$$P(\text{hitting streak} \geq 35) \approx \frac{7}{10000} = 0.0007.$$

For a hitter of Utley's ability, the estimated chance of observing a hitting streak at least 35 games is only 0.0007. This conclusion should be made with some caution, since we chose Utley's hitting record since he had a long hitting streak. The chance that *some* hitter during the 2006 baseball would have a hitting streak of 35 or more games is a larger probability, since there are more outcomes involved in the computation.

Exercises

11.1 (Playing Roulette). Suppose one plays roulette repeatedly at a casino. In a single play, one bets \$5 on "red"; the player wins \$5 with probability 18/38 and loses \$5 with probability 20/38. If the roulette game (with the same bet) is played 20 times, then the individual play winnings can be viewed as a sample of size 20 selected with replacement from the vector (5, −5), where the respective probabilities are given in the vector (18/38, 20/38). These play winnings can be simulated using the function `sample` with the `prob` vector that gives the sampling probabilities.

```
sample(c(5, -5), size=20, replace=TRUE,
  prob=c(18 / 38, 20 / 38))
```

a. Write a short function to compute the sum of the winnings from 20 plays at the roulette wheel. Use the `replicate` function to repeat this "20 play simulation" 100 times. Find the approximate probability that the total winning is positive.
b. The number of winning plays is a binomial random variable with 20 trials where the probability of success is 18/38. Using the `dbinom` function, find the exact probability that your total winning is positive and check that the approximate answer in part (a) is close to the exact probability.
c. Suppose you keep track of your cumulative winning during the game and record the number of plays P where your cumulative winning is positive. If the individual play winnings are stored in the vector `winnings`, the expression `cumsum(winnings)` computes the cumulative winnings, and the expression `sum(cumsum(winnings)>0)` computes a value of P. Adjust your function from part (a) to compute the value of P. Simulate the process 500 times and construct a frequency table of the outcomes. Graph the outcomes and discuss which values of P are likely to occur.

11.2 (Checking hats). Suppose that men in the old days wore only two types of hats, say black and grey, and hats of a particular type are indistinguishable. Assume 20 men with hats visit the restaurant and half of the men are wearing each type of hat. The hats are randomly mixed, and we are interested in the number of men who leave the restaurant with the correct hat.

a. Modify the function `scramble.hats` to compute the number of correct matches in this setting. (The only change is the definition of the vector `hats` – if one represents a black hat and a grey hat by 1 and 2, respectively, then `hats` consists of ten 1's and ten 2's.)
b. Using the function `replicate`, repeat this simulation for 1000 trials. Store the number of matches for the 1000 experiments in the vector `matches`.
c. From the simulated values, approximate the probability that 10 or more men receive the correct hats. Also, find the expected number of correct matches.

11.3 (Birthday problem). Suppose you have a class of n students and you're interested in the probability that at least two students in the class share the same birthday. If one assumes that each birthday is equally likely from the set $\{1, 2, ..., 365\}$, then collecting birthdays can be viewed as a sample of size n chosen with replacement from the set.

a. Write a function with argument n that samples n birthdays and computes the number of unique birthdays observed.
b. Using the `replicate` function, repeat this simulation 1000 times for the special case of $n = 30$ students.
c. From the output, approximate the probability that there is at least one matching birthday among the 30 birthdays.
d. It can be shown that the exact probability of a birthday match is given by

$$P(match) = 1 - \frac{{}_{365}P_{30}}{365^{30}},$$

where ${}_NP_k$ is the number of ways of arranging a sample of k of N distinct items. Compare the approximate probability in part (d) with the exact probability.
e. The R function `pbirthday`, with argument n, computes the (approximate) probability of at least one match among n birthdays. Use this function to check your answer in part (d).

11.4 (Streakiness). In the chapter, the focus was on the length of the longest streak in a binary sequence. Another way to measure streakiness in a sequence is the number of switches from 0 to 1 or from 1 for 0. For example, if the binary sequence is given by

```
0 1 0 0 0 1 0 0 1
```

there are three switches from 0 to 1 and two switches from 1 to 0 and so the total number of switches is equal to five. If `y` is a vector containing the binary sequence, then the R expression

```
sum(abs(diff(y)))
```

will compute the number of switches.

a. Construct a function `switches` that computes the number of switches for a binary vector y. Test this function by finding the number of switches in Chase Utley's game hitting sequence for the 2006 season.
b. By making a slight change to the function `random.streak`, construct a function that computes the number of switches for a random permutation of the 1's and 0's in the vector y.
c. Use the `replicate` function to repeat the random permutation in part (b) for 10,000 simulations. Construct a histogram of the number of switches for these 10,000 random sequences. Is the number of switches in Utley's sequence consistent with the values generated from these random sequences? Using the number of switches statistic, did Utley display unusually streaky behavior during this season?

11.5 (Collecting state quarters). In 1999, the United States launched the 50 State Quarters program where each of the 50 states was honored with a special quarter. Suppose you purchase 100 "state" quarters where each quarter is equally likely to feature one of the 50 states.

a. Write a function using the `sample` function to simulate the purchase of 100 quarters and record the number of unique quarters that are purchased.
b. Using the `replicate` function, repeat this process for 1000 purchases. Construct a table of the number of unique quarters you obtain in these 1000 simulations. Use this table to estimate the probability that you obtain at least 45 unique quarters.
c. Use the output from part (b) to find the expected number of unique quarters.
d. Suppose you are able to complete your quarter set by purchasing state quarters from a coin shop for \$2 for each quarter. Revise your function to compute the total (random) cost of completing the quarter set. Using the `replicate` function, repeat the quarter-purchasing process 1000 times and compute the expected cost of completing your set.

11.6 (How many students are excluded?). In [35], Frederick Mosteller mentioned the following interesting probability question that can be addressed by a simulation experiment. Suppose there is a class of 10 students and each student independently chooses two other class members at random. How many students will not be chosen by any other students?

Here is an outline how one can simulate this experiment, following Mosteller's suggestions:

- One represents the students by a vector consisting of the integers 1 through 10.

  ```
  students = 1:10
  ```

- One represents the outcomes of choosing students by a matrix `m` of 10 rows and 10 columns consisting of zeros and ones. The (i, j) entry in the matrix is equal to one if student i chooses student j. One can initially define the matrix with all zeros by the R code.

```
m = matrix(0, nrow=10, ncol=10)
```

- Student j will choose two other students from the vector `students` excluding the value j. This can be done by the `sample` function – the students who are sampled are stored in the vector `s`.

```
s = sample(students[-j], size=2)
```

- In the matrix `m`, one records the selected students by assigning the columns labeled `s` in the jth row with the value 1.

```
m[j, s] = c(1, 1)
```

- If the above procedure is repeated for each of the 10 students, then one can represent all selections by the matrix `m`. Here is the matrix for one simulation.

```
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
 [1,]    0    1    0    0    1    0    0    0    0     0
 [2,]    0    0    0    1    0    0    1    0    0     0
 [3,]    0    0    0    0    0    0    0    1    0     1
 [4,]    1    0    0    0    1    0    0    0    0     0
 [5,]    0    1    0    0    0    1    0    0    0     0
 [6,]    0    0    0    1    1    0    0    0    0     0
 [7,]    0    1    0    0    0    1    0    0    0     0
 [8,]    0    0    1    0    0    1    0    0    0     0
 [9,]    0    1    0    0    0    0    1    0    0     0
[10,]    1    0    0    0    1    0    0    0    0     0
```

By looking at the sum of the matrix over rows, one can see how many students were not chosen by any students. (In this example, one student was not chosen.)

a. Write a function `mosteller` that will run a single simulation of this experiment.
b. Using the `replicate` function, repeat this simulation for 100 iterations.
c. Use the `table` function to tabulate the number of students not chosen over the 100 experiments.
d. What is the most likely number of students not chosen? What is the approximate probability of this outcome?