Chapter 2
Probability

Finite and Infinite Probability

One's ability to determine the probability of an event is based upon whether the event occurs in a finite or infinite population. In a **finite population**, the number of objects or events is known. An exact **probability** or fraction can be determined. For example, given a population of 1,000 cars with 500 Ford, 200 Chevrolet, 200 Chrysler, and 100 Oldsmobile, the probability of selecting a Ford is one-half or 50% (500/1,000). The probability of selecting a Chevrolet is one-fifth or 20% (200/1,000), the probability of selecting a Chrysler is one-fifth or 20% (200/1,000), and the probability of selecting an Oldsmobile is one-tenth or 10% (100/1,000). The individual probabilities add up to 100%.

In an **infinite population**, the numbers of objects or events are so numerous that exact probabilities are difficult to determine. One approach to determine the probability of an event occurring in an infinite population is to use the relative frequency definition of probability. Using this approach, trials are repeated a large number of times, N. The number of times the event occurs is counted, and the probability of the event is approximated by $P(A) \approx n(A)/N$ in which n(A) is the number of times event A occurred out of N trials.

For example, the probability of obtaining heads when a coin is tossed could be determined by tossing the coin 500 times, counting the number of heads, $n(\text{heads}) = 241$, and computing $P(\text{heads}) \approx 241/500 = 0.482$. The probability of getting heads in the population is therefore approximately 48%. The probability of *not* getting heads is 52%, since the two events must sum to 100%. We know from experience that the probability of obtaining heads when a coin is tossed should be approximately 50% or one-half, given an unbiased coin (a coin that is not weighted or not a trick coin).

R. Schumacker and S. Tomek, *Understanding Statistics Using R*,
DOI 10.1007/978-1-4614-6227-9_2,

An important point needs to be emphasized. In order for an approximation to be reasonable (representative), the relative frequency of the event (e.g., heads) must begin to stabilize and approach some fixed number as the number of trials increases. If this does not occur, then very different approximations would be assigned to the same event as the number of trials increases. Typically, more trials (coin tosses) are required to potentially achieve the 50% probability of obtaining heads. Experience in the real world has shown that the relative frequency of obtaining heads when coins are tossed stabilizes or better approximates the expected probability of 50%, as the number of trials increases.

This approximation phenomenon (stabilization or representativeness) occurs in the relative frequencies of other events too. There is no actual proof of this because of the nature of an infinite population, but experience does support it. Using the relative frequency definition, the approximation, which the relative frequencies provide, is regarded as our best estimate of the actual probability of the event.

In this chapter, you will have an opportunity to observe the stabilization of the relative frequencies. You will be able to choose the probability of the event and the R program will simulate the number of trials. As the number of trials increase, the new results are pooled with the earlier ones, and the relative frequency of the event is computed and plotted on a graph. The first program example starts with a sample size of 100 and increases the sample size in increments of 100 up to 1,000.

In the real world we observe that as the number of trials increases the relative frequency of an event approaches a fixed value. The probability of an event can be defined as the fixed value approximated by the relative frequencies of the event as the number of trials increase. The relative frequency definition of probability assumes that the relative frequencies stabilize as the number of trials increase. Although the relative frequencies get closer to a fixed value as the number of trials increase, it is possible for a certain number of trials to produce a relative frequency that is not closer to the approximated fixed value. An event with a probability close to 0.1 or 0.9 will have relative frequencies that stabilize faster than an event with probability close to 0.50.

PROBABILITY R Program

The PROBABILITY program simulates flipping a coin a different number of times for different samples, and observing the different frequencies across sample sizes. The population probability is set in the variable *Probability* and the *SampleSizes* are created using the **seq** function, instead of specifying each value within a **c** function (which would be 10 numbers in this case). The **seq** function creates a vector of values from 100 to 1,000 with intervals of 100. Next, the *SampleFreqs* object is set to **NULL** so that it may be constructed into a vector by appending values using the **c** function. A **for** loop iterates through all values within the *SampleSizes* vector, assigning each value in turn to the *SampleSize* variable.

The *SampleFreqs* vector is then increased by appending the sum of the samples of size *SampleSize* taken from the population of 0 and 1. This results in 0 having a probability of (1 − *Probability*) and 1 having a probability of *Probability*, which is divided by *SampleSize* to get the relative frequency. The *SampleFreqs* vector now contains the relative frequencies of heads in each sample size. These relative frequencies are plotted with a line graph using the generic **plot** function. *SampleSize* is used for the data on the x-axis and *SampleFreqs* for the data on the y-axis. **Type** = "l" (a lower case L) is for line graph and the **ylim** keyword sets the upper and lower limits of values for the y-axis.

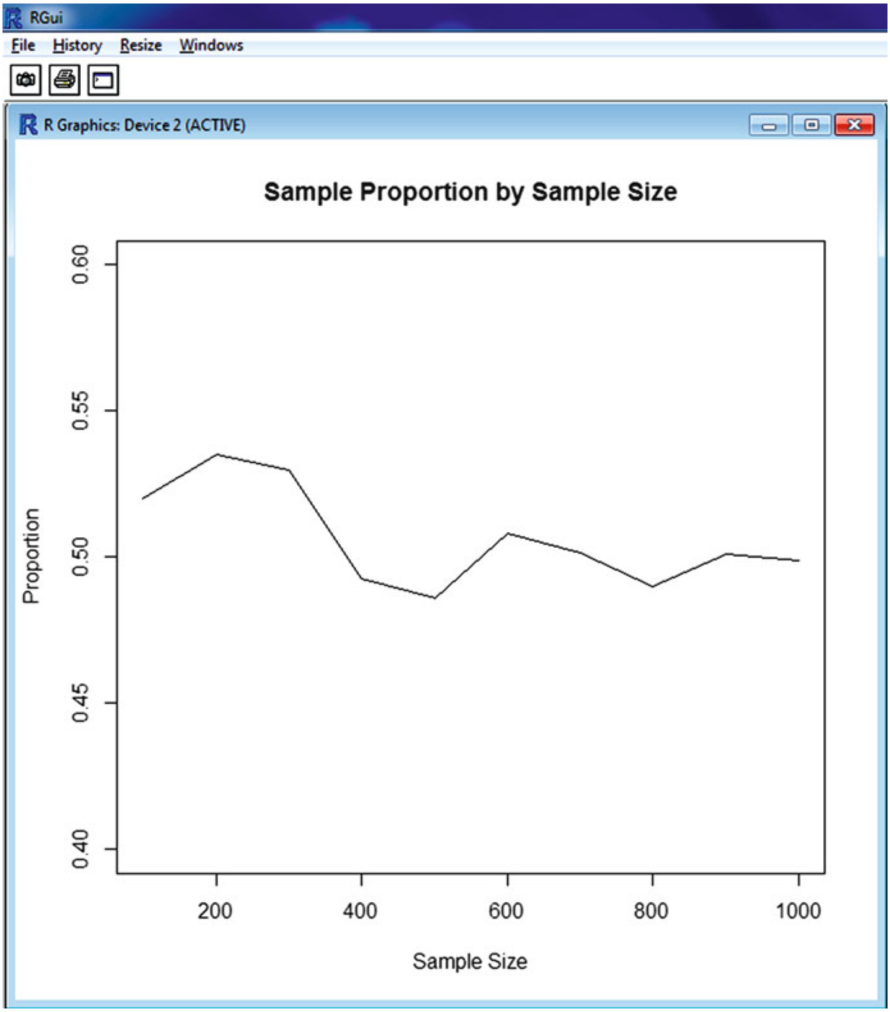
Using the full range of possible values for the y-axis (0 to 1) resulted in difficulty distinguishing differences in the graphs because of the small variation of the values compared to the overall scale. Since values rarely fall 0.10 above or 0.10 below the population probability, the y-axis limits were set to *Probability* −0.10 and *Probability* +0.10. The x-axis label, y-axis label, and main title are all set by use of keywords.

The values from the graph are constructed into a matrix for output. The matrix starts out as a **NULL** object that is built using **rbind** (row bind). A row of values is added to the matrix for each iteration of the **for** loop, appending the relative frequency of the given sample, the population probability, and the error of estimation for each sample (relative frequency—population probability). After the matrix is constructed and the **for** loop is ended, the **dimnames** function is used to assign dimension names to the constructed matrix. The **paste** function is again utilized to create a vector of labels for the rows of the matrix resulting in "sample size = 100", etc. The **print** function is used to output the matrix in order to make the rows and columns printed. The Error is the difference between the sample percent and the true population percent.

NOTE: As noted in Chap. 1, use set.seed(13579) prior to running the R programs to get identical results presented below.

PROBABILITY R Program Output

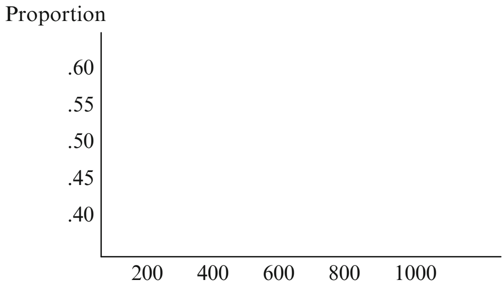
```
                        Sample %   Population %   Error
sample size = 100        0. 52       0.500          0.02
sample size = 200        0.535       0.500          0.035
sample size = 300        0. 53       0.500          0.03
sample size = 400        0.492       0.500         -0.008
sample size = 500        0.486       0.500         -0.014
sample size = 600        0.508       0.500          0.008
sample size = 700        0.501       0.500          0.001
sample size = 800        0. 49       0.500         -0.01
sample size = 900        0.501       0.500          0.001
sample size = 1000       0.499       0.500         -0.001
```



Finite and Infinite Exercises

1. Run PROBABILITY for sample sizes of 50 in increments of 50 up to 1,000, i.e., SampleSizes <- seq(50,1000,50). This is a simulation of flipping an unbiased, balanced coin and recording the relative frequency of obtaining heads, which has an expected probability of p = 0.50.

   a. Complete the graph of relative frequencies for sample sizes of n = 50, in increments of 50, up to 1,000, for $p = 0.50$.



b. Complete the table below. Does the Error (difference between the sample percent and the population percent of 0.50) ever become less than 0.01? ________________

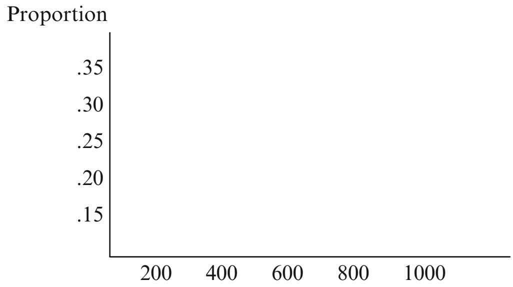
If so, for what sample sizes? ____

Table of sample and population percents for coin toss

	SAMPLE %	POPULATION %	ERROR
SAMPLE SIZE = 50		0.500	
SAMPLE SIZE = 100		0.500	
SAMPLE SIZE = 150		0.500	
SAMPLE SIZE = 200		0.500	
SAMPLE SIZE = 250		0.500	
SAMPLE SIZE = 300		0.500	
SAMPLE SIZE = 350		0.500	
SAMPLE SIZE = 400		0.500	
SAMPLE SIZE = 450		0.500	
SAMPLE SIZE = 500		0.500	
SAMPLE SIZE = 550		0.500	
SAMPLE SIZE = 600		0.500	
SAMPLE SIZE = 650		0.500	
SAMPLE SIZE = 700		0.500	
SAMPLE SIZE = 750		0.500	
SAMPLE SIZE = 800		0.500	
SAMPLE SIZE = 850		0.500	
SAMPLE SIZE = 900		0.500	
SAMPLE SIZE = 950		0.500	
SAMPLE SIZE = 1000		0.500	

2. Run PROBABILITY for sample sizes of 50 in increments of 50 up to 1,000, i.e., SampleSizes <- seq(50,1000,50). This time change the population percent to 25%, i.e., Probability <- 0.25. You are simulating the flipping of a biased, unbalanced coin.

   a. Complete the graph of relative frequencies for sample sizes of n = 50, in increments of 50, up to 1,000, for $p = 0.25$.



b. Complete the table below. Does the absolute difference between the sample percent and the population percent of 0.250 ever become less than 0.01? __________

If so, for what sample sizes? ____

Table of sample and population percents for coin toss

	SAMPLE %	POPULATION %	ERROR
SAMPLE SIZE = 50		0.250	
SAMPLE SIZE = 100		0.250	
SAMPLE SIZE = 150		0.250	
SAMPLE SIZE = 200		0.250	
SAMPLE SIZE = 250		0.250	
SAMPLE SIZE = 300		0.250	
SAMPLE SIZE = 350		0.250	
SAMPLE SIZE = 400		0.250	
SAMPLE SIZE = 450		0.250	
SAMPLE SIZE = 500		0.250	
SAMPLE SIZE = 550		0.250	
SAMPLE SIZE = 600		0.250	
SAMPLE SIZE = 650		0.250	
SAMPLE SIZE = 700		0.250	
SAMPLE SIZE = 750		0.250	
SAMPLE SIZE = 800		0.250	
SAMPLE SIZE = 850		0.250	
SAMPLE SIZE = 900		0.250	
SAMPLE SIZE = 950		0.250	
SAMPLE SIZE = 1000		0.250	

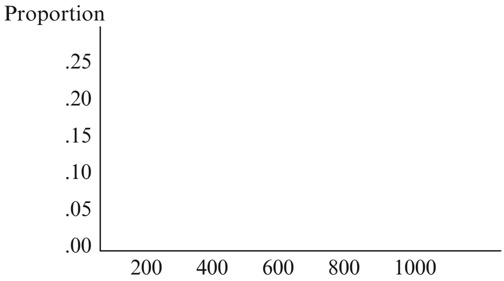
c. In what way is this graph for $p = 0.25$ different from the first graph for $p = 0.50$?

____

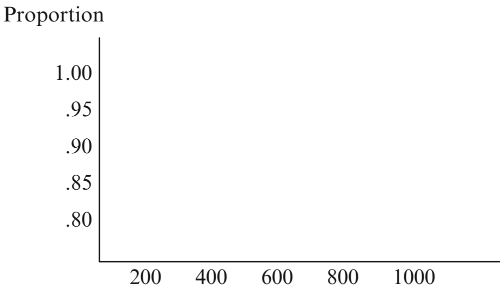
____

3. Run PROBABILITY again for sample sizes of 50, in increments of 50, up to 1,000, but this time for $p = 0.10$ and $p = 0.90$. Draw the graphs below.

p = .10



p = .90



a. In what way are these graphs different from the graphs for a probability of 0.50?

____

____

____

b. What is the implication of this difference when you approximate a very small or a very large probability?

____

____

____

c. Run PROBABILITY for $p = 0.20$, 0.30, 0.40, 0.60, 0.70, and 0.80. Describe the graphs in comparison with those for probabilities of 0.10, 0.90, and 0.50.

____

____

____

Joint Probability

The theoretical probability for the joint occurrence of two independent events is reflected in the relative frequency of their joint occurrence. There is a multiplication and addition law of probability for two independent or mutually exclusive events. The theoretical probability for the union of two events is reflected in the relative frequency of the occurrence of either event.

If an unbiased coin is flipped two times, the possible outcomes form a sample space, S. The sample space S = {HH, HT, TH, TT}, in which H stands for a head and T for a tail, with the pair of letters indicating the order of the outcomes. Therefore, with two separate flips of a coin, four possible outcomes can occur: two heads, first a head then a tail, first a tail then a head, or two tails. The sample space that contains the number of heads in the two flips is S = {0, 1, 2}. If an unbiased coin is flipped twice, a large number of times, the outcomes can be used to compute the frequencies. The frequencies can be used to approximate the joint probabilities of the outcomes in the sample space.

Probabilities can also be assigned to the outcomes by using a theoretical approach. Since a head and a tail are equally likely to occur on a single flip of an unbiased coin, the theoretical approach uses a definition of probability that is applicable to equally likely events. The probability of a head, P(H), is 1/2 because a head is one of the two equally likely outcomes. The probability of a tail, P(T), is 1/2, since a tail represents the other equally likely outcome. Since the two flips of the coin are independent, the multiplication law of probability for independent events can be used to find the joint probability for the pairs in the sample space. For example, the probability of flipping an unbiased coin and getting heads both times is: P(HH) = (1/2)*(1/2) = 1/4. The probability of getting a head and then a tail would be: P(HT) = (1/2)*(1/2) = 1/4, with the other pairs in the sample space determined in the same manner.

If the coin is *biased*, meaning that P(H) is some value other than 0.50, for example 0.60, then P(T) = 1 − P(H) = 1 − 0.60 = 0.40. The independence of the coin flips can be used to find the **joint probability** for the pair. For example, P(HT) = P(H)*P(T) = (0.60)*(0.40) = 0.24.

If the sample space being used is S = {0, 1, 2}, with the integers representing the number of heads, then the frequency of 0 is the frequency of TT; the frequency of 1 is the frequency of HT plus the frequency of TH; and the frequency of 2 is the frequency of HH. The theoretical probabilities for S can also be obtained by using the addition law of probability for mutually exclusive events. For example, P(1) = P(HT) + P(TH).

The multiplication and addition laws of probability for independent events reflects the properties of frequencies. If two events A and B are independent, then P(A and B) = P(A)*P(B). If two events A and B are mutually exclusive, then P(A and B) = P(A) + P(B).

JOINT PROBABILITY R Program

The JOINT R program specifies the probability of tossing a head P(H), and the number of repetitions of the two coin flips. The program will simulate tossing the coin, compute the frequencies and the probabilities. The frequencies approximate the probabilities. This supports the conclusion that the theoretical laws used to compute the probabilities give results similar to the frequency of data obtained in practice. Each time the program is run, the frequencies will be different because random flips of the coin are simulated.

The program simulates tossing two (or more) coins *N* number of times. The program begins by initializing the probability of obtaining a head, the number of coins to be tossed, and the number of times to toss each coin. A vector of heads (1) or tails (0) values is created, and then grouped into a **matrix** with the number of columns equal to the number of coins (column 1 = coin 1, etc.) and the number of rows equal to the number of times each coin is tossed. Next, a vector is initialized and then filled using a **for** loop with the sum of the number of heads in each round of tosses. A complex nested function allows for any number of coins to be tossed.

Vectors for the event labels (HH, HT, TH, TT), the event probabilities, and the number of heads present in each event are initialized with the appropriate values. The outer loop represents the range of possible events given the number of coins. The number of possible events is 2^*numCoins*, which is read 2 to the power of *numCoins*. The loop range is set to 0 for 2^*numCoins*-1. A temporary holding variable is set to the current value of the outer loop counter, *i,* and then the inner loop begins, which represents each coin tossed in a given round taken in reverse order.

In order to make each event unique, a binary coding system is used whereby the event's value (*i*) is broken down into binary values for each toss of a coin in the group. An event value of zero for two coins would mean tails–tails (or 0-0). An event value of one, for two coin tosses, would mean heads–tails (or 1-0). The first coin tossed has a value of either 0 or 1. The second coin tossed a value of either 0 or 2. The third coin tossed (if there were a third coin) would have a value of 0 or 4. The *nth* toss would have a value of 0 or 2^(n – 1), i.e., 2^*numCoins*-1. In this manner, all the unique events (from 0 to 2^*numCoins*-1) are broken down into whether the first and/or second (and/or third and/or fourth, etc) coins are heads or tails for that event. Labels are created with ordered letters representing what the binary coding represents internally. The label "HH" for an event of head–head is more readily understood than an event code of 3.

The number of total heads for each event is recorded. The vector containing the heads count is factored for all possible values and then counted by means of a **table** function to determine the total number of events resulting in 0 heads, 1 head, 2 heads, and so forth, depending upon the number of coin tosses. The total number of events is then used in calculating the probabilities in the **for** loop.

The **for** loop calculates the probabilities for each number of heads by the order of the event. If the probability of getting a head is 0.60, then the probability of getting two tails (or no heads) on the first toss is (1)(0.40)(0.40) = 0.16. This implies that there is only one way to get two tails (1), times the probability of tails (0.40), times the probability of tails (0.40). The probability of getting one head and one tail is (2)(0.60)(0.40) = 0.48. There are two different ways (head–tail or tail–head) you could get the pair of heads and tails (2), times probability of heads (0.60), times probability of tails (0.40). The loop variable, *i,* represents the number of heads and *numEvents*[*i*+1] represents the number of events in the event space for that number of heads. The probability of a head (*pH*) is taken to the power of the number of heads obtained (*i*) and any coins that aren't heads must be tails (*numCoins* − *i*), so the probability of a tail (1 − *pH*) is taken to the power of that value. For an event that involves flipping a coin two times, the loop will go from 0 to 2 and the *numEvents* vector will contain 1, 2, 1 (one event with no heads, two events with one head, and one event with two heads). The probability of heads can be set to any value between 0 and 1.

A second loop codes all of the rounds of tosses into a binary coding scheme in order to count the number in each group. The results are now put into matrices in order to print. The **table** and **factor** functions are invaluable in sorting categorical data for summarizing and reporting. The first matrix contains: (1) row labels with the possible number of heads that could be obtained (0 to *numCoins*); (2) the frequency of each round of flips that obtained that number of heads divided by the total number of rounds (giving the frequency); and (3) the theoretical probability of obtaining that many heads. The second matrix contains: (1) row labels with the event labels (HH, TH, HT, TT); (2) the frequency of each event obtained during all rounds; and (3) the theoretical probability of obtaining each event. The last two lines of the program prints out the matrices. The number of coins selected should not exceed 5 and sample sizes larger than 5,000 will require more time for the program to run.

Given these values:

pH <- 0.5
numCoins <- 2
N <- 100

```
         Sample %  Population %

0 Heads      0.28          0.25
1 Heads      0.54          0.50
2 Heads      0.18          0.25

         Sample %  Population %

TT           0.28          0.25
HT           0.33          0.25
TH           0.21          0.25
HH           0.18          0.25
```

Given these values:

```
pH <- 0.5
numCoins <- 3
N <- 100
```

```
          Sample %  Population %

0 Heads      0.17         0.125
1 Heads      0.42         0.375
2 Heads      0.35         0.375
3 Heads      0.06         0.125

          Sample %  Population %

TTT          0.17         0.125
HTT          0.19         0.125
THT          0.11         0.125
HHT          0.13         0.125
TTH          0.12         0.125
HTH          0.11         0.125
THH          0.11         0.125
HHH          0.06         0.125
```

JOINT PROBABILITY Exercises

1. Run JOINT program with pH = 0.50 and *numCoins* = 2 for the following sample sizes: 100, 1000, and 5,000. Complete the table.

RELATIVE FREQUENCY				
EVENT	N = 100	N = 1,000	N = 5,000	PROBABILITY (P)
TT	____	____	____	____
HT	____	____	____	____
TH	____	____	____	____
HH	____	____	____	____
HEADS	____	____	____	____
0	____	____	____	____
1	____	____	____	____
2	____	____	____	____

a. Compare the relative frequency of TT, HT, TH, and HH with the probability of these events. Do the relative frequencies provide a reasonable approximation to the probabilities?

Yes______ No ______

b. For which sample size does the relative frequency give the best approximation?

N = 100 _______ N = 1,000 _______ N = 5,000 __________

c. Under HEADS, a value of 1 gives the joint probability for HT and TH.

P(1) = P(HT) + P(TH) by the addition law. Compute P(1) for each sample size.
N = 100 _________ N = 1,000 ________ N = 5,000 _________

d. Show that the same is true for the frequency.

F(1) = F(HT) + F(TH) by the addition law. Compute F(1) for each sample size.

Note: F = P*N

N = 100 ________ N = 1,000 _________ N = 5,000 _________

2. From the previous table for N = 100, compute the ERROR by subtracting the probability from the proportion.

Note: ERROR = SAMPLE % − POPULATION %. Keep the +/− sign for each error.

EVENT	SAMPLE %	POPULATION %	ERROR
TT	_________	______________	_________
HT	_________	______________	_________
TH	_________	______________	_________
HH	_________	______________	_________
HEADS	_________	______________	_________
0	_________	______________	_________
1	_________	______________	_________
2	_________	______________	_________

a. Is the ERROR under HEADS for a value of 1 related to the errors for HT and TH?

YES ______ NO _____

b. What is the sum of the ERRORS for the four events? ______________

Addition Law of Probability

We will use the computer to simulate the rolling of two dice to compare the relative frequencies of the sums of the numbers on the two dice with corresponding theoretical probabilities. This will show how the theoretical probabilities for the sums are computed.

We will use the addition law of probability to find the probability of an even sum and the law of complements to find the probability of an odd sum.

Probability can help determine the odds of events occurring in practice. For example, a deck of cards contains 52 cards. A deck of cards has four suits (Hearts, Diamonds, Spades, and Clubs). Therefore each suit has 13 cards ($4 \times 13 = 52$). The probability of selecting any Heart from a deck of cards would be $13/52 = 0.25$. This would be the same probability for selecting any Diamond, Spade, or Club, assuming selection with replacement of the card each time. Similarly, there are four Kings (one in each suit). The probability of selecting a King out of a deck of cards would be $4/52 = 0.076923$.

The sample space for the sum of the numbers on the two dice can be conceptualized as follows:

		FIRST DIE					
	+	1	2	3	4	5	6
SECOND DIE	1	2	3	4	5	6	7
	2	3	4	5	6	7	8
	3	4	5	6	7	8	9
	4	5	6	7	8	9	10
	5	6	7	8	9	10	11
	6	7	8	9	10	11	12

If the dice are unbiased, all of the 36 outcomes are equally likely. The probability of any sum, S, can be calculated theoretically by the formula: P(S) = (Number of ways S can occur)/36. For example, $P(7) = 6/36 = 1/6$ (a number 7 occurs in the diagonal six times).

The theory of probability relates to possible outcomes of events occurring in a sample space. The relative frequencies for the different sums are not readily apparent. Our earlier approach assumed that all events were equally likely. In the dice example, we discover that the sums have 36 outcomes, which are equally likely outcomes, but certain sums occur more frequently (e.g., sum = 6). The theory of probability helps us to understand the frequency of outcomes and apply this in practice.

The theoretical probabilities for the sums of the numbers on two dice agree well with what happens in practice. The theoretical probabilities for the sums can be found by listing all of the outcomes in a two-way table and using the "equally likely" definition of probability; for the sum S, P(S) = (Number of ways S can occur)/36. The relative frequencies of the sums get very close to the theoretical probabilities given large sample sizes. The probability of an odd sum can be found from the probability of an even sum by using the **law of complements**: P(ODD) = 1 − P(EVEN).

ADDITION R Program

The ADDITION R program simulates the tossing of two dice. It records the number of times that each of the possible sums of the two dice occurs, and then changes these counts into relative frequencies. The relative frequencies of each sum for each sample size, along with the theoretical probability, are printed. Since the computation of the theoretical probabilities depends on the "equally likely" definition of probability, the exercises illustrate how the definition of probability is reasonable and does reflect practice. The different events are examined simultaneously, but are independent.

The program starts with a vector of sample sizes and then creates a vector of probabilities that correspond to the chances of obtaining a 2 through 12 from rolling two dice. The *DiceFreq* object is set to **NULL** so that it may be used to build a matrix within the main processing loop. The loop iterates through the values in *SampleSizes* and obtains a random sample of values from 1 to 6 of size *SampleSize* for *Die1* and then repeats the process for *Die2*. The two vectors of simulated rolls are summed together to obtain a vector for the total of both dice. [Note: the same vector could have been obtained by removing the *Die1* and *Die2* variables and just typing *DiceSum* <- **sample**(2:12,**size**=*SampleSize*,**replace**=T), but that hides the fact that we have two independent events and destroys the chance to analyze specific dice combinations.] The relative frequency of each outcome (2 through 12) is appended to the *DiceFreq* matrix for each different sample size as the loop continues through the values of the *SampleSizes* vector. Finally, the *outputMatrix* is built from the *DiceFreq* matrix, **cbind** is used in the *Probs* vector to yield a matrix with relative frequencies for each value outcome of the dice, for each sample size, along with the theoretical probabilities of obtaining each value outcome. The **print** function is used to output the information. Run the ADDITION program using various sample sizes to see how closely you can approximate the theoretical probabilities.

ADDITION Program Output

```
  N= 100  N= 500   N= 1000   N= 5000    Prob.
2    0.04    0.034     0.036     0.0290  0.0278
3    0.06    0.064     0.068     0.0580  0.0556
4    0.08    0.072     0.078     0.0798  0.0833
```

5	0.13	0.112	0.092	0.1082	0.1111
6	0.08	0.142	0.130	0.1462	0.1389
7	0.12	0.188	0.172	0.1622	0.1667
8	0.13	0.116	0.145	0.1450	0.1389
9	0.14	0.114	0.127	0.1162	0.1111
10	0.15	0.088	0.085	0.0790	0.0833
11	0.04	0.042	0.049	0.0538	0.0556
12	0.03	0.028	0.018	0.0226	0.0278

ADDITION Law Exercises

1. Run ADDITION for the sample sizes indicated below. Complete the table.

RELATIVE FREQUENCY				
SUM	N=360	N=1,200	N=7,200	PROBABILITY
2	______	______	______	______
3	______	______	______	______
4	______	______	______	______
5	______	______	______	______
6	______	______	______	______
7	______	______	______	______
8	______	______	______	______
9	______	______	______	______
10	______	______	______	______
11	______	______	______	______
12	______	______	______	______

a. Check that the probabilities listed correspond to values in the sequence 1/36, 2/36, 3/36, 4/36, 5/36, 6/36, 5/36, 4/36, 3/36, 2/36, and 1/36 (sum = 1 within rounding).

b. Which sample size provides the best estimate of the probabilities? ______

2. The addition law for mutually exclusive events states that the sum of relative frequencies for *even* numbers should be about 50% and the sum of relative frequencies for *odd* numbers should be about 50%.

a. For N = 7,200 above, add the relative frequency for all even sums (2, 4, 6, 8, 10, 12) and the relative frequency for all odd sums (1, 3, 5, 7, 9, 11). Enter the two relative frequencies in the table below.

SUM	FREQUENCY	PROBABILITY
EVEN		0.50
ODD		0.50

b. Why do you expect these frequencies to be around 50%?

____

____

3. Using the sum of frequencies for all *even* numbers and all *odd* numbers, answer the following questions.

   a. How can the probability of all *odd* numbers be obtained from the probability of all *even* numbers?

   ____

   ____

   b. What is the name of this probability law?

   ____

Multiplication Law of Probability

One of the basic properties of **probability** is the multiplication law for independent events. For example, if two dice are tossed and the events A, B, and C occur as follows:

A: An odd number on the first die
B: An odd number on the second die
C: An odd number on both dice

then the **multiplication law** for independent events states that: P(Event C) = P(Event A) × P(Event B).

This multiplication law of probability reflects the properties of relative frequency in practice. If two dice are tossed a large number of times, the relative frequencies of events A, B, and C should approximate the probabilities of these events. Also, the product of the relative frequency of A times the relative frequency of B should approximate the relative frequency of C. This can be stated as: RelativeFrequency (Event C) ≈ RelativeFrequency(Event A) × RelativeFrequency(Event B). The multiplication law for independent events states that if two events A and B are independent, then P(A and B) = P(A) × P(B). The multiplication law for independent events is modeled on the behavior of relative frequency. For relative frequency, RelativeFrequency(A and B) is approximately RelativeFrequency(A) × Relative Frequency(B). As sample size increases, RelativeFrequency(A and B) tends to be closer to RelativeFrequency(A) × RelativeFrequency(B).

MULTIPLICATON R Program

The MULTIPLICATION R program simulates the tossing of two dice and records the frequency of an odd number on the first die, an odd number on the second die, and an odd number on both of the dice. The frequencies are then changed to relative frequencies, and the results are rounded to three decimal places. The program inputs the number of times the two dice are tossed. The program illustrates a comparison between relative frequency and the multiplication law for the probability of independent events. Since sample size can be changed, the effect of sample size on the relative frequency as it relates to the multiplication law can be observed.

The program is a modification of the ADDITION R program. The program reflects how the probability of both dice ending up odd relates to the probability of either of the dice being odd. It begins with a vector of sample sizes, creates a **NULL** object to become a matrix, builds a processing loop to iterate through the values of *SampleSizes*, and simulates the rolling of two dice. The next two lines calculate the relative frequency of odds in the first die and the relative frequency of odds in the second die. It does this using a modulo operator (%%). The modulo operator performs an integer division and returns only the remainder portion. This means that 13%%5 would equal 3, because 13 divided by 5 equals 2 with remainder 3. Performing a modulo operation with a 2 on each die result would give values of 0 for even numbers and 1 for odd numbers. The odd numbers are counted up using the **sum** function and then divided by the sample size to get the relative frequency of odds in the sample. Next, the modulo 2 result for both dice are added together, so if both were odd then the result would be 2, otherwise it would be 0 or 1. The integer number is divided by 2 (%/%), which only returns a whole number. This would result in 0 for values of 0 or 1, because 2 doesn't go into either of those numbers evenly, but would be 1 for a value of 2. In this way, the rolls in which both dice come up odd are added together and divided by the sample size to give the relative frequency of both being odd in the sample. These three values are **rbind**ed into *outputMatrix* along with the difference of the relative frequency of both being odd and the product of the relative frequencies of each being odd. After the loop is completed, dimension names are assigned to the rows and columns of the matrix and it is printed using the **print(matrix)** function. Different sample sizes can be input to see the effect sample size has on the relative frequencies and to observe what sample sizes are required to reduce the error to a minimum.

MULTIPLICATION Program Output

Given the following sample sizes: SampleSizes <- c(100,500,1000)

```
         1st Odd  2nd Odd  Both Odd   F1*F2    Error
 N= 100    0.480    0.460     0.210   0.221   -0.011
 N= 500    0.504    0.516     0.260   0.260    0.000
 N= 1000   0.513    0.510     0.259   0.262   -0.003
```

Given the following sample sizes: SampleSizes <- c(1000,2000,3000)

```
          1st Odd  2nd Odd  Both Odd   F1*F2    Error
N= 1000    0.491    0.502     0.251    0.246    0.005
N= 2000    0.486    0.493     0.230    0.240   -0.009
N= 3000    0.486    0.495     0.241    0.241    0.000
```

Multiplication Law Exercises

1. Run MULTIPLICATION for samples sizes 100, 500, and 1,000.
   Record the results below.

RELATIVE FREQUENCY					
SAMPLE SIZE	FIRST ODD	SECOND ODD	BOTH ODD	RF1*RF2	ERROR
N = 100					
N = 500					
N = 1,000					

   a. What is the theoretical probability that the first die will be odd?________________

   b. What is the theoretical probability that the second die will be odd?______________

   c. What is the theoretical probability that both dice are odd?________________

   d. What law of probability are you using to find the probability that both are odd? ____

   e. What effect does sample size have on the sample approximations? ________
   ____
   ____

   f. Compute the error in this approximation by: ERROR = BOTH ODD − (F1 × F2)

   Do all of the differences have the same sign?
   YES ________ NO __________

   g. Does sample size have an effect on the amount of error?

   YES ________ NO___________

2. Run MULTIPLICATION for samples sizes 1,000, 2,000, and 3,000.
   Record the results below.

RELATIVE FREQUENCY					
SAMPLE SIZE	FIRST ODD	SECOND ODD	BOTH ODD	RF1*RF2	ERROR
N = 1,000					
N = 2,000					
N = 3,000					

a. For N = 3,000, what is the relative frequency that the first die will be odd? ____________

b. For N = 3,000, what is the relative frequency that the second die will be odd? __________

c. For N = 3,000, verify that F1*F2 is correct. ____________________________________

d. Why is the relative frequency of BOTH ODD different from your answer in 2c? ____

e. Do all of the error terms have the same sign?

   YES ________ NO __________

f. Does sample size have an effect on the amount of error?

   YES ________ NO___________

Conditional Probability

A child has a toy train that requires six "C" batteries to run. The child has accidentally mixed four good batteries with two bad batteries. If we were to randomly select two of the six batteries without replacement, the odds of getting a bad battery are conditionally determined. Let's assume that the first battery chosen is bad (Event A) and the second battery chosen is good (Event B). The two selections of the two batteries are dependent events. The probability of event B has two different values depending upon whether or not event A occurs. If event A occurs, then there are four good batteries among the remaining five batteries, and the probability of event B is 4/5. If a battery is chosen and event A does not occur, then there are only three good batteries remaining among the five batteries, and the probability of B is 3/5.

In probability terms this can be represented by: $P(B|A) = 4/5$ and $P(B|\text{not-}A) = 3/5$. These terms are read, "the probability of B given A" and "the probability of B given not-A", respectively. Probabilities of this type are called **conditional probabilities** because the probability of B is conditional upon the occurrence or nonoccurrence of A. Conditional probabilities are related to joint probabilities and marginal probabilities. This relationship can be illustrated by the following example. Consider a sample space that contains all pairs of batteries selected from the six batteries without replacement. The X's in the table below indicate the 30 possible outcomes.

		SECOND BATTERY					
		DEAD1	DEAD2	GOOD1	GOOD2	GOOD3	GOOD4
	DEAD1		**X**	**X**	**X**	**X**	**X**
	DEAD2	**X**		**X**	**X**	**X**	**X**
FIRST BATTERY	GOOD1	**X**	**X**		**X**	**X**	**X**
	GOOD2	**X**	**X**	**X**		**X**	**X**
	GOOD3	**X**	**X**	**X**	**X**		**X**
	GOOD4	**X**	**X**	**X**	**X**	**X**	

Since these 30 outcomes are equally likely, the **joint probabilities**, P(A and B), can be summarized in the following table.

	JOINT PROBABILITIES	SECOND BATTERY		
		DEAD	GOOD	MARGINAL PROBABILITY
FIRST BATTERY	DEAD	2/30	8/30	10/30
	GOOD	8/30	12/30	20/30
MARGINAL PROBABILITY		10/30	20/30	30/30 (Total)

The row totals are the **marginal probabilities** for the first battery:

P(First is dead) = 10/30
P(First is good) = 20/30.

The column totals are the **marginal probabilities** for the second battery:

P(Second is dead) = 10/30
P(Second is good) = 20/30.

Conditional probabilities are related to these joint probabilities and marginal probabilities by the following formula:

$$P(B|A) = P(B \text{ and } A)/P(A).$$

If event A results in a bad battery and event B results in a good battery, then

$$P(B|A) = P(\text{Second is good} \mid \text{First is dead})$$
$$= P(\text{Second is good and First is dead}) / P(\text{First is dead})$$
$$= (4/15)/(5/15)$$
$$= 4/5.$$

These conditional probabilities are theoretical probabilities assigned to the events by making use of the definition of probability for **equally likely events** (it is assumed that each of the batteries and each of the pairs are equally likely to be chosen). If these conditional probabilities are reasonable, they should reflect what happens in practice. Consequently, given a large number of replications in which two batteries are selected from a group of six batteries (in which two of the batteries are dead), the relative frequencies for the conditional events should approximate the theoretical conditional probabilities.

The conditional probability of B given A is the joint probability of A and B divided by the marginal probability of A, if P(A)≠0. The conditional probability agrees with the behavior of relative frequency for conditional events. For large sample sizes, the relative frequency of a conditional event is a good approximation of the conditional probability.

CONDITIONAL R Program

The CONDITIONAL R program simulates random selection without replacement of two batteries from a group of six in which two of the batteries are dead. The number of replications can be specified in the program. The relative frequencies from a small number of replications will not provide good approximations of the probabilities. A large number of replications should provide relative frequencies that will be very close to the theoretical probabilities. The program will permit you to observe that the theoretical rules of probability for conditional events do in fact reflect practice. The theoretical probabilities are important in statistics because they provide reasonable rules to adopt for analyzing what happens in the real world.

The CONDITIONAL R program determines the probability of conditional events by selecting two batteries from a group of batteries with a certain number of good batteries and a certain number of bad batteries. The total number of batteries is assigned, followed by the number of bad batteries and the number of times the two selections should be replicated. The number of good batteries is determined by subtraction of the number of bad batteries from the total number of batteries. The probabilities of the possible event outcomes are then determined and assigned to the variables: *pGG*, *pBB*, *pGB*, and *pBG*.

The total number of batteries is defined so that sampling can occur from a finite population. The **rep** function creates *numGood* (number of 0s) and the **c** function concatenates those with *numBad* (number of 1s) to complete the population of batteries. The receiving objects *FirstBattery* and *SecondBattery* are set to **NULL** before the main processing loop begins. The loop takes a sample of two batteries WITHOUT replacement, since these are conditional events. The two batteries are then added to their respective vectors. They are also added to an *eventList* vector using the same type of binary encoding scheme presented in earlier chapters. The encoding is much simpler since the number of picks is fixed at two.

After the processing loop is finished, output matrices are created. The *eventTable* vector is built from the *eventList* vector factored for all the possible coded event values from 0 to 3. Two vectors of values are then created and put into the matrices for even column spacing in the output. The first vector is moved into a 4 by 4 matrix for display with no **dimnames** set, since the column and row headers were included within the vector. The same thing is done with the second vector, only it is printed out using **print(matrix) command**, since it doesn't have multiple column or row headers. The program permits different numbers of total and bad batteries, as well as different sample sizes.

CONDITIONAL R Program Output

Given the following values:

numBatteries <- 6
numBad <- 2
SampleSize <- 1000

```
                    Second   Battery
                    Bad      Good
  First    Bad      0.061    0.242
  Battery Good      0.277    0.42

  No. Bad    Rel Freq    Probability
  0          0.42        0.4
  1          0.519       0.534
  2          0.061       0.067
```

Given the following values:

numBatteries <- 6
numBad <- 2
SampleSize <- 5000

```
                    Second   Battery
                    Bad      Good
  First    Bad      0.071    0.272
  Battery Good      0.271   0.386

  No. Bad    Rel Freq    Probability
  0          0.386       0.4
  1          0.543       0.534
  2          0.071       0.067
```

CONDITIONAL Probability Exercises

1. Run CONDITIONAL for N = 1,000 with 6 total batteries and 2 bad batteries.

   a. Enter the probabilities of the joint events and the marginal probabilities in the table.

		Second battery Bad	Second battery Good	Marginal Probability
First Battery	Bad			
	Good			
Marginal Probability				(Total)

b. Do the marginal probabilities indicate that approximately 1/3 (0.33) of the batteries are bad and 2/3 (0.67) of the batteries are good?

YES _________ NO __________

c. Do the marginal probabilities sum to 1.0?

YES _________ NO __________

2. From the CONDITIONAL program with N = 1,000, enter the relative frequencies of 0, 1, and 2 bad batteries.

a. Compute the error and record it in the table.

ERROR = REL FREQ − PROBABILITY

No. BAD	REL FREQ	PROBABILITY	ERROR
0			
1			
2			

b. Some of the errors should be positive and others negative.

Do the errors sum to zero (0)? YES ________ NO ________

c. Do the relative frequencies sum to 1.0? YES ________ NO ________
d. Do the probabilities sum to 1.0? YES ________ NO ________

3. Run CONDITIONAL for N = 5,000 with 6 total batteries and 2 bad batteries.

a. Enter the probabilities of the joint events and the marginal probabilities in the table.

		Second Battery		Marginal probability
		Bad	Good	
First Battery	Bad			
	Good			
Marginal probability				(Total)

b. Do the marginal probabilities indicate that approximately 1/3 (0.33) of the batteries are bad and 2/3 (0.67) of the batteries are good?

YES _________ NO __________

c. Do the marginal probabilities sum to 1.0?

YES _________ NO __________

4. From the CONDITIONAL program with N = 5,000, enter the relative frequencies of 0, 1, and 2 bad batteries.

a. Compute the error and record it in the table.

ERROR = REL FREQ − PROBABILITY

No. BAD	REL FREQ	PROBABILITY	ERROR
0			
1			
2			

b. Some of the errors should be positive and others negative.

Do the errors sum to zero (0)? YES ________ NO ________

c. Do the relative frequencies sum to 1.0? YES ________ NO ________

d. Do the probabilities sum to 1.0? YES ________ NO ________

Combinations and Permutations

Probability theory helps us to determine characteristics of a population from a random sample. A **random sample** is chosen so that every object, event, or individual in the population has an equal chance of being selected. The probability that the object, event, or individual will be selected is based upon the relative frequency of occurrence of the object, event, or individual in the population. For example, if a population consisted of 1,000 individuals with 700 men and 300 women, then the probability of selecting a male is 700/1,000 or 0.70. The probability of selecting a woman is 300/1,000 or 0.30. The important idea is that the selection of the individual is a chance event.

Probability theory operates under seven fundamental rules. These seven rules can be succinctly stated as:

1. The probability of a single event occurring in a set of equally likely events is one divided by the number of events, i.e., P (single event) = 1/N. For example, a single marble from a set of 100 marbles has a 1/100 chance of being selected.
2. If there is more than one event in a group, then the probability of selecting an event from the group is equal to the group frequency divided by the number of events, i.e., P(Group|single event) = group frequency/N. For example, a set of 100 marbles contains 20 red, 50 green, 20 yellow, and 10 black. The probability of picking a black marble is 10/100 or 1/10.
3. The probability of an event ranges between 0 and 1, i.e., there are no negative probabilities and no probabilities greater than one. Probability ranges between 0 and 1 in equally likely chance events, i.e., $0 \leq P\ (\text{event}) \leq 1$.
4. The sum of the probabilities in a population equal one, i.e., the sum of all frequencies of occurrence equals 1.0, i.e., $\Sigma(\text{Probabilities}) = 1$.
5. The probability of an event occurring plus the probability of an event *not* occurring is equal to one. If the probability of selecting a black marble is 1/10, then the

probability of *not* selecting a black marble is 9/10, i.e., P + Q = 1 where P = probability of occurrence and Q = 1 − P.

6. The probability that any one event from a set of mutually exclusive events will occur is the sum of the probabilities (addition rule of probability). The probability of selecting a black marble (10/100) *or* a yellow marble (20/100) is the sum of their individual probabilities (30/100 or 3/10), i.e., P(B or Y) = P(B) + P(Y).
7. The probability that a combination of independent events will occur is the product of their separate probabilities (multiplication rule of probability). Assuming sampling with replacement, the probability that a yellow marble will be selected the first time (2/10) and the probability that a yellow marble will be selected the second time (2/10) combine by multiplication to produce the probability of getting a yellow marble on both selections (2/10 × 2/10 = 4/100 or 0.04), i.e., P(Y and Y) = P(Y)*P(Y).

Factorial notation is useful for designating probability when samples are taken *without* replacement. For example, a corporate executive officer (CEO) must rank the top five department managers according to their sales productivity. After ranking the first manager, only four managers are remaining to choose from. After ranking the second manager, only three managers remain, and so forth, until only one manager remains. If the CEO selects managers at random, then the probability of any particular manager order is: 1/5*1/4*1/3*1/2*1/1, or 1/120.

The probability is based upon the total number of possible ways the five managers could be ranked by the CEO. This is based upon the number of managers in the company available to select from, which changes each time. Consequently, the product yields the total number of choices available: 5*4*3*2*1 = 120. This product is referred to as **factoring** and uses **factorial notation** to reflect the product multiplication, i.e., n!. The factorial notation, 3! (read 3-factorial), would imply, 3*2*1, or 6, which indicates the number of different ways three things could be ordered. Imagine a restaurant that serves hamburgers with the following toppings: pickle, onion, and tomato. How many different ways could you order these ingredients on top of your hamburger?

Permutations involve selecting objects, events, or individuals from a group and then determining the number of different ways they can be ordered. The number of permutations (different ways you can order something) is designated as **n** objects taken **x** at a time, or:

$$P(n, x) = \frac{n!}{(n - x)!}$$

For example, if a teacher needed to select three students from a group of five and order them according to mathematics ability, the number of *permutations* (or different ways three out of five students could be selected and ranked) would be:

$$P(n,x) = \frac{n!}{(n-x)!} = \frac{5!}{(5-3)!} = \frac{5*4*3*2*1}{2*1} = 60$$

Probability can also be based upon the number of *combinations* possible when choosing a certain number of objects, events, or individuals from a group. The ordering of observations (permutations) is not important when determining the number of combinations. For example, a teacher must only choose the three best students with mathematics ability from a group of five (no ordering occurs). The number of possible combinations of three students out of a group of five is designated as:

$$P(n,x) = \frac{n!}{x!(n-x)!} \text{ or } \frac{5!}{3!(5-3)!} = 10$$

The number of possible combinations can be illustrated by determining the number of students in a classroom that have their birthday on the same day. This classic example can be used in a class to determine the probability that two students have the same birthday. The probability of two students having a common birthday, given five students in a class, can be estimated as follows (assuming 365 days per year and equally likely chance):

$$P(2|5) = 1 - \frac{365}{365} * \frac{364}{365} * \frac{363}{365} * \frac{362}{365} * \frac{361}{365} = 0.027$$

The numerator decreases by one because as each student's birthday is selected, there is one less day available.

The probability of at least two students out of five *not* having the same birthday is $1-P$ (see probability rule 5). The probability of *no* students having a birthday in common for a class of five students is computed as:

$$P(No2|5) = \frac{365}{365} * \frac{364}{365} * \frac{363}{365} * \frac{362}{365} * \frac{361}{365} = 0.973$$

Therefore, the probability of at least two students having the same birthday is the complement of *no* students having the same birthday, or $P(2|5) = 1 - 0.973 = 0.027$. The formula clearly indicates that this probability would increase quickly as the number of objects, events, or individuals in the group increases.

The seven rules of probability apply to everyday occurrences. The number of possible outcomes of independent events is designated as a factorial (n!). Permutations refer to the number of possible ways to order things when selected from a group (**n** objects, **x** order).

Combinations refer to the number of possible sub-groups of a given size from a larger group (**n** objects, **x** size). The birthday problem is a classic example of how to determine whether two individuals in a group of size N have the same birthdays. The relative frequencies produced by a simulation of the birthday problem are good approximations of the actual probabilities. The *accuracy* of the relative frequencies as approximations of the actual probabilities in the birthday problem is not affected by the size of the group of people, rather by increasing the number of repetitions in the program.

Combination and Permutation R Program

The Combination and Permutation R program simulates an example for N individuals by using a random number generator and checking for a common birthday. The relative frequency of at least one common birthday is reported. This relative frequency approximates the probability of occurrence in a group of N individuals. The size of the group and the number of replications can be changed.

The program simulates selecting groups of people of various sizes over a given number of replications in order to compute estimates of probabilities. The sizes of the groups of people are assigned to the vector *numPeople*. The *replicationSize* variable represents the number of times that the selection of random birthdays will occur for each group size. In the initial program settings, the probability of five birthdays in common is chosen, and duplication or non-duplication reported for 250 replications. The *numPeople* vector then indicates that 10 birthdays will be chosen at a time for the 250 replications. This is repeated for 20 and 50 birthdays. The sampling and replications can be time-consuming for a computer, so it would be wise *not* to select 10,000 replications for a *numPeople* vector of from 1 to 100, unless you are willing to wait.

The *repeatVector* object is simply a vector containing the number of times there was a common birthday for a given group size. The outer processing loop iterates through the values of group sizes and the inner processing loop defines each replication for the given group size. Within this inner loop, a random sample is taken from the range of values from 1 to 365 with replacement and a sample size of *numPeople*[*i*]. The vector created is run through the **table** function to group the sample points that fell on the same day, and if the **max** of that table is greater than 1, then it means there was at least one birthday in common. If this occurs, then the corresponding value in the vector for the number of replications containing repeated values is increased by one. Because this takes place within the inner-processing loop, it continues for all replications of all group sizes.

Once the simulation in the loops is concluded, the counts within the *repeatVector* are changed into relative frequencies by dividing the *repeatVector* by the replication size. This creates a new vector of values that contains the relative frequencies of replications with birthday duplications. Theoretical probabilities are computed using a small processing loop that iterates through the group sizes and creates a vector of probabilities of duplication for each group size. The single line of code within the loop represents the mathematical notation, 1 – (365/365)*(364/365)*(363/365) …((366 – group size)/365).

A **list** object is created to hold the dimension labels for the output matrix. The output matrix is built by concatenating the relative frequency vector, the theoretical probability vector, and the difference between the relative frequency vector and the theoretical probability vector, giving an error vector. The **dimnames** keyword is given the value of the **list** object that was created making the line easier to read. The matrix is output using **print(matrix)command**. Values are reported within three decimal places using the **nsmall** keyword of the **format** function.

Combination and Permutation Program Output

Given these values:

numPeople <- c(5,10,20,50)
replicationSize <- 250

```
       Rel. Freq.  Common   Birthday Error
N= 5     0.032       0.027           0.005
N= 10    0.104       0.117          -0.013
N= 20    0.428       0.411           0.017
N= 50    0.988       0.970           0.018
```

Given these values:

numPeople <- c(5,10,20,50)
replicationSize <- 500

```
       Rel. Freq.  Common   Birthday Error
N= 5     0.040       0.027           0.013
N= 10    0.118       0.117           0.001
N= 20    0.418       0.411           0.007
N= 50    0.970       0.970           0.000
```

Combination and Permutation Exercises

1. Run BIRTHDAY for the following sample sizes and complete the table.

GROUP SIZE	REL. FREQ.	COMMON BIRTHDAY	ERROR
N=5	______	______	______
N=10	______	______	______
N=15	______	______	______
N=20	______	______	______

a. As the size of the group increases, does the probability of a common birthday

GROUP SIZE	REL. FREQ.	COMMON BIRTHDAY	ERROR
N = 5	____________	___________	______
N = 10	____________	___________	______
N = 15	____________	___________	______
N = 20	____________	___________	______

increase?

YES __________ NO __________

b. As the size of the group increases, do the relative frequencies more closely approximate the common birthday probabilities? Hint: Does error decrease?

YES __________ NO __________

GROUP SIZE	REL. FREQ.	COMMON BIRTHDAY	ERROR
N = 10	____________	___________	______
N = 20	____________	___________	______
N = 30	____________	___________	______
N = 40	____________	___________	______
N = 50	____________	___________	______

2. Run BIRTHDAY again for the same sample sizes. Complete the table.

a. Are the common birthday probabilities the same?

YES __________ NO __________

b. Are the relative frequencies close to the common birthday probabilities?

YES __________ NO __________

3. Run BIRTHDAY again using sample sizes listed below with 500 replications. Complete the table.

a. As the size of the group increases, does the probability of a common birthday increase?

YES __________ NO __________

b. As the size of the group increases, do the relative frequencies more closely approximate the common birthday probabilities? Hint: Does error decrease?

YES __________ NO __________

True or False Questions

Finite and Infinite Probability

T F a. As additional trials are conducted, the relative frequency of heads is always closer to 0.5 than for any previous smaller sample size.

T F b. As sample size increases, the relative frequency of an event approaches a fixed value.

T F c. The relative frequency of an event with probability of 0.65 stabilizes faster than an event with probability of 0.10.

T F d. In understanding probability, it is assumed that the relative frequencies approach a fixed number as the sample size increases because this corresponds to our experience of the real world.

T F e. The relative frequency of an event in one hundred trials is the probability of the event.

Joint Probability

T F a. The addition and multiplication laws of probability are reasonable because these properties are true for frequencies.

T F b. If two events are independent, the addition law is used to find their joint probability.

T F c. The sum of the probabilities for all of the events in a sample space is 1.

T F d. $P(1) = P(HT) + P(TH)$ because HT and TH are independent events.

T F e. $F(HT) \approx P(H)*P(T)$ because H and T are independent events.

Addition Law of Probability

T F a. Probabilities would be the same if the dice were biased.

T F b. Since there are 12 distinct sums and 6 of them are even, the probability of an even sum is 6/12.

T F c. The stabilizing property of relative frequencies is true for a group of outcomes as well as for a single outcome.

T F d. Large numbers of repetitions will provide good estimates of probabilities.

T F e. Each time the program is run for N = 1,200, the relative frequencies will be the same.

Multiplication Law of Probability

T F a. If two events are independent, then the probability of both events occurring is the product of their probabilities.

T F b. If two events are independent, then the relative frequency of both events occurring is the product of their relative frequencies.

T F c. In general, relative frequencies obtained from small samples give the best approximations of probabilities.

T F d. If two biased dice were tossed, then the events FIRST ODD and SECOND ODD are not independent.

T F e. The events FIRST ODD and BOTH ODD are mutually exclusive events.

Conditional Probability

T F a. If two batteries are selected with replacement from a group of six batteries, in which two of the batteries are bad, the FIRST BAD and the SECOND GOOD are dependent events.

T F b. $P(A \text{ and } B) = P(A) \times P(B|A)$

T F c. If the probability of event A is not affected by whether or not event B occurs, then A and B are independent events.

T F d. P(A), the marginal probability for event A, is equal to the sum of the joint probabilities of A and all other events that can occur with A.

T F e. If two events A and B are independent, then $P(B|A) = P(B)$.

Combination and Permutation

T F a. As the size of the group increases, the probability decreases that two people have the same birthday.

T F b. The probability of *no* common birthday is the complement of the probability of having a common birthday.

T F c. If a group consists of only two people, the probability that they have the same birthday is 1/365.

T F d. In a group of 50 people, there will *always* be at least two people with the same birthday.

T F e. The error in the relative frequency as an approximation of the probability is reduced for large groups of people.