the following theorem.

Theorem 3.1 The total number of permutations of a set A of n elements is given by $n \cdot (n - 1) \cdot (n-2) \cdot \ldots \cdot 1$. $\square$

It is sometimes helpful to consider orderings of subsets of a given set. This prompts the following definition.

Definition 3.2 Let A be an n-element set, and let k be an integer between 0 and n. Then a k-permutation of A is an ordered listing of a subset of A of size k. $\square$

Using the same techniques as in the last theorem, the following result is easily proved.

Theorem 3.2 The total number of k-permutations of a set A of n elements is given by $n \cdot (n-1) \cdot (n-2) \cdot \ldots \cdot (n-k+1)$. $\square$

Factorials

The number given in Theorem 3.1 is called n *factorial,* and is denoted by $n!$. The expression $0!$ is defined to be 1 to make certain formulas come out simpler. The first few values of this function are shown in Table 3.3. The reader will note that this function grows very rapidly.

The expression $n!$ will enter into many of our calculations, and we shall need to have some estimate of its magnitude when n is large. It is clearly not practical to make exact calculations in this case. We shall instead use a result called *Stirling's formula.* Before stating this formula we need a definition.

n	$n!$	Approximation	Ratio
1	1	.922	1.084
2	2	1.919	1.042
3	6	5.836	1.028
4	24	23.506	1.021
5	120	118.019	1.016
6	720	710.078	1.013
7	5040	4980.396	1.011
8	40320	39902.395	1.010
9	362880	359536.873	1.009
10	3628800	3598696.619	1.008

Table 3.4: Stirling approximations to the factorial function.

Definition 3.3 Let a_n and b_n be two sequences of numbers. We say that a_n is *asymptotically equal to* b_n, and write $a_n \sim b_n$, if

$$\lim_{n\to\infty} \frac{a_n}{b_n} = 1 \ .$$

□

Example 3.4 If $a_n = n + \sqrt{n}$ and $b_n = n$ then, since $a_n/b_n = 1 + 1/\sqrt{n}$ and this ratio tends to 1 as n tends to infinity, we have $a_n \sim b_n$. □

Theorem 3.3 (Stirling's Formula) The sequence $n!$ is asymptotically equal to

$$n^n e^{-n} \sqrt{2\pi n} \ .$$

□

The proof of Stirling's formula may be found in most analysis texts. Let us verify this approximation by using the computer. The program **StirlingApproximations** prints $n!$, the Stirling approximation, and, finally, the ratio of these two numbers. Sample output of this program is shown in Table 3.4. Note that, while the ratio of the numbers is getting closer to 1, the difference between the exact value and the approximation is increasing, and indeed, this difference will tend to infinity as n tends to infinity, even though the ratio tends to 1. (This was also true in our Example 3.4 where $n + \sqrt{n} \sim n$, but the difference is $\sqrt{n}$.)

Generating Random Permutations

We now consider the question of generating a random permutation of the integers between 1 and n. Consider the following experiment. We start with a deck of n cards, labelled 1 through n. We choose a random card out of the deck, note its label, and put the card aside. We repeat this process until all n cards have been chosen. It is clear that each permutation of the integers from 1 to n can occur as a sequence

Number of fixed points	Fraction of permutations		
	n = 10	n = 20	n = 30
0	.362	.370	.358
1	.368	.396	.358
2	.202	.164	.192
3	.052	.060	.070
4	.012	.008	.020
5	.004	.002	.002
Average number of fixed points	.996	.948	1.042

Table 3.5: Fixed point distributions.

of labels in this experiment, and that each sequence of labels is equally likely to occur. In our implementations of the computer algorithms, the above procedure is called **RandomPermutation**.

Fixed Points

There are many interesting problems that relate to properties of a permutation chosen at random from the set of all permutations of a given finite set. For example, since a permutation is a one-to-one mapping of the set onto itself, it is interesting to ask how many points are mapped onto themselves. We call such points *fixed points* of the mapping.

Let $p_k(n)$ be the probability that a random permutation of the set $\{1, 2, \ldots, n\}$ has exactly k fixed points. We will attempt to learn something about these probabilities using simulation. The program **FixedPoints** uses the procedure **RandomPermutation** to generate random permutations and count fixed points. The program prints the proportion of times that there are k fixed points as well as the average number of fixed points. The results of this program for 500 simulations for the cases $n = 10$, 20, and 30 are shown in Table 3.5. Notice the rather surprising fact that our estimates for the probabilities do not seem to depend very heavily on the number of elements in the permutation. For example, the probability that there are no fixed points, when $n = 10,\ 20$, or 30 is estimated to be between .35 and .37. We shall see later (see Example 3.12) that for $n \geq 10$ the exact probabilities $p_n(0)$ are, to six decimal place accuracy, equal to $1/e \approx .367879$. Thus, for all practical purposes, after $n = 10$ the probability that a random permutation of the set $\{1, 2, \ldots, n\}$ has no fixed points does not depend upon n. These simulations also suggest that the average number of fixed points is close to 1. It can be shown (see Example 6.8) that the average is exactly equal to 1 for all n.

More picturesque versions of the fixed-point problem are: You have arranged the books on your book shelf in alphabetical order by author and they get returned to your shelf at random; what is the probability that exactly k of the books end up in their correct position? (The library problem.) In a restaurant n hats are checked and they are hopelessly scrambled; what is the probability that no one gets his own hat back? (The hat check problem.) In the Historical Remarks at the end of this section, we give one method for solving the hat check problem exactly. Another

Date	Snowfall in inches
1974	75
1975	88
1976	72
1977	110
1978	85
1979	30
1980	55
1981	86
1982	51
1983	64

Table 3.6: Snowfall in Hanover.

Year	1	2	3	4	5	6	7	8	9	10
Ranking	6	9	5	10	7	1	3	8	2	4

Table 3.7: Ranking of total snowfall.

method is given in Example 3.12.

Records

Here is another interesting probability problem that involves permutations. Estimates for the amount of measured snow in inches in Hanover, New Hampshire, in the ten years from 1974 to 1983 are shown in Table 3.6. Suppose we have started keeping records in 1974. Then our first year's snowfall could be considered a record snowfall starting from this year. A new record was established in 1975; the next record was established in 1977, and there were no new records established after this year. Thus, in this ten-year period, there were three records established: 1974, 1975, and 1977. The question that we ask is: How many records should we expect to be established in such a ten-year period? We can count the number of records in terms of a permutation as follows: We number the years from 1 to 10. The actual amounts of snowfall are not important but their relative sizes are. We can, therefore, change the numbers measuring snowfalls to numbers 1 to 10 by replacing the smallest number by 1, the next smallest by 2, and so forth. (We assume that there are no ties.) For our example, we obtain the data shown in Table 3.7.

This gives us a permutation of the numbers from 1 to 10 and, from this permutation, we can read off the records; they are in years 1, 2, and 4. Thus we can define records for a permutation as follows:

Definition 3.4 Let σ be a permutation of the set $\{1, 2, \ldots, n\}$. Then i is a *record* of σ if either $i = 1$ or $\sigma(j) < \sigma(i)$ for every $j = 1, \ldots, i - 1$. □

Now if we regard all rankings of snowfalls over an n-year period to be equally likely (and allow no ties), we can estimate the probability that there will be k records in n years as well as the average number of records by simulation.

We have written a program **Records** that counts the number of records in randomly chosen permutations. We have run this program for the cases $n = 10$, 20, 30. For $n = 10$ the average number of records is 2.968, for 20 it is 3.656, and for 30 it is 3.960. We see now that the averages increase, but very slowly. We shall see later (see Example 6.11) that the average number is approximately $\log n$. Since $\log 10 = 2.3$, $\log 20 = 3$, and $\log 30 = 3.4$, this is consistent with the results of our simulations.

As remarked earlier, we shall be able to obtain formulas for exact results of certain problems of the above type. However, only minor changes in the problem make this impossible. The power of simulation is that minor changes in a problem do not make the simulation much more difficult. (See Exercise 20 for an interesting variation of the hat check problem.)

List of Permutations

Another method to solve problems that is not sensitive to small changes in the problem is to have the computer simply list all possible permutations and count the fraction that have the desired property. The program **AllPermutations** produces a list of all of the permutations of n. When we try running this program, we run into a limitation on the use of the computer. The number of permutations of n increases so rapidly that even to list all permutations of 20 objects is impractical.

Historical Remarks

Our basic counting principle stated that if you can do one thing in r ways and for each of these another thing in s ways, then you can do the pair in rs ways. This is such a self-evident result that you might expect that it occurred very early in mathematics. N. L. Biggs suggests that we might trace an example of this principle as follows: First, he relates a popular nursery rhyme dating back to at least 1730:

> As I was going to St. Ives,
> I met a man with seven wives,
> Each wife had seven sacks,
> Each sack had seven cats,
> Each cat had seven kits.
> Kits, cats, sacks and wives,
> How many were going to St. Ives?

(You need our principle only if you are not clever enough to realize that you are supposed to answer *one,* since only the narrator is going to St. Ives; the others are going in the other direction!)

He also gives a problem appearing on one of the oldest surviving mathematical manuscripts of about 1650 B.C., roughly translated as: