	B	C
3	**Random**	
4	**Number**	**Letter**
5	=RAND()	=IF(B5<0.55,"C",IF(B5<0.99,"A","R"))

Figure 6.13

6.3 Three Famous Problems

In this section we show how simulations can be used to approximate the solutions to three famous problems in elementary probability: the *Monty Hall Problem*, the *Birthday Problem*. and *Buffon's Needle Problem*.

The Monty Hall Problem

In the famous game show *Let's Make a Deal*, hosted by Monty Hall, one of the games required a contestant to choose one of three doors. Behind one of the doors was a prize (like money or a car), and behind the other two doors were dummy prizes (like a donkey). Once the choice was made, Monty Hall would open up one of the un-chosen doors revealing one of the dummy prizes. The contestant was then given a choice to either switch to the remaining unopened door or keep the door that was already chosen. The contestant would get whatever "prize" was behind the door.

To illustrate this game, consider the scenario in Figure 6.14 where the real prize is behind door 2 (unbeknownst to the constentant) and the contestant chooses door 1. Monty would then open door 3 revealing a dummy prize. The contestant then had to decide whether to switch to door 2 (and consequently win the real prize) or don't switch (and consequently not win the real prize).

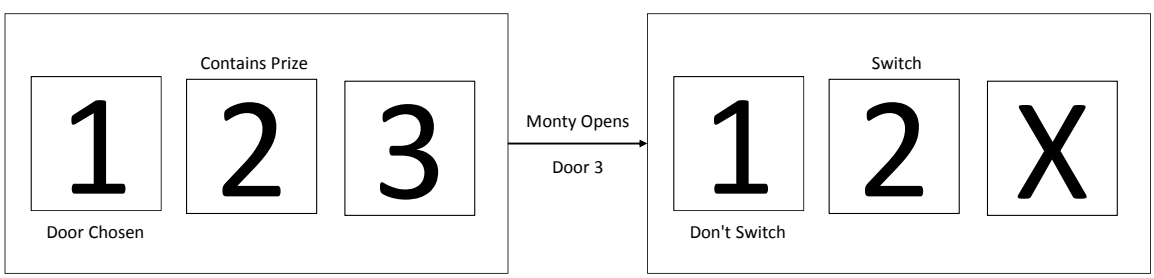

Figure 6.14

The dillemma facing the contestant is the decision of whether to switch or not. To help make this decision, we will use a simulation to estimate the following two probabilities:

1. Probability of winning if switching and

2. Probability of winning if not switching.

The key to estimating these probabilities is to realize that if not switching, the only way to win is to initially choose the door with the real prize. If not switching, the prize is won if the door with the real prize is not initially chosen.

Algorithm

1. Randomly designate the door with the real prize.
2. Randomly choose a door.
3. Determine if the real prize is won if switching.
4. Determine if the real prize is won if not switching.
5. Repeat for 1,000 trials.
6. Calculate $P(\text{win if switch}) \approx \dfrac{\text{number of times won when switching}}{\text{number of trials}}$.
7. Calculate $P(\text{win if don't switch}) \approx \dfrac{\text{number of times won when not switching}}{\text{number of trials}}$.

Rename a blank worksheet "**Monty**" and format it as in Figure 6.15. Copy row 8 down to row 1007 to perform 1000 trials.

	A	B	C	D	E
1		**Probabilities**			
2		**Win if**	**Win if don't**		
3		**switch**	**switch**		
4		=AVERAGE(D8:D1007)	=AVERAGE(E8:E1007)		
5					
6		**Door**	**Door**	**Win if**	**Win if don't**
7	**Trial**	**with prize**	**chosen**	**switch?**	**switch?**
8	1	=RANDBETWEEN(1,3)	=RANDBETWEEN(1,3)	=IF(B8<>C8,1,0)	=IF(B8=C8,1,0)

Figure 6.15

We see from the simulation that the probability of winning if switching is 2/3 and the probability of winning if not switching is 1/3. Thus the contestant should switch. This does not guarantee that the contestant will win, but the contestant is twice as likely to win if switching than if not switching.

The Birthday Problem

In a class of n students, what's the probability that at least two students will share a birthday (month and day)? This famous problem is known as the *birthday problem*. We assume that birthdays are uniformly distributed throughout the year (i.e. no day is more or less likely to be a birthday than any other day) and we will ignore leap years.

Algorithm

1. Randomly generate an integer between 1 and 365 for each student in the class to represent birthdays (1 = January 1, 2 = January 2, etc.).
2. For each day of the year, count the number of students in the class that have that day as their birthday.
3. Determine if some birthday is shared by at least two students. This is considered a "success."
4. Repeat for 200 trials.
5. Determine the number of successes.
6. Calculate $P\,(\text{at least two people sharing a birthday}) \approx \dfrac{\text{number of successes}}{\text{number of trials}}$.

To implement this algorithm, follow these steps:

1. Rename a blank worksheet "**Birthday**" and format it as in Figure 6.16. Copy row 3 down to row 102 to simulate a class of up to 100 students.

	A	B
1	**# Students =**	23
2	**Student**	**Birthday**
3	1	=IF(A3<=$B$1,RANDBETWEEN(1,365),0)

Figure 6.16

2. Add the formulas in Figure 6.17 and copy row 4 down to row 367. These formulas count the number of students who have a birthday on each day of the year and determine if the trial is a success.

	D	E
1	**Success?**	=IF(COUNTIF(E3:E367,">=2")>=1,1,0)
2	**Day**	**Count**
3	1	=COUNTIF($B$3:$B$102,D3)
4	=D3+1	=COUNTIF($B$3:$B$102,D4)

Figure 6.17

3. Add the formulas in Figure 6.18 to set up a table to store the results of 200 trials and calculate the estimated probability. Copy row 6 down to row 204. Create a table in the range **G4:H204** to store the results from 200 trials. Select **F1** as the column input cell. Press **F9** to repeat the simulation several times. Note that for a class of 23 students the simulation gives a probability of approximately 0.50.

	G	H
1	**# Successes**	=SUM(H5:H204)
2	**Probability**	=H1/200
3	**Trial**	**Success?**
4		=E1
5	1	
6	=G5+1	

Figure 6.18

One benefit of using a simulation is that we can easily modify it to estimate more complicated probabilities. For instance, if we wanted to estimate the probability that at least 3 students share a birthday in a class of 50 students, we could simply change the value of "# students" to 50 and modify the formula to determine if the trial is a success as in Figure 6.19.

	D	E
1	**Success?**	=IF(COUNTIF(E3:E367,">=3")>=1,1,0)

Figure 6.19

Buffon's Needle Problem

A version of this problem was first solved by the French naturalist and mathematician, the Comte de Buffon (1707-1788). Suppose we randomly drop a needle of length $L \leq 1$ on a wood floor in which the joints between the planks are 1 unit apart. Find the probability that the needle "hits," or intersects, one of the joints.

Let x denote the distance of the midpoint of the needle to the nearest joint between the planks and θ denote the angle as illustrated in Figure 6.20. Note that $0 \leq x \leq 0.5$ and $0 \leq \theta \leq \pi/2$.

To determine if the needle intersects a joint, observe that this can only occur if the hypotenuse of the right triangle in Figure 6.20 is less than $L/2$ (if the needle does not intersect a joint, then the needle must be "extended" to form the triangle, making the hypotenuse more than $L/2$). If h denotes the length of the hypotenuse, then using trigonometry we

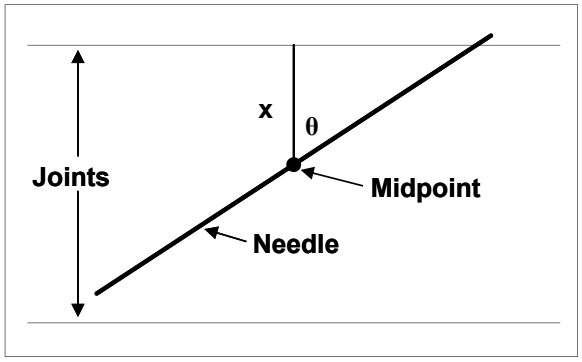

Figure 6.20

have

$$\cos\theta = \frac{x}{h} \quad \Rightarrow \quad h = \frac{x}{\cos\theta}.$$

Thus the needle intersects a joint if $x/\cos\theta < L/2$.

Algorithm

1. Randomly generate a number x between 0 and 0.5.
2. Randomly generate a number θ between 0 and $\pi/2$.
3. Calculate $h = x/\cos\theta$.
4. Determine if $h \leq L/2$. If this is true, then the trial is a "success."
5. Repeat for 1000 trials.
6. Calculate $P\,(\text{intersecting a joint}) \approx \dfrac{\text{number of successes}}{\text{number of trials}}$.

Rename a blank worksheet "**Buffon**" and format it as in Figure 6.21. Adjust the value of L and observe that as L gets smaller, so does the probability.

	A	B	C	D	E
1	**L =**	1	**Probability =**	=AVERAGE(E4:E1003)	
2					
3	**Trial**	**x**	**θ**	**h**	**Hit Joint?**
4	1	=0.5*RAND()	=PI()/2*RAND()	=B4/COS(C4)	=IF(D4<=$B$1/2,1,0)

Figure 6.21

Exercises

6.3.1 Consider the Monty Hall problem.

a. In the simulation, the "door chosen" was random. What if the contestant always chooses door 1? Does this change either of the probabilities of winning? What if door 2 is always chosen? What about door 3?

b. Consider the following strategy:

> Always initially choose door 1. If door 2 is opened, then switch. Otherwise don't switch.

Create a simulation to estimate the probability of winning if this strategy were used every time. Does this strategy do any better than switching every time?

6.3.2 Consider a generalization of the Monty Hall problem where there are N doors, exactly one of which contains the real prize. Assume the game is played the following way:

1. The contestant always initially chooses door 1.
2. If the door with the prize is door 1 or door N, then door 2 is opened. In this case, if the contestant switches, the door switched to is randomly selected between 3 and N, inclusive.
3. If the door with the prize is not door 1 or door N, then door N is opened. In this case, if the contestant switches, the door switched to is randomly selected between 2 and $N-1$, inclusive.

Create a simulation of this game where the user can input the value of N and estimate the probabilities of winning if switching and if not switching. (**Suggestion**: Create a column to determine the door switch to. Use this to determine if the game is won if a switch is made.)

a. Create a table which gives the probabilities for values of N between 3 and 25.

b. What happens to the difference between the probabilities of winning if switching and the probability if not switching as N gets larger? For "large" values of N, does it really matter if the contestant switches or not?

6.3.3 Consider the birthday problem and let P_n denote the probability that in a class of n students, at least 2 students share a birthday. It can be shown that P_n satisfies the following recursive relationship:

$$P_n = 1 - (1 - P_{n-1}) \cdot \frac{N - (n-1)}{N}$$

where N is the number of days in a year and $P_1 = 0$.

a. Create a spreadsheet to calculate the values of P_n for n between 1 and 100. Allow the user to enter the value of N. If $N = 365$, what is the value of P_{23}? Does this agree with the results of the simulation?

b. Graph P_n vs n when $N = 365$. For what values of n is $P_n > 0.90$?

c. Find that probability that in a class of 6 students, at least 2 students share a birth *month*.

6.3.4 Modify the worksheet **Birthday** to estimate the solution to the following generalization of the birthday problem: If a teacher asks a class of n students to write down an integer between a and b, what's the probability that at least m of them will write down the same number.

1. Your simulation should allow the user to input the values of n, a, b, and m, and automatically calculate the results.
2. Use a total of 500 trials.
3. Assume that $n \leq 200$, $0 \leq a < b \leq 365$, and that the students' choices are uniformly distributed.
4. If the value of m is in cell **E1**, then consider modifying the formula to determine a success as in Figure 6.22.

	D	E
2	**Success?**	=IF(COUNTIF(E4:E368,">="&E1)>=1,1,0)

Figure 6.22

6.3.5 For Buffon's needle problem, create a table which gives the probability of an intersection for different values of L between 0 and 1. Create a graph of the probability vs L, fit a curve to the data, and use the curve to hypothesize the theoretical relationship between the probability and L. (**Hint**: It is a very simple relationship involving the number $2/\pi$.)

6.3.6 Two numbers x and y are randomly chosen from the interval (0, 1). Design a simulation to estimate the probability that the closest integer to y/x is even. (**Hint**: Calculate y/x and round it off to a whole number using the **ROUND** function. To determine if this number is even, use the **MOD** function.)

6.4 Random Number Generators

In this section we will look at how computers generate "random" numbers. Random numbers are an essential part of all computer simulations. A simple definition of a **list of random numbers** is that it is a list of numbers in which there is ***no*** pattern and all possible numbers occur with equal frequency. The only way to get a truly random list of numbers is by mechanical means (i.e. numbered balls tumbling in a cage, rolling a die, etc.).

A computer generates a list of "random" numbers by using an iterative function where one output becomes the next input. The initial input, called the *seed*, is arbitrary (it is often chosen according to the clock time at which the algorithm begins) and each output becomes a number in the list.

Because the computer uses a deterministic algorithm, there *will* be a pattern to the list of numbers. Therefore, computers can never generate a true list of random numbers. The "random" numbers they generate are called **pseudorandom** numbers and the algorithm