Enumeration

All sequences of outcomes

Given a set of outcomes, a standard problem in combinatorics is to *enumerate* all possible sequences of outcomes formed by repeatedly choosing an outcome from the set. For this class page, we will restrict our attention to the case where repeated outcomes are allowed. This problem typically arises when one is analyzing the probabilities associated with a sequence of trials.

As a starting point, let's consider the question of how many sequences of length n are possible if the set of outcomes has size m. This question can be answered using a simple counting argument. There are m choices for the first element in the sequence. Since repetitions are allowed, there are also m choices for the second item and so on. In general, the number of sequences of length n is

$$m \times m \times m \times ... \times m = m^n.$$

This program computes the set of all possible sequences of outcomes of a specified length using the function `gen_all_sequences`. (The program also includes a function `gen_sorted_sequences` which we will discuss later.) Note that these functions represent sequences as tuples since the members of a set must be immutable.

Sorted sequences of outcomes

In some applications, the ordering of the outcomes in each sequence is unimportant. For example, in most dice games (including Yahtzee), the two sequences of rolls $(4, 3, 2, 1, 2)$ and $(1, 2, 2, 3, 4)$ each represent the same hand. In cases such as these, it may make sense to treat these two sequences as being equivalent to improve the performance of your program. (We won't bother with incorporating this optimization into Yahtzee since it will keep your code simpler.)

A standard technique for accomplishing this goal is to group all sequences of outcomes that have the same number of instances of each outcome (but in different orders) in a single cluster and then choose a single representative sequence for this cluster. One simple choice for this representative is to take the sequence in which the outcomes appear in sorted (ascending) order. In the dice example in the previous paragraph, the sorted sequence $(1, 2, 2, 3, 4)$ would be the representative sequence for all sequences that contain one 1, two 2's, one 3, and one 4.

While building a function that directly generates these sorted sequences is possible, we note that a simpler approach is to generate all possible sequences, sort each individual sequence, and add them into a new set. Insertion into the set will automatically eliminate the duplicate sequences that arise from sorting. The function `gen_sorted_sequences` generates the set of all sorted sequences using this method. Note that this is a substantially smaller subset of the set of sequences produced by `gen_all_sequences`. (For those that are interested, this set has size $\frac{(n+m-1)!}{n!(m-1)!}$.)

Applications to probability

In many applications in probability, we are interested in enumerating the sequences of outcomes associated with a sequence of trials. For this class, we restrict our attention to the situation in which each trial is *independent*; that is the outcome of the trial does not affect the probabilities associated with the outcomes of other trials. For a sequence of independent trials, the probability of a particular sequence of outcomes is the product of the probabilities associated with each individual outcome.

A simple example of an independent sequence of trials would be rolling a pair of dice. In this case, the roll of the first die would have six outcomes $\{1, 2, 3, 4, 5, 6\}$ and the roll of the second die would have six outcomes $\{1, 2, 3, 4, 5, 6\}$. Joining an outcome from the first roll with an outcome from the second roll yields a set of all 36 possible sequences of outcomes,

$$\{(1,1),(1,2),(1,3),(1,4),(1,5),(1,6),(2,1),\ldots,(6,3),(6,4),(6,5),(6,6)\}$$

Since we considered all sequences (not just sorted sequences), the probability associated with each sequence of outcomes is the same ($\frac{1}{36}$).

In this setting, we can now discuss events that correspond to a subset of this set of sequences. For the case of two dice, a typical question might be: what is the probability that sum of the dice is exactly 7? Since each of the 36 possible sequences of outcomes has a probability of $\frac{1}{36}$ and six of these outcomes $\{(1,6),(2,5),(3,4),(4,3),(5,2),(6,1)\}$ correspond to an event where the sum is 7, the probability that the sum of two dice is 7 is exactly $\frac{6}{36} = \frac{1}{6}$. Note the sequences $(2,5)$ and $(5,2)$ must be treated as distinct sequences for this analysis.

Created Sun 17 Aug 20148:37 PM BST

Last Modified Sun 17 Aug 2014 8:37 PM BST