MINITAB is perhaps one of the most popular statistical analysis software packages in the world. It is powerful and yet easy to learn and use. The availability of inexpensive student editions of the software has made MINITAB even more accessible in recent years. Many books on elementary and advanced statistics have now included computer assignments using MINITAB.

This book however is not about using MINITAB as a statistical analysis package. The intent of this guide is to extend the use of MINITAB beyond standard statistical data analysis. This guide is about using MINITAB in an entirely different way – as a laboratory for conducting simulation experiments to solve probability problems. For example, simulating the toss of a fair die to compute the probability of obtaining at least a 6 in four tosses; or obtaining an approximate answer using simulation to the well-known birthday problem.

Standard MINITAB routines cannot answer the kind of problems listed directly. However, its powerful macro facility and available commands allow one to write simple short macros to easily conduct probabilistic experiments that will provide quick answers to problems like those above. However before one can write the macro, how an experiment is to be carried out must first be thought out carefully. This often provides excellent insights into the problem being studied. It also provides one with a better understanding of random sampling, probability distributions, expected values, and statistical concepts like p-values, independence, etc.
Students, teachers, and other users of probability and statistics can use this book as a supplementary text or as a laboratory manual for courses dealing with probability and statistics. It can also be used as a guide for those who are already familiar with probability and statistics and wants to use the techniques discussed in the book for conducting their own probabilistic experiments.

This book has been successfully used since 1995 by the author as a laboratory manual for an undergraduate course in probability and statistics for engineers and as supplementary notes for a graduate course in statistical methods for engineers.

Leonard M. Lye
# TABLE OF CONTENTS

PREFACE  
TABLE OF CONTENTS  
1. INTRODUCTION  
2. BASIC PROBABILITY PROBLEMS  
   2.1 Birthday Problem  
   2.2 Alexander’s Dilemma  
   2.3 De Mere’s Die Tossing Problem  
   2.4 Exercises  
3. FURTHER PROBABILITY PROBLEMS  
   3.1 Catching the Counterfeiter and Risk Analysis  
   3.2 Catching a Greedy Counterfeiter  
   3.3 Basketball Problem  
   3.4 Case of the Defective Concrete  
   3.5 Winning the Lottery  
   3.6 Meeting for Lunch  
   3.7 Exercises  
4. MORE ADVANCED PROBABILITY PROBLEMS  
   4.1 The Archer’s Problem  
   4.2 Flying Off in Time  
   4.3 Beware of Icebergs  
   4.4 Rainy Weeks and Total Rainfall
4.5 Exercises 66

5. REFERENCES 67
1. INTRODUCTION

Whenever we study science and engineering subject such as Physics, Chemistry, Biology, or Fluid Mechanics, there is always a laboratory component of the course where we can verify for ourselves using experiments, the theories or phenomena that were discussed in class. It is not that we do not believe what is taught, but it is more for enhancing our understanding and to see with our own eyes and feel with our own senses what the theories or phenomena are about. If what we have observed in the laboratory agrees with what is predicted by theory, then definitely we feel reassured that the theory is correct. At times when no theory or no analytical solution is available one must resort to experimentation to obtain an answer.

The study of probability and statistics should be similar to the study of any scientific subject with a laboratory component. The laboratory component would provide the student with the opportunity to verify using experimentation, the appropriateness of probability formulae used, check the effect of different assumptions on analytical results, or even come up with new statistical testing procedures.

A major problem in conducting probabilistic experiments is of course the time it takes for us to collect a reasonable number of samples to analyze. For example, consider the problem of trying to determine the probability of obtaining at least a six in 4 tosses of a fair die using experimentation:

*Take a fair die and toss it 4 times. Note whether a six has turned up in the four tosses. If at least one six is observed, then count it as a “yes” or “1”, if no six is observed in any of the four tosses, then count it as a “no” or “0”. For a fairly reliable answer, the procedure must be repeated N times where N is at least a hundred or even a thousand. The probability of obtaining at least a six in four tosses of a die is then the number of “yes’s” or “1’s” divided by N.*
Doing the above experiment using a real die would take several hours at least! A faster way is to use write a computer program to simulate the die tossing experiment. A program written in BASIC would look something like:

```basic
RANDOMIZE TIMER
INPUT "Number of trials"; N
S=0
For I=1 to N
   For J=1 to 4
      U=INT(RND*6+1)
      If U=6 THEN 100
   Next J: GOTO 200
100   S=S+1
200   NEXT I
PRINT "Probability of at least 1 six in 4 tosses in a die ="; S/1000
END
```

The above program would take only seconds to run for N=1000 on a personal computer. While there is nothing wrong with writing a computer program to perform the simulation, many students or users, however, of probability and statistics are not “up to speed” in computer programming. This is especially true when one considers more complicated problems that require graphical displays or special routines for the generation of random numbers. Then instead of learning about probability and statistics, one is bogged down with the programming.

Since a statistical analysis software package like MINITAB is often used in a probability and statistics course to enhance the learning of the subject and to take the drudgery of hand computations, why not use MINITAB for the simulation experiments? This is indeed possible and in fact rather simple to implement. Consider the above die tossing problem. The macro shown below (DICE) will perform the die tossing simulation.
Macro ‘DICE’
GMACRO
DICE
DO k1=1:1000
SAMPLE 4 c1 c2;
REPLACE.
LET c3(k1)=SUM(c2)
ENDDO
LET c4=(c3>=1)
LET k2=SUM(c4)/1000
NOTE: THE PROBABILITY OF OBTAINING AT LEAST A SIX IN FOUR
TOSSES OF A DIE IS k2
PRINT k2
ENDMACRO

Before running the macros, first enter in column C1 of the MINITAB worksheet
five 0’s and one 1. The 1 represents the six on the die. To run the macros, just type %
DICE. It is assumed that the macro created using a text editor such as Notepad has been
saved with a .mac extension in the macro sub-directory of the MINITAB main directory.
The number of trials used is 1000 here. More trials can be carried out if necessary. The
number of trials possible would depend on the version of MINITAB used. An interactive
version of the macro is given later.

One can see that the MINITAB macro for the above die tossing problem is rather
short and quite simple. For some problems, the MINITAB macros are in fact a lot more
direct and shorter than BASIC programs because they can make use of standard one to
two lines MINITAB commands for drawing histograms, scatter plots, regression,
drawing random samples of various distributions, probability plotting, etc.

Although MINITAB version 13 for the Windows OS was used for all problems in
this guide, the macros can be easily adapted to other versions of MINITAB on other
platforms.
This book contains thirteen Minitab macros for a variety of interesting probability problems in addition to those listed in the Preface. Many of the are well-known probability problems taken from the classic book of Frederick Mosteller (1965) “Fifty Challenging Probability Problems in Probability”, and some are similar to those used by Julian Simon (1992) in his delightful and radical book “Resampling: The New Statistics”. Exercises for the student to try on their own are also given.
2. BASIC PROBABILITY PROBLEMS

This chapter considers several fundamental probability problems using simulation. Most of these problems are classic probability problems and some are less well-known but still rather interesting from a teaching point of view. Where exact analytical solutions are available, they will be given for comparison with those obtained using simulation.

2.1 The Birthday Problem

The birthday problem is a classic problem in probability theory. It has even been a basis of interesting debate during the Tonight show! This is one of those counter-intuitive problems that confound most people.

Problem: What is the probability that in a randomly selected group of N people (say N=25 people), there will be at least 2 people with the same birthday?

Let’s consider a group of 25 people and let’s assume that there are 365 days in the year. This means that each person’s birthday will be on one of those 365 days in the year. What we are interested in is the probability that in this group of 25 people, at least 2 of them have birthdays falling on the same day. Most people would think that the probability is very small, perhaps even close to zero. The actual probability is of course much higher – more than 50% in fact for N=25.
**Solution via Simulation**

The birthday problem can be tackled using simulation. The procedure is as follows:
generate a set of N random integers between 1 and 365; count the number of duplicate integers (there may be none, or maybe one or more); repeat the last two steps over many trials e.g. 1000; then the number of trials, where there are one or more pairs with the same integers (same birthdays), divided by the total number of trials is the required probability.

On MINITAB, the above simulation is easily accomplished using the macro BIRTH.MAC. The macro reads in N and the number of trials to be performed. It also does the counting of the number of trials where one or more pairs of integers are the same, and calculates the required probability. The random integer generation, checking for pairs of duplicates in each trial is also done by this macro.

How do we check for duplicates on MINITAB? There is no MINITAB command that can do this task. Here a simple trick was used. After the N random integers are generated, they are then sorted in ascending order and ranked using the RANK command. If two of the integers are the same, they have equal ranks or tied ranks. If none of the integers are the same, then each has a unique rank. So if we take the absolute difference between the ranks and the values 1 to N (no tied ranks) and then sum it, a non-zero sum would indicate that there are duplicates. In fact, a non-zero sum can also indicate triplicates. If the sum is zero, then all integers are unique. The following example will clarify this trick. Let N=5. Let the 5 generated random integers be 3, 34, 89, 12, and 34.
The sum of 1.0 above indicates that there is one pair of integers with the same value (or there are two people with the same birthday). If three of the integers have the same value, say we have three 34’s (i.e. the integers are 3, 12, 34, 34 and 34), then the three 34’s will each have a rank of 4. Calculating the sum of the absolute differences between the ranks and the values 1 to N will also give a non-zero value. Therefore, as long as the sum is non-zero, would indicate that there is at least a pair of integers with the same value.

### Display 2.1: Sample MINITAB session with BIRTH macro

```
MTB>%BIRTH
Number of people and no. of trials? (e.g. 25 100)
DATA>25 500
Probability that at least 2 people share the same birthday is k2.
k2       0.580
MTB>
```
Display 2.2: BIRTH macro

```
BIRTH.MAC

GMACRO
BIRTH
NOTE Number of people and no. of trials? (e.g. 25 100)
SET c50;
FILE “terminal”;
NOBS 2.
COPY c50 k50 k51
SET c3
1:k50
END.
DO k1=1:k51
RANDOM k50 c1;
INTEGER 1 365.
SORT c1 c1
RANK c1 c2
LET c4=ABS(c2-c3)
LET C5(K1)=SUM(c4)
ENDDO
LET c6=(c5>0)
LET k10=SUM(c6)/k51
NOTE Probability that at least 2 people share the same birthday is k10
PRINT k10
ENDMACRO
```

To run the macro, type %BIRTH at the MINITAB prompt (MTB>). Type in the required N and number of trials. A sample session is shown in Display 2.1. The macro BIRTH.MAC is shown in Display 2.2.

**Analytical Solution to the Birthday Problem**

Let’s consider the case when N=25. The probability of no duplicates in birthdays is given by:

\[
\text{Probability of no birthdays in common} = \frac{365 \times 364 \times \ldots \times 341}{365^{25}} = 0.431
\]

Probability of at least 2 people with the same birthday = 1 - 0.431 = 0.569
One can see that the answer obtained via simulation is quite close to the exact answer. The accuracy of the simulation would of course improve with more trials. If N=23, the probability that 2 people share the same birthday is 0.507 or approximately 50%. If N=50, the probability increases dramatically to 0.970. Isn’t this surprising?

2.2 Alexander’s Dilemma

Alexander’s dilemma is a problem that introduces making decisions under conditions of uncertainty. It also introduces the concepts of expected monetary value, risk taking, and risk aversion.

Problem: I owe my son Alexander $50 per month for doing chores around the house. Instead of giving him the $50, each month I will let him reach into a bag which contains a $100 bill and five $10 bills, and draw two of the bills. Should Alexander go with this scheme, or in other words, is my scheme fair?

If Alexander is a risk taker, then he will have a chance of making $110 instead of $50 per month if he is lucky. On the other hand, if he is risk averse, then he figures that he will more likely be getting $20 rather than $50 per month. So the question is: In the long run, will he be better off taking the $50 or go with my scheme?

Solution via Simulation

There are six bills in the bag each with an equal likelihood of being drawn. For a single trial of drawing two bills, a total of $110 or $20 is possible. The simulation can be carried out much like tossing a 6-sided die as follows: generate randomly two integers
between 1 and 6 (if a 6 is generated, designate that as the $100 bill, integers 1 to 5 represent the $10 bills); keep track of the two integers generated and the resultant total amount; repeat the last two steps many trials (say 1000); calculate the average earnings over the total number of trials.

On MINITAB, simulation is carried out using the macro ALEX.MAC. The macro reads in the number of trials to conduct, puts the 6 bills (five $10 and one $100) in column C1 to be sampled, sample 2 numbers from C1, puts them in C2, adds up the amount and puts it in a row in C3 for each trial. The command DESCRIBE has been used to give the summary statistics of C3. The probabilities of obtaining $20 and $110 are also computed.

To run the macro, type %ALEX at the MINITAB prompt (MTB>), then type in the number of trials desired. A sample session is shown in Display 2.3. The macro ALEX.MAC is shown in Display 2.4.

**Analytical Solution to Alexander’s Dilemma**

Computing the long run average or expected value is easily done by enumerating all the possible combinations of the two bills that could be drawn from the bag. If the two bills are denoted as H for $100 and T1, T2, T3, T4 T5 for each of the $10 bills, the possible combinations of the two bills are:

<table>
<thead>
<tr>
<th>H-T1</th>
<th>T1-T2</th>
<th>T2-T3</th>
<th>T3-T4</th>
<th>T4-T5</th>
</tr>
</thead>
<tbody>
<tr>
<td>H-T2</td>
<td>T1-T3</td>
<td>T2-T4</td>
<td>T3-T5</td>
<td></td>
</tr>
<tr>
<td>H-T3</td>
<td>T1-T4</td>
<td>T2-T5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H-T4</td>
<td>T1-T5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H-T5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Display 2.3: Sample MINITAB session with ALEX macro

<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>MEAN</th>
<th>MEDIAN</th>
<th>TRMEAN</th>
<th>STDEV</th>
<th>SEMEAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>earnings</td>
<td>500</td>
<td>49.34</td>
<td>20.00</td>
<td>47.60</td>
<td>42.23</td>
<td>1.89</td>
</tr>
<tr>
<td>MIN</td>
<td>20.00</td>
<td>20.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAX</td>
<td>110.00</td>
<td>110.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q1</td>
<td>20.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q3</td>
<td>110.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Probability of earning $20 is k2
k2 = 0.674

Probability of earning $110 is k3
k3 = 0.326

A third of these combinations yields a value of $110, and two-thirds yields a value of $20. Thus the expected value of the two bills is:

$$\frac{1}{3} \times 110 + \frac{2}{3} \times 20 = 50$$

So, in the long run, Alexander will neither gain nor lose by my scheme. In other words, the offer is fair.

Display 2.4: ALEX macro

```
ALEX.MAC
GMACRO
ALEX
ERASE c1-c50
NAME c3 ‘earnings’
SET c1
10 10 10 10 10 100
END
NOTE Number of trials? (e.g. 100)
SET c50;
file “terminal”;
NOBS 1.
COPY c50 k50
DO k1=1:k50
SAMPLE 2 c1 c2
LET c3(k1)=SUM(C2)
ENDDO
DESCRIBE c3
LET c4=(C3=20)
```
It can be seen that the answer obtained by simulation is quite good. The macro can of course be run several times to get a sense of variability of the results from experiment to experiment.

2.3 De Mere’s Die Tossing Problem

The De Mere’s die tossing problem was briefly alluded to in the Introduction. This problem was supposed to have started the study of the theory of probability. History has it that the Chavelier de Mere, a French gambler with some interest in mathematics, consulted his friend the eminent mathematician Blaise Pascal concerning the problem posed below.

**Problem:** Which is likelier: Rolling at least one six in 4 tosses of a single die, or rolling at least a double-six in 24 tosses of 2 dice?

De Mere somehow has reasoned that the average number of successful tosses was the same for both gambles. The chance of a six is 1/6, so the average number in four tosses should be 4 x 1/6 or 2/3. For the two dice, the chance of a double-six is 1/36, so in 24 tosses, the average number should be 24 x 1/36 or 2/3. He was puzzled when he lost
more often with the second gamble. De Mere’s reasoning was flawed of course, as we will see later.

**Solution via Simulation**

First consider the case of a single die. Randomly generate integers between 1 and 6 four times. If at least one six is observed in that four times, then count it as a “yes” or “1”, if no six is observed, then count it as “no” or “0”. For a fairly reliable answer, the procedure must be repeated N times where N is at least a hundred or a thousand. The probability of obtaining at least a six in four tosses of a die is then the number of “yes’s” or “1’s” divided by N.

On MINITAB the macro (DIE.MAC) will perform the die tossing simulation. This macro is an interactive version of the DICE macro given in the Introduction. First five 0’s and one 1 is put in column C1. The required number of trials to perform is then entered. The macro then randomly samples from column C1 four times *with replacement* and puts the values in column C2. This step is in fact the same as tossing the die four times separately. The values in C2 are then summed. If a non-zero sum is obtained, then a six must be present. A zero sum would mean that there is no six in any of the four tosses. The number of non-zero sums divided by the total number of trials is the required probability.

To run the macros, type %DIE. Four sample runs of 100 trials each using the macro are shown in Display 2.5. The macro DIE.MAC is shown in Display 2.6. The average probability from the four separate runs shown in Display 2.5 is 0.515.
Display 2.5: Sample MINITAB session with DIE macro

MTB> %DIE

Number of trials? (e.g. 100)
DATA> 100
The probability of at least 1 six in four tosses is k2:
k2 0.510
MTB> %DIE

Number of trials? (e.g. 100)
DATA> 100
The probability of at least 1 six in four tosses is k2:
k2 0.540
MTB> %DIE

Number of trials? (e.g. 100)
DATA> 100
The probability of at least 1 six in four tosses is k2:
k2 0.490
MTB> %DIE

Number of trials? (e.g. 100)
DATA> 100
The probability of at least 1 six in four tosses is k2:
k2 0.520

MTB>

Display 2.6: DIE macro

DIE.MAC

GMACRO
DIE
ERASE c1-c50
SET c1
5(0) 1
END

NOTE Number of trials? e.g. 1000
SET c50;
FILE “terminal”;
NOBS 1.
COPY c50 k50
DO k1=1:k50
SAMPLE 4 c1 c2:
REPLACE.
LET c3(k1)=SUM (c2)
ENDDO
LET c4=(C3>=1)
LET k2=SUM(c4)/k50
NOTE: The probability of obtaining at least a six in four tosses of a die is k2
PRINT k2
ENDMACRO

For the case of two dice, the simulation of MINITAB is carried by the macro TWODICE.MAC. The macro is quite similar to the single die macro except that sampling from column C1 is done 24 times with replacement twice, and putting the results in columns C2 and C3. If the same row of C2 and C3 contains a “1”, then a double six has occurred. This will be indicated by a “1” in column C4, otherwise it will be a “0”. A non-zero sum in C4 would indicate that at least a double-six has occurred. The number of non-zero sums divided by the total number of trials would be required probability. To run the macro, type %TWODICE. A sample run of 1000 trials using the macro is shown in Display 2.7. The macro TWODICE is shown in Display 2.8.

**Display 2.7: Sample MINITAB session with TWODICE macro**

```
MTB> %TWODICE
Number of trials? (e.g. 100)
DATA> 1000
Probability that at least a double-six is k2:
k2  0.489000

MTB>
```
Display 2.8: TWODICE macro

```
TWODICE.MAC

GMACRO
TWODICE
ERASE c1-c50
SET c1
5(0) 1
END
NOTE Number of trials? (e.g. 100)
SET c50;
FILE “terminal”;
NOBS 1.
COPY c50 k50
DO k1=1:k50
SAMPLE 24 c1 c2;
REPLACE.
SAMPLE 24 c1 c3;
REPLACE
LET c4=(c2=1 AND c3=1)
LET c5(k1)=SUM(c4)
ENDDO
LET c6=(c5>=1)
LET k2=SUM(c6)/k50
NOTE Probability of at least a double-six is k2:
PRINT k2
ENDMACRO
```

**Analytical Solution to De Mere’s Dice Tossing Problem**

Consider the case of a single die. The probability of getting a six for a fair die is 1/6 per toss. The probability of not getting a six is thus 5/6. Hence, for four tosses:

\[
\text{Probability of not getting a six in four tosses} = \left( \frac{5}{6} \right)^4 = 0.482
\]

The probability of at least a six in four tosses is then 1-0.482=0.518.

Similarly, for two dice, the probability of getting a double-six is 1/36 per toss. The probability of not getting a double-six is thus 35/36. Hence, for 24 tosses:
Probability of not getting a double six in 24 tosses \[= \left( \frac{35}{36} \right)^{24} \approx 0.509 \]

The probability of at least a double-six is 1-0.509=0.491. Hence it can be seen that the second gamble less likely than the first. This assumes that dice are fair and that the tosses are independent. That is, the result of one toss does not affect that of the next or other tosses.

2.4 Exercises

Each problem given below can be solved quite easily using simulation on MINITAB.

2.1 This question was given to the columnist Marilyn vos Savant of Parade Magazine.

Suppose that you are on a television show and they show you three doors. Behind one door is a car, and behind each of the two others a goat. You are asked to choose one of the doors. You pick a door, say no. 1, which, however, is not opened. The host, who knows what is behind all three doors, opens one of the other two doors, say no. 3, and out comes a goat. (The host never opens the door which hides the car). He then says to you: You are allowed to switch from door no. 1 to door no. 2 if you find that advantageous. Should you switch or not?

2.2 This problem is from Mosteller’s classic book of challenging probability problems.

When 100 coins are tossed, what is the probability that exactly 50 are heads?

2.3 When we say: 4 out of 5, or 8 out of 10, do these numbers convey exactly the same meaning? That is, do they all mean 80% or is 80 out of 100 more convincing statistically than 4 out of 5?
3. MORE PROBABILITY PROBLEMS

This chapter considers several more classic probability problems. These problems often have very useful applications in science and engineering. Each of these problems will be solved using simulation and where exact analytical solutions are available, they will be given for comparison with those obtained using simulation.

3.1 Catching the Counterfeiter and Risk Analysis

This classic catching the counterfeiter (more accurately the cautious counterfeiter) problem, depending on how the question is posed, is closely linked to the de Mere’s dice tossing problem and to what is known as risk analysis. This is another one of those counter-intuitive probability problems with surprising results. The question will be first posed as the classic counterfeiter problem. Risk analysis will be considered later.

**Problem:** You are the ruler of the Kingdom of Belle Isle and you suspect that your minter is robbing you by substituting counterfeit gold coins for real ones. The coins are packed 50 to a bag, and in fact the minter is placing one counterfeit gold coin in every bag. You command the minter to bring in 50 bags of coins, and from each bag you select a coin for analysis. What is the probability that you will find a counterfeit coin? If both 50’s are replaced by n, what is the probability now?

Most people’s reaction to the problem is likely to be that the minter is very safe. After all, there is only one counterfeit coin in a bag of 50, so the chance of being detected must be 1/50. With more coins per bag, the minter should be even safer. Is this true?
**Solution via Simulation**

The simulation procedure is similar to the die tossing problem. Instead of a 6-sided die, and tossing it four times, here we need a 50-sided die and tossing it 50 times. If we designate one of the 50 integers (say 13) as the fake coin, then when the “13” is observed one or more times in 50 tosses, then count it as a “yes”, if no “13” is observed, then count it as a “no”. The procedure is then repeated N times. The probability of obtaining at least a “13” or fake coin in 50 bags is then the number of “yes’s” divided by N. Alternatively use 49 “0” for good coins and a single “1” to represent the fake coin. If the “1” is observed one or more times in among the 50 bags, then count it as a “yes”, if no “1” is observed, then count it as a “no”. Using the second method is much easier to implement on MINITAB.

On MINITAB the macro COUNTF.MAC will perform the catching the counterfeiter simulation. First, forty nine 0’s and one 1 is put in column C1. The required number of trials to perform is then entered. The macro then randomly samples from column C1 50 times with replacement and puts the values in column C2. The values in C2 are then summed. If a non-zero is obtained, then a “1” must be present. A zero sum would mean that there is no “1” in any of the 50 bags. The number of non-zero sums divided by the total number of trials is the required probability.

Sample results from running the macro with 100 trials each are given in Display 3.1. The first two runs were with different number of coins per bag with number of bags equal to the number of coins. A third run was also made with the number of bags twice the number of coins. The macro COUNTF.MAC is shown in Display 3.2.
Risk Analysis

Consider the risk of being involved in an automobile accident (fender bender, minor, life threatening, or fatal) over a person’s lifetime. On any given trip the chance is about 1 in 1,000 by some estimate. However, the chance of an accident increases significantly when we think not in terms of a single trip but in terms of automobile trips taken in a month, which may be 500 trips depending on a person’s occupation, etc. So when we talk about risk of an accident, we consider not only the probability per trip, we also consider the number of trips. Another example is in the design of a dike for a flood protection scheme. The dike may be designed for a flood level with an average return period of 100 years (i.e. a flood level that will be equalled or exceeded with a probability of 1% in any year). This design level may seem very safe. However, if we consider the risk that the dike will be overtopped during its lifetime, which may be 100 years, then, the dike may not be that safe.

**Display 3.1: Sample MINITAB session with COUNTF macro**

```
MTB> %COUNTF
No. of trials, no. of coins to a bag, and no. of bags? (e.g. 100 50 50)
DATA>  100 50 50
The probability of catching counterfeiter is k2:
k2       0.65000
MTB>  %COUNTF
No. of trials, no. of coins to a bag, and no. of bags? (e.g. 100 50 50)
DATA>  100 100 100
The probability of catching counterfeiter is k2:
k2       0.64000
MTB>  %COUNTF
No. of trials, no. of coins to a bag, and no. of bags? (e.g. 100 50 50)
DATA>  100 50 100
The probability of catching counterfeiter is k2:
k2       0.85000
MTB>
```
The calculation of the risk thus takes into account the number of encounters (number of trips, design life of dike, etc.), as well as the probability of occurrence per encounter (probability of accident per trip, probability of dike overtopping per year, etc). These types of problems as one can see are identical to the “catching the counterfeiter” problem and dice tossing problem. Probability of accident per trip is equivalent to probability of detecting a fake coin per bag, and number of trips in a month is equivalent to total number of bags examined.

**Display 3.2: COUNTF macros**

```plaintext
COUNTF.MAC

GMACRO
COUNTF
ERASE c1-c50
NOTE No. of trials, no. of coins to a bag, and no. of bags? (e.g. 100 50 50)
SET c50;
FILE “terminal”;
NOBS 3.
COPY c50 k50 k51 k5
LET k3=k51-1
SET c1
k3(0) 1
END
DO k1=1:k50
SAMPLE k52 c1 c2;
REPLACE.
LET c3(k1)=SUM(c2)
ENDDO
LET c4=(c3>=1)
LET k2=SUM(c4)/k50
NOTE The probability of catching counterfeiter is k2
PRINT k2
NOTE
ENDMACRO
```

The macro RISK.MAC for risk calculation is given in Display 3.4. A small change was made to the COUNTF.MAC macro for the risk calculations. Instead of
sampling from a column with zeros and ones, the zeros and ones are generated using the RANDOM and DISCRETE commands on MINITAB. This command generates discrete random variables from column C1 according their probability in column C2. This way require less space for storing intermediate results and also the probability per encounter can be entered as an input value directly. On executing the macro, the number of trials, the probability per encounter (e.g. 0.01 for the dike problem), and the number of encounters (e.g. 100 for the dike problem) will be requested by the macro.

Two sets of results from running the RISK macros with trials of 100 are shown in Display 3.3. The first set of results considers the risk of an automobile accident in a month, and the second set of results considers risk of dike failure over its design life.

**Display 3.3: Sample MINITAB session with RISK macro**

```
MTB> %RISK
No. of trials, probability per encounter, no. of encounters? (e.g. 100 .2 25)
DATA> 100 0.001 500
The probability of at least one encounter (risk) is k2:
K2   0.4000
MTB> %RISK
No. of trials, probability per encounter, no. of encounters? (e.g. 100 .2 25)
DATA> 100 0.01 100
The probability of at least one encounter (risk) is k2:
K2   0.6400
MTB>
```
Display 3.4: RISK macro

RISK.MAC

GMACRO
RISK
ERASE c1-c50
NOTE No. of trials, probability per encounter, no. of encounters? (e.g. 100 .2 25)
SET c50;
FILE “terminal”;
NOBS 3.
COPY c50 k50 k51 k52
LET k53=1-k51
SET c1
0 1
END
SET c2
k53 k51
END
DO k1=1:k50
RANDOM k52 c3;
DISCRETE c1 c2
LET c4(k1)=SUM(c3)
ENDDO
LET c5=(c4>=1)
LET k2=SUM(c5)/k50
NOTE The probability of at least one encounter (risk) is k2
PRINT k2
ENDMACRO

Analytical Solution to Catching the Counterfeiter and Risk Problem

The analytical solution is very similar to the die tossing problem. The differences are in the context and the numbers involved. Consider the case where there are 50 coins per bag and 50 bags were examined. The probability of finding a fake gold coin in each bag is 1/50. The probability of not finding a fake gold coin is thus 49/50. Hence, for 50 bags:

\[
\text{Probability of not finding a fake gold coin in 50 bags} = \left( \frac{49}{50} \right)^{50} = 0.364
\]

The probability of finding at least fake gold coin is then 1-0.364=0.636.
For 100 coins per bag and 100 bags, similar calculations give the probability of finding at least one fake gold coin is 0.634. This of course is the same analytical solution as the risk of dike overtopping where probability of overtopping is 1/100 per year and design life is 100 years. In fact it can be shown that in general if we have \( n \) gold coins per bag and \( n \) bags:

\[
\text{Probability of finding at least 1 fake gold coin in n bags} = 1 - \left(1 - \frac{1}{n}\right)^n \rightarrow 1 - \frac{1}{e} = 0.63212.
\]

as \( n \) tends to infinity!

For the case where we have \( N \) encounters and the probability per encounter is \( p \):

\[
\text{Probability of at least 1 occurrence} = 1 - (1 - p)^N
\]

For the automobile accident example, \( p = 0.001 \) and \( N = 500 \), giving a probability of at least one accident in a month to be 0.3936.

### 3.2 Catching a Greedy Counterfeiter

The catching a greedy counterfeiter problem is a generalization of the previous problem discussed in section 3.1. In fact the previous problem is a special case of the problem to be considered here.

**Problem:** You are again the ruler of the Kingdom of Belle Isle and you suspect that your minter is robbing you by substituting counterfeit gold coins for real ones. The coins are packed \( n \) to a bag. This time the minter is placing \( m \) counterfeit gold coins in each bag. You command the minter to bring in \( b \) bags of coins, and from each bag you select a coin for testing. What is the probability that the sample of \( b \) bags of coins contains exactly \( r \) counterfeit coins?
In this problem, instead of 1 counterfeit coin per bag, it is now \( m \) counterfeit coins per bag. Hence the term “greedy” is used. Also, we are interested in the probability of finding exactly \( r \) counterfeit coins in the sample of size \( b \). In the previous problem we were interested in the probability of finding at least one counterfeit coin in a sample of size \( n \).

**Solution via Simulation**

The simulation procedure is again very similar to the catching the cautious counterfeiter problem. The only difference is that we need to keep track of the number of counterfeit coins found per trial. At the end of the experiment, we can then obtain the distribution of the number of counterfeit coins, and the probability of finding \( r = 0, 1, 2, 3, \ldots \) counterfeit coins in the sample of \( b \) bags.

On MINITAB the macro (GREEDY.MAC) will perform the greedy counterfeiter simulation. The number of trials, number of coins (\( n \)) per bag, number of counterfeit coins (\( m \)) per bag, and the number of bags (\( b \)) to be examined are first entered. The number of bags (\( b \)) may be the same size as \( n \) or some other number. Then \( n-m \) 0’s and \( m \) 1’s are put in column C1. The macro then randomly samples from column C1 \( b \) times with replacement and puts the values in column C2. The values in C2 is then summed and stored in C3. A histogram of values in C3 will show the distribution of the number of counterfeit coins. The probability of finding \( r = 0, 1, 2, \ldots \) counterfeit coins can be obtained by using results from the histogram. Sample results from running the macro GREEDY with 1000 trials are given in Display 3.5. The run used \( n=25 \), \( m=5m \), and \( b=25 \). The macro GREEDY.MAC is shown in Display 3.6.
Display 3.5: Sample MINITAB session with GREEDY macro

MTB>%GREEDY
No. of trials, no of coins to a bag, no. of fakes, and no. of bags? (e.g. 100 50 5 50)
DATA> 1000 25 5 25

Distribution of the number of counter coins

Histogram of fakes   N=1000
Each * represents 5 obs.
Midpoint  Count
0.00           1  *
1.00          26  *****
2.00          68  ********
3.00        151  ************
4.00        183  ****************
5.00        191  *****************
6.00        167  ****************
7.00        105  *********
8.00           64  ****
9.00           27  ***
10.00          13  **
11.00           3  *
12.0            1  *

The mid-point values are the number of counterfeit coins r.
Probability of exactly r counterfeits = Count/no. of trials.

MTB> # e.g. Probability of exactly 4 counterfeit coins = 183/1000 = 0.183.

MTB>

Analytical Solution to the Greedy Counterfeiter Problem

The solution to the greedy counterfeiter problem makes use of the binomial distribution if it is assumed that the drawings of the coins are independent. If there are m counterfeits in a bag of n coins, the probability of drawing a counterfeit coin is m/n. From the binomial distribution, with b bags:

\[
\text{Probability of } r \text{ counterfeit coins} = \binom{b}{r} \left( \frac{m}{n} \right)^r \left( 1 - \frac{m}{n} \right)^{b-r}
\]
For example, for n=25, m=5, b=25, and r=4,

\[
\text{Probability of 4 counterfeit coins} = \binom{25}{4} \left( \frac{5}{25} \right)^4 \left( 1 - \frac{5}{25} \right)^{21} = 0.187
\]

If m=1, and we are interested in the probability of finding at least one counterfeit coin, then this becomes the same problem at the cautious counterfeiter problem. Therefore, the greedy counterfeiter problem is the more general case.

**Display 3.6: GREEDY macros**

```
GREEDY.MAC

GMACRO
GREEDY
ERASE c1-c50
NOTE No. of trials, no. of coins to a bag, no. of fakes, and no. of bags? (e.g. 100 50 5 50)
SET c50;
FILE “terminal”;
NOBS 4.
COPY c50 k50 k51 k52 k53
LET k54=k51-k52
SET c1
k54(0) k52(1)
END
DO k1=k1:k50
SAMPLE k53 c1 c2;
REPLACE.
LET c3(k1)=SUM(c2)
ENDDO
NOTE Distribution of the number of counterfeit coins
NAME c3 ‘FAKES’
GSTD
HISTOGRAM c3
NOTE
NOTE The mid-point values are the number of counterfeit coins r.
NOTE Probability of exactly r counterfeits = Count/no. of trials.
ENDMACRO
```
3.3 Basketball Problem

The basketball problem is another special case of the previous problem. It is given here because of the different context, and it may be of interest to the millions of sport fans who may identify with this problem better than the greedy counterfeiter problem! In addition it provides an introduction to p-values and $\alpha$-values.

**Problem:** Your favorite basketball star Larry H. Moe who averages 47% success in shooting, has missed 7 out of his last 10 shots. Is he really in a slump and should be replaced from the line up, or is the streak of misses just a chance occurrence.

To answer the above question, we need to find the probability that he will miss 7 or more out of 10 shots assuming that his probability of making successful shots remains at 47%. As one can see, this problem is in fact similar to the greedy counterfeiter problem with the $b = 10$, $m/n = 1-0.47 = 0.53$, and $R = 7$. Here $m/n$ is the probability of missing each shot which is one minus the probability of success (shooting percentage).

**Solution via Simulation**

The simulation procedure is practically the same as the greedy counterfeiter problem. Instead of keeping track of the number of counterfeit coins, we keep track of the number of missed shot per trial. At the end of the experiment, we can then obtain the distribution of the number of missing $r = 0, 1, 2, \ldots$ shots in $b$ attempts at the basket. The probability of missing 7 or more shots in 10 attempts can then be computed.

On MINITAB the macro (BASKET.MAC) will perform the basketball simulation. The macros are basically similar to the GREEDY.MAC macro. A small
change was made to use the RANDOM and DISCRETE commands for sampling instead of setting 0’s and 1’s in column C1. This approach was also used in the RISK.MAC macro earlier. After the BASKET.MAC macro is run, the number of trials, shooting percentage, and number of attempts at basket are requested. A histogram of values in C4 will show the distribution of the number of misses. The probability of missing \( r = 0, 1, 2, \ldots \) shots is done outside the macro by using results from the histogram. Sample results from running the macro (BASKET.MAC) with 1000 trials are given in Display 3.7. The run used a shooting percentage of 47% and 10 attempts at basket. The macro BASKET.MAC is shown in Display 3.8.

**Display 3.7: Sample MINITAB session with BASKET macro**

<table>
<thead>
<tr>
<th>Midpoint</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>1</td>
</tr>
<tr>
<td>1.00</td>
<td>4</td>
</tr>
<tr>
<td>2.00</td>
<td>30</td>
</tr>
<tr>
<td>3.00</td>
<td>86</td>
</tr>
<tr>
<td>4.00</td>
<td>170</td>
</tr>
<tr>
<td>5.00</td>
<td>248</td>
</tr>
<tr>
<td>6.00</td>
<td>235</td>
</tr>
<tr>
<td>7.00</td>
<td>141</td>
</tr>
<tr>
<td>8.00</td>
<td>69</td>
</tr>
<tr>
<td>9.00</td>
<td>15</td>
</tr>
<tr>
<td>10.00</td>
<td>1</td>
</tr>
</tbody>
</table>

The mid-point values are the number of missed shots \( r \). Probability of exactly \( r \) misses = Count/no. of trials.

MTB> # e.g. Probability of exactly 7 misses = 141/1000 = 0.141.

MTB> # e.g. Probability of missing 7 or more shots = \((141+69+15+1)/1000\) = 0.226
Analytical Solution to the Basketball Problem

The solution to the basketball problem also uses the binomial distribution. If the probability of making a successful basket is 0.47, then the probability of a miss is 0.53.

For 10 attempts at basket, the probability of missing 7 shots is:

\[
\text{Probability of 7 misses} = \binom{10}{7}(0.53)^7(0.47)^3 = 0.1463
\]

**Display 3.8: BASKET macro**

BASKET.MAC

```
GMACRO
BASKET
ERASE c1-c50
NOTE N. of trials, shooting percentage, no . of attempts? (e.g. 100 45 10)
SET c50;
FILE “terminal”
NOBS 3.
COPY c50 k50 k51 k52
LET k53=k51/100
LET k54=1-k53
SET c1
0 1
END
SET c2
k53 k54
END
DO k1=1:k50
RANDOM k52 c3;
DISCRETE c1 c2.
LET c4(k1)=SUM (c3)
ENDDO
NOTE Distribution of the number of misses:
NAME c4 ‘misses’
GSTD
HISTOGRAM C4;
INCREMENT 1.
NOTE The midpoint values are the number of missed shots r.
NOTE Probability of exactly r misses = Count/no. of trials
ENDMACRO
```
To calculate the probability of 7 or more misses, we need to add together the probability of 7, 8, 9, and 10 misses, i.e.

\[
\text{Probability of 7 or more misses} = \binom{10}{7}(0.53)^7(0.47)^3 + \binom{10}{8}(0.53)^8(0.47)^2 + \binom{10}{9}(0.53)^9(0.47)^1 + (0.53)^{10} = 0.2255
\]

From the above, we can see that simulation with 1000 trials gave fairly accurate results. Also, the probability of missing 7 shots in 10 attempts for a basketball star with shooting percentage of 47% is about 0.146. If we consider 7 or more misses, the probability is about 0.23 which is about \(\frac{1}{4}\) of the time. Therefore, for the star player to sometimes miss 7 or more of 10 shots is due to chance. There is no need to replace him! However if we find that the probability of missing 7 or more shots is small, say smaller than 0.05, then there may be cause for worry. In other words, the probability is small enough that it may not be due to chance. The probability calculated is called the p-value, and the benchmark probability value e.g. 0.05 is called the \(\alpha\) -value. If the p-value <\(\alpha\) -value, then the results are statistically significant. For the problem at hand, the p-value is 0.226 (for 7 or more misses) which is greater than 0.05, so the results is not statistically significant at \(\alpha = 0.05\). Computing p-values, etc. are in the realm of statistical hypothesis testing.

### 3.4 Case of the Defective Concrete

The case of the defective concrete is an example in the area of acceptance sampling which is in the realm of quality control. The problem while closely related to the
binomial case differs from the two previous problems because of the way in which samples are selected.

**Problem:** A contractor has received a shipment of 20 concrete cylinders, five for project A and the others for project B. Suppose that 6 of the 20 are defective. If the five required for project A are randomly selected from the 20, what is the probability that among the five, r of the cylinders are defective?

If we think in terms of the greedy counterfeiter’s problem, this problem is like having only one bag (the shipment) of 20 coins (concrete cylinders) in which 6 are counterfeits (defective concrete). Five are randomly selected from the 20 and we want to know the probability that r of the five are counterfeit coins. So the main differences between this problem and the greedy counterfeiter’s problem are: only 1 bag instead of 20 bags, 5 are randomly selected from the 1 bag instead of 1 from each of the 20 bags, and we are interested in the probability that r of the 5 selected are fakes instead of r of the 20 (1 from each bag) are fakes.

**Solution via Simulation**

There are 20 concrete cylinders in which 6 are defectives. In simulation, we randomly generate 5 integers between 1 and 20. If we have designated the numbers 1 to 6 to represent defective concrete, then if any number from 1 to 6 appearing in the sample of 5 will be counted as defective concrete. This process is repeated over many times (trials), and the probability of having 4 defectives would then be the number of times 4 defectives
was observed in each trial divided by the total number of trials. The most important thing to note is that each the five integers generated per trial must be unique, i.e. there is no duplication. In other words, this time we have to sample without replacement.

Macro CONCRETE.MAC shown in Display 3.10 will perform the simulation. On MINITAB it is easier to use 0’s and 1’s to represent non-defective and defective concrete, respectively. Hence if we have 20 cylinders and 6 are defective, we put fourteen 0’s and six 1’s in column C1 and sample 5 times without replacement from C1. On MINITAB, the default for the SAMPLE command is sampling without replacement, hence no sub-command is required. Keeping track of the number of defectives is similar to those of previous macros. After the CONCRETE macro is executed, the number of trials, number of cylinders, number of defectives, and the number of samples are requested. A histogram of values in C3 will show the distribution of the number of defectives. The probability of $r = 0, 1, 2, \ldots$ defectives is done outside the macros by using results from the histogram.

Sample results from running the macro (CONCRETE.MAC) with 1000 trials are given in Display 3.9. The run used 20 cylinders in which 6 are defective, and five of the 20 are selected. The macro CONCRETE.MAC is shown in Display 3.10.

**Analytical Solution to the Case of the Defective Concrete**

The solution to the defective concrete problem uses the hypergeometric distribution. If $R$ is the number of $S$’s (successes) in a complete random sample of size $n$ drawn from a population consisting of $M$ $S$’s and $(N-M)$ $F$’s (failures), then the probability distribution of $R$ is given by:
Probability that \((R = r) = \binom{M}{r} \binom{n-M}{n-r} \binom{N}{n}\)

For the problem at hand, \(N = 20, M = 6, n = 5\), and if we are interested in the probability of exactly 2 defectives in the sample of 5, then:

\[
\text{Probability that } (R = 2) = \binom{6}{2} \binom{20-6}{5-2} \binom{20}{5} = 0.352
\]

and the probability of 2 or more defectiveness is

\[
\text{PR}(R=2)+\text{Pr}(R=3)+\text{Pr}(R=4)+\text{Pr}(R=5)=0.483
\]

Both answers are fairly close to the simulation results using the macro.

**Display 3.9: Sample MINITAB session with CONCRETE macro**

```
MTB> % CONCRETE
No. of trials, no. of cylinders, no. of defectives, no. selected? (e.g. 100 20 6 5)
DATA>  1000 20 6 5

Distribution of the number of defectives:

Histogram of Defects     N = 1000
Each * represents 10 obs.

<table>
<thead>
<tr>
<th>Midpoint</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>130</td>
</tr>
<tr>
<td>1.00</td>
<td>380</td>
</tr>
<tr>
<td>2.00</td>
<td>361</td>
</tr>
<tr>
<td>3.00</td>
<td>115</td>
</tr>
<tr>
<td>4.00</td>
<td>13</td>
</tr>
<tr>
<td>5.00</td>
<td>1</td>
</tr>
</tbody>
</table>

The mid-point values are the number of defectives \(r\).
Probability of exactly \(r\) defectives = Count/no. of trials.

MTB> # e.g. Probability of exactly 2 defectives = 361/1000 = 0.361
MTB> # e.g. Probability of 2 or more defectives = (361+115+13+1)/1000 = 0.490
```
### Display 3.10: CONCRETE macro

```plaintext
CONCRETE.MAC

GMACRO
CONCRETE
ERASE c1-c50
NOTE Number of trials, no. of cylinders, no. of defectives, no selected?
NOTE (e.g. 100 20 6 5)
SET c50;
FILE “terminal”;
NOBS 4.
COPY c50 k50 k51 k52 k53
LET k54=k51-k52
SET c1
k54(0) k52(1)
END
do k1=k1:k50
SAMPLE k13 c1 c2
LET c3(k1)=SUM(c2)
ENDDO
NOTE Distribution of the number of defective cylinders:
NAME c3 ‘Defects”
GSTD
HISTOGRAM c3;
INCREMENT 1.
NOTE The mid-point values are the number of defectives 4.
NOTE Probability of exactly 4 defectives = Count/no. of trials.
ENDMACRO
```

### 3.5 Winning the Lottery

The probability of winning the lottery is another classic problem. Two types of lotteries will be discussed. The first type of lottery is commonly called Lotto 649 in Canada, and the second type is the more traditional lottery ticket. In Lotto 649, bettors are asked to choose 6 unique numbers from 1-49. If all 6 numbers are correctly chosen (order is not important), then the bettor stand to win millions of dollars. In the traditional lottery, tickets are serially numbered from 0000 to 9999 in four-digit lottery tickets (prizes are
usually small), or from 00000000 to 99999999 in eight-digit lottery tickets (prizes are usually substantial). The bettor usually does not have much of a choice in the numbers.

**Lotto**

To illustrate the use of MINITAB to compute the probability of winning in Lotto 649, a simpler version Lotto 410 will be used. That is, instead of choosing 6 numbers from the numbers 1 to 49, here we will choose 4 numbers from the numbers 1 to 10. To compute the probability of winning Lotto 649 using simulation is of course possible except that it would require at least 100 million trials for a fairly accurate answer!

**Problem:** What is the probability of winning at Lotto 410?

The four unique numbers chosen by the bettor must exactly match the four numbers randomly chosen by the Lotto organizers. For example, if you bet on the numbers 1, 7, 3, and 4, then if the four numbers randomly drawn by the organizers are also 1, 7, 3, and 4, in any order then you win! Sometimes smaller prizes are also given for partial matches. The current problem is in fact a special case of the defective concrete problem; it is the same as asking in the previous problem: what is the probability that all five concrete cylinders selected are defective?

**Solution via Simulation**

The simulation procedure is actually quite similar to the case of the defective concrete problem. First designate any four the numbers say 1, 2, 3, and 4 as the winning numbers. Then randomly generate 4 integers between 1 and 10. If the four numbers generated
matches the four numbers we have designated as winning numbers, then count it as a “yes” or “1”, if not then a “no” or “0”. Repeat this step a large number of times. Count the number of times there was a match and divide this number by the number of trials to give the required probability.

On MINITAB, the macros LOTTO.MAC will perform the required simulation. On MINITAB it is easier to use four 1’s to represent the winning numbers and six 0’s to represent the other six numbers. These 10 numbers (four 1’s and six 0’s) are put in column C1. Then we randomly sample 4 numbers from C1 without replacement and put them in C2. If the numbers in column C2 summed to 4, then all the 1’s must have been sampled and this indicates a “hit”, otherwise it is a “miss”. The “hits” and “misses” are represented by 1’s and 0’s and these are stored in column C3. The number of such “hits” divided by the number of trials is the required probability.

Display 3.11 shows a sample run with the LOTTO macro. Here 1000 trials have been used. The LOTTO macro is given in Display 3.12.

**Display 3.11: Sample MINITAB session with LOTTO macro**

<table>
<thead>
<tr>
<th>MTB&gt; %LOTTO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of trials? (e.g. 1000)</td>
</tr>
<tr>
<td>Probability of winning Lotto 410 is k2:</td>
</tr>
<tr>
<td>MTB&gt;</td>
</tr>
</tbody>
</table>
Traditional Lottery

In the traditional lottery ticket, the order of the digits is important. For example, in a four
digit lottery ticket, if the bettor buys a ticket with the number 2678, then only this number
will win. That is, other rearrangements of the four digits say 6287, or 7826, etc. would
not be eligible to win. To illustrate the use of MINITAB to compute the probability of
winning a traditional m-digit lottery, the simpler 2-digit case will be considered. More
digits would require more computer time and worksheet space on MINITAB.
**Problem:** What is the probability of winning a 2-digit (00-99) lottery?

There are 2-digit numbers between 00 and 99 which is the winning number. There are 100 numbers in total, hence the probability of winning is 1/100. For a 4-digit lottery, the numbers are from 0000 to 9999, which in total has 10,000 numbers. Hence probability of winning is a 4-digit lottery is 1/10000.

**Solution via Simulation**

If one is convinced of the above argument, the probability of winning the traditional lottery can be easily verified using simulation. Designate any one of the numbers between 00 to 99 as the winning number. Then randomly generate a large number (N) of integers between 0 and 99. Count the number of times that the randomly generated integers match the designated number. The number of matches divided by N is the required probability.

On MINITAB, the macro LOTTERY.MAC will perform the simulation and computation. When the macro is executed, the number of trials N, and the designated winning 2-digit number will be requested. Then N rows of integers between 0 and 99 will be randomly generated using the RANDOM command and INTEGER subcommand, and put in column C1. The designated winning number is then subtracted from C1. If the result from the subtraction is 0, then there is a match. The number of matches is kept track of in C2. The number of matches divided by N is the required probability.
Display 3.13: Sample MINITAB session with LOTTERY macro

MTB>%LOTTERY
Number of trials and Winning 2-digit number (00 – 99)? (e.g. 1000 15)
DATA> 1000 15

Probability of winning at 2-D lottery is k2:

K2  0.0120000
MTB> %LOTTERY
DATA> 1000 09
Probability of winning at 2-D lottery is k12:
k12  0.0090000
MTB> LET k5 = (0.012+0.009)/2
MTB>
MTB>PRINT k5
k5  0.0105
MTB>

Display 3.14: LOTTERY macro

LOTTERY.MAC

GMACRO
LOTTERY
ERASE c1-c50
NOTE Number of trials and Winning 2-digit number (00-99)? (e.g. 1000 15)
SET c50;
FILE “terminal”
NOBS 2.
COPY c50 k10 k51
RANDOM k50 c1;
INTEGER 0 99.
LET c2=(c1-k51=0)
LET k2=SUM(c2)/k50
NOTE Probability of winning at 2-D lottery is k2:
PRINT k2
ENDMACRO

The macro can be run several times and the results averaged. Display 3.13 shows 2 sample runs with the LOTTERY macro. Here 1000 trials per run have been used and the results averaged. The LOTTERY macro is given in Display 3.14.
Analytical Solution to Winning the Lottery

The probability of winning at Lotto 410 or Lotto 649 can be calculated using the hypergeometric distribution as discussed in the analytical solution to the case of the defective concrete problem in Section 3.4. Cast in terms of the defective concrete problem, the Lotto 410 problem is like saying 4 of 10 concrete cylinders are defective, and we need to randomly select 4 from the 10, and we are interested in the probability that all 4 are defective. That is,

\[
\text{Probability that } (R = 4) = \frac{\binom{4}{4} \binom{10-4}{4-4}}{\binom{10}{4}} = \frac{1}{\binom{10}{4}} = 0.00476
\]

The probability obtained using the LOTTO macro is 0.005 which is reasonably close to the exact answer. For the Lotto 649 case,

\[
\text{Probability that } (R = 6) = \frac{1}{\binom{49}{6}} = \frac{1}{13,983,816} = 0.7151 \times 10^{-7}
\]

The above answer shows that the probability is actually equal to 1 divided by the number of ways of choosing 6 numbers from 49.

For the traditional lottery, the probability of winning is straightforward. If there is a m-digit lottery, then probability of winning is \(1/(10^m)\).

3.6 Meeting for Lunch

The meeting for lunch problem is another classic probability problem. This problem is different from the others because it deals with a continuous variable and it requires the use of geometry for an analytical solution. The meeting for lunch problem is the same kind of problem as “The Hurried Duelers” problem given in Mosteller’s book.
**Problem:** Two friends who have unpredictable lunch hours agree to meet for lunch at their favourite restaurant whenever possible. Neither wishes to eat at that particular restaurant alone and each dislikes waiting for the other, so they agree that:

1. Each will arrive at a random time between noon and 1:00 p.m.
2. Each will wait for the other either for 15 minutes or until 1:00 p.m.

On a given day, what is the probability that the friends will meet for lunch?

At first glance, it seems obvious that they will wait 30 minutes between them. So the probability is 0.50. As we will see later, this reasoning is wrong. Each friend can arrive at any instant between noon and 1:00 p.m. If the arrival at each instant is equally probable, then the probability that the friends will arrive at the same instant is of course zero. However, in this case, each friend will have to wait for each other for 15 minutes. Hence, there will be an interval of time on certain days when they will meet and have lunch together.

**Solution via Simulation**

Since the arrival time of the friends are equally probable between noon and 1 p.m. (one hour period), we can randomly generate two uniformly distributed random numbers between 0 and 1 to represent the arrival times of each friend. If the arrival times are within 15 minutes (0.25) hour of each other, then they meet. The total times they meet in N days is the required probability.

On MINITAB the MEETING.MAC will perform the simulation. When the macro is executed, the number of days, N, to simulate will be requested. Then two columns of N rows of uniformly distributed random numbers between 0 and 1 will be
generated using the RANDOM command and UNIFORM subcommand, and put in columns C1 and C2. If the absolute difference between the two column of numbers is less than or equal to 0.25 (15 minutes) then there is a meeting. The number of meetings is kept track of in C3. The number of meetings divided by N is the required probability. Display 3.15 shows a sample run with the MEETING macro. Here 100 days were simulated. The meeting macro is given in Display 3.16.

**Display 3.15: Sample MINITAB session with MEETING macro**

```
MTB>%MEETING
Number of days? (e.g. 100)
DATA>100
Probability of meeting for lunch is k2:
  k2       0.450000

MTB>
```

**Display 3.16: MEETING macro**

```
MEETING.MAC
GMACRO
MEETING
ERASE c1-c50
NOTE Number of days to simulate? (e.g. 100)
SET c50;
FILE “terminal”;
NOBS 1.
COPY c50 k10
RANDOM k50 c1-c2;
UNIFORM 0 1.
LET c3=(ABS(c1-c2)<0.25)
LET k2=SUM(c3)/k10
NOTE Probability of meeting for lunch is k2:
NOTE
PRINT k2
ENDMACRO
```
Analytical Solution to the Meeting for Lunch Problem

Let \( x \) and \( y \) be the arrival times of the two friends measured in parts of an hour from 12 noon to 1 pm, respectively. See Figure 3.1. A point \((x, y)\), within the square will represent a possible set of arrival times for friend 1 and friend 2. The shaded region of the display shows the arrival times for which the friends meet. That is, any point that lies within the shaded region represents arrival times that are 15 minutes or less apart.

![Figure 3.1: Geometrical Representation of Meeting for Lunch Problem](image)

Comparing the area of the non-shaded triangles to unity, we see that the problem of time that the friends do not meet for lunch is:

\[
2 \times \text{areas of each triangle} = 2 \times \frac{1}{2} \times \frac{3}{4} \times \frac{3}{4} = \frac{9}{16}
\]

So the proportion of time they do meet is \(1-(9/16)=7/16=0.4375\).

The simulation result as can be seen is in close agreement with the analytical result.
3.7 Exercises

Some of the problems given here are similar to those discussed in this chapter. All can be easily solved by simulation.

3.1 This is another problem from Mosteller’s book.

Samuel Pepys wrote Newton to ask which of three events is more likely: that a person get (a) at least 1 six when 6 dice are rolled, (b) at least 2 sixes when 12 dice are rolled, or (c) at least 3 sixes when 18 dices are rolled. What is the answer?

3.2 In a family with five kids, what is the probability that at least two of the kids are girls? This is the same as asking the probability of obtaining at least two heads in 5 tosses of a fair coin. In fact the same question can be asked in many other contexts.

3.3 If 6 of 18 new buildings in a city violate the building code, what is the probability that a building inspector, who randomly selects four of the new buildings for inspection, will catch a) none of the new buildings that violate the building code; b) at least three of the new buildings that violate the building code?
4. MORE ADVANCED PROBABILITY PROBLEMS

This chapter considers slightly more advanced probability problems. A few more theoretical discrete and continuous distributions will be introduced. In addition, it will be shown how conditional probability problems and those involving the use of the total probability and Bayes theorems can be easily explained and solved using simulation.

4.1 The Archer’s Problem

The archer’s problem considers the case where there are more than 2 possible outcomes for a given trial. This problem is unlike the previous binomial problems (e.g. basketball and counterfeiter) where there are only two possible outcomes. Problems where there are more than two outcomes are called multinomial problems. The probability of each outcome can be unequal, but the sum of the probabilities of each outcome must of course be unity.

Problem: Based on past performances, an archer puts 10% of his shots in the bullseye, and 60% of his shots in the white ring around the bullseye, and misses 30% of his shots. How likely is it that in three shots the archer will get exactly one bullseye, two in the white, and no misses?

As you can see this problem is slightly more complicated than the basketball or the counterfeiter problem. If the problem is simplified to the case where the archer either
hits the bullseye or misses the bullseye, than the problem reduces to the binomial case. Here we have to deal with the third outcome.

**Solution via Simulation**

There are several ways in which the archer’s problem can be solved using simulation. Since the probabilities are nice round numbers, one approach is to use integers from 1 to 10, where “1” would represent the bullseye, “2 to 7” would represent the white, and “8 to 10” would represent misses. Randomly generate 3 numbers (between 1 and 10), and check whether there are one “1” and two numbers between “2 to 7”. If so, record this as a success or “yes”, otherwise “no”. Repeat the last step a large number of times and the proportion of “yeses” recorded would be the required probability.

On MINITAB the macro ARCHER.MAC will perform the simulation. A different approach for simulation was used in the macro. Sampling from a discrete distribution was used instead of numbers from 1 to 10 as discussed above. The reason for using the discrete distribution approach is that it is easier to define the probabilities of a bullseye, white, or miss. The integers 1, 2, and 3 in column C1 represent bullseye, white, and miss, and column C2 contains the respective probabilities. A random sample of 3 is then taken from C1 according to the probabilities in C2 using the RANDOM command and DISCRETE subcommand. These 3 numbers are put into C3. The number “1” appearing in C3 is kept track of in C4, the number “2” is kept track of in C5, and the number “3” is kept track of in C6. The number of “1”, “2”, and “3” in each trial is kept in C7, C8 and C9, respectively. Column C10 then keeps track of the cases where there are one “1”, two “2”, and zero “3” in columns C7, C8 and C9, respectively. The number
of cases where the conditions are met divided by the number of trials gives the required probability.

Display 4.1 shows a sample run with the ARCHER macro. Here 1000 trials has been used in the simulation. The ARCHER.MAC macro is given in Display 4.2.

**Display 4.1: Sample MINITAB session with ARCHER macro**

<table>
<thead>
<tr>
<th>MTB=&gt;%ARCHER</th>
<th>DATA&gt;1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of trials? (e.g. 1000)</td>
<td>Probability of one bullseye, two in white and no misses in 3 shots in k2:</td>
</tr>
<tr>
<td>K2 0.107000</td>
<td>MTB&gt;</td>
</tr>
</tbody>
</table>

**Display 4.2: Archer macro**

```plaintext
ARCHER.MAC

GMACRO
ARCHER
ERASE c1-c50
SET c1
1 2 3
END
SET c2
.1 .6 .3
END
NOTE Number of trails? (e.g. 100)
SET c50;
FILE “terminal”; 
NOBS 1.
COPY c50 k50
DO k1=1:k50
RANDOM 3 c3;
DISCRETE c1 c2.
LET c4=(c3=1)
LET c5=(c3=2)
LET c6=(c3=3)
LET c7(k1)=SUM(c4)
LET c8(k1)=SUM(c5)
```
Analytical Solution to the Archer’s Problem

The solution to the archer’s problem requires the use of the multinomial distribution. This distribution is a generalization of the binomial distribution. The probability function of a multinomial distributed random variable is:

\[
\frac{M!}{m_1!m_2!\cdots m_k!} p_1^{m_1} p_2^{m_2} \cdots p_k^{m_k}
\]

in which \( m_i \) is the number of trials in which outcome \( i \) occurred, \( p_i \) is probability that outcome \( i \) will occur on a trial, and \( M \) is the total number of trials in the experiment and is equal to the sum of the \( m_i \)'s. It is assumed that the probabilities will remain constant from trial to trial.

In the context of the archer’s problem,

\[
M = 3, m_1 = 1, m_2 = 2, m_3 = 0, p_1 = 0.1, p_2 = 0.6, p_3 = 0.3.
\]

Substituting these values into the multinomial probability function gives:

\[
Pr(1 \text{ bullseye, 2 whites and 0 misses}) = \frac{3!}{1!2!} 0.1^1 0.6^2 0.3^0 = 0.1080
\]

As you can see, the simulation results gave a fairly accurate answer.
4.2 Flying Off in Time

The following problem is adapted from an exercise given in Ang and Tang (1975). This problem involves two different probability distributions, and dependency. With this problem the concepts of conditional probability, total probability, and Bayes theorem are introduced. These more advanced probability concepts that have many practical applications.

Problem: The waiting time at a very busy airport is equally likely to range from 1 to 4 hours. The waiting time is measured from the time a traveler enters the terminal to the time when she is airborne. The travel time from the hotel to the airport depends on the transportation mode and may be assumed to be 0.75, 1.00, and 1.25 hours corresponding to travel by rapid transit, taxi, and limousine, respectively. The probability of the traveler’s taking each mode of transportation is as follows:

\[
\begin{align*}
P(\text{rapid transit}) &= 0.3 \\
P(\text{taxi}) &= 0.5 \\
P(\text{limousine}) &= 0.2
\end{align*}
\]

What is the probability that the traveler will be airborne within 3 hours after leaving the hotel? If the traveler is airborne within 3 hours, what is the probability that she took the taxi?

Solution via Simulation

This problem can be easily solved using simulation. First we randomly generate the mode of transportation which will give us the time by that mode of transportation. Then we randomly generate the waiting time at the airport. The sum of the transportation time and
waiting time gives the total time from the hotel to airborne. In MINITAB, first enter in column C1 the travel time by each mode of transportation. In column C2 we put the respective probabilities. We then use the RANDOM command and DISCRETE sub-command to generate the mode of transportation many times (say 1000) and put the results in column C3. We then use the RANDOM command and UNIFORM sub-command to generate an equal number of times the waiting time at the airport which is a uniform random number between 1 and 4. Store this in C4. The sum of the ground transportation time and waiting time gives the total time to airborne and this is stored in column C5. Column C6 keeps track of the cases where the total time is less than or equal to 3 hours. The probability that the traveler is airborne within 3 hours is then the number of times that the total time was less or equal to 3 hours divided by 1000. To determine the probability that the traveler have used the taxi given that she was airborne within 3 hours, we need to count the number of times that she was airborne within 3 hours and had used the taxi (a joint event). Then the conditional probability is given by the number of times the joint event occurred divided by the number of times she was airborne with 3 hours. The joint event condition kept in column C7. The FLYING macro is given in Display 4.7 and a sample session using the macro is shown in Display 4.8.

**Display 4.7: Flying macro**

```
GMACRO
FLYING
NOTE Enter the number of trials (e.g. 1000)
SET c50;
FILE "terminal";
NOBS 1.
COPY c50 K50
RANDOM k50 c3;
DISCRETE c1 c2.
```
RANDOM k50 c4;
UNIFORM 1 4.
LET c5=c3+c4
NAME c3 'Tran Time' c4 'Wait Time' c5 'Total Time'
LET c6=(C5<=3)
LET k1=sum(c6)/k50
NOTE
NOTE The probability that she is airborne within 3 hours is k1
PRINT k1
NOTE
LET c7=(c5<=3 and c3=1.0)
LET k2=sum(c7)/sum(c6)
NOTE The probability that she took the taxi given
NOTE that she was airborne within 3 hours is k2:
PRINT k2
ENDMACRO

Display 4.8: Sample session with the FLYING macro

MTB > %FLYING
Enter the number of trials (e.g. 1000)
DATA> 1000

The probability that she is airborne within 3 hours is k1
K1    0.340

The probability that she took the taxi given
that she was airborne within 3 hours is k2:
K2    0.484

Analytical Solution to the Flying Off in Time Problem

The analytical solution to this problem requires the use of conditional probability, total probability, and Bayes theorems. A simple way to display the information we have and then to solve the problem is to use a two-way table. The general layout of the table and
where the correct information should be put is shown first. Calculations of the required probabilities to fill the table follow the basic laws of probabilities.

**Table 4.1: Two-way table for the flying off in time problem**

<table>
<thead>
<tr>
<th>Transportation Mode, Travel Time, and Probability</th>
<th>RT (T=0.75 h)</th>
<th>Taxi (T=1 h)</th>
<th>Limo (T=1.25 h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P(RT) = 0.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P(Taxi) = 0.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P(Limo) = 0.2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Airborne within 3 h (A)</th>
<th>P(A 1 RT)</th>
<th>P(A 1 Taxi)</th>
<th>P(A 1 Limo)</th>
<th>P (A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P(A</td>
<td>RT)</td>
<td>P(A</td>
<td>Taxi)</td>
<td>P(A</td>
</tr>
<tr>
<td>Not Airborne within 3 (A’)</td>
<td>P(A’1 RT)</td>
<td>P(A’1 Taxi)</td>
<td>P(A’1 Limo)</td>
<td>P(A’)</td>
</tr>
<tr>
<td>P(A’</td>
<td>RT)</td>
<td>P(A’</td>
<td>Taxi)</td>
<td>P(A’</td>
</tr>
</tbody>
</table>

Let A be the event that the total time to airborne is within 3 hours, and A’ be the complement of A (not within 3 hours). To determine the probability of A, we need to apply the total probability theorem. That is,

\[
\]

where, \( P(A 1 RT) = P(A|RT) P(RT) \),

\( P(A 1 Taxi) = P(A|Taxi) P(Taxi) \), and

\( P(A 1 Limo) = P(A|Limo) P(Limo) \)

If rapid transit is taken, which takes 0.75 h, for the traveler to be airborne within 3 hours, the waiting time (WT) at the airport must be less than or equal to 2.25 hours. In other words, \( P(A|RT) = P(WT \leq 2.25) = 0.417 \). Similarly, \( P(A|Taxi) = P(WT \leq 2.0) = 0.333 \), and \( P(A|Limo) = P(WT \leq 1.75) = 0.250 \). Substituting these into the above equations, we get:
\[ P(A) = 0.417 \times 0.3 + 0.333 \times 0.5 + 0.250 \times 0.2 = 0.342 \]

\[ P(A') = 1 - P(A) = 0.658 \]

The probability that the traveler took the taxi given that she was airborne within 3 hours is given by:

\[ P(\text{Taxi}|A) = P(A|\text{Taxi}) \times P(\text{Taxi})/P(A) = 0.333 \times 0.5/0.342 = 0.487 \]

The results obtained by simulation were quite close to the analytical results. With more replications, the results would be even closer.

### 4.3 Beware of Icebergs

This problem illustrates the use of conditional and total probability concepts to solve an interesting real risk analysis problem. Again, we have a discrete and a continuous distribution. But, this time, the discrete distribution is used to model the potential number of icebergs arriving at a site per year where an offshore structure is to be built and the continuous distribution is used to model the magnitude of the force when the iceberg hits the structure. Since not all hits are damaging to the structure, we are only concerned with those hits that are above a certain magnitude and to determine the probability of that happening each year.

**Problem:** Consider the design of an offshore structure in the ice infested waters of the North Atlantic Ocean. The damage to the offshore structure due to hits by moving icebergs must be considered. From data collected at the potential site, the number of icebergs that is likely to hit the structure per year has the probability distribution shown in Figure 4.1.
The magnitude of the iceberg impact force on the structure is assumed to follow a normal distribution with a mean of 80 MN and a standard deviation of 20 MN. Damage to the structure only occurs when the magnitude of the impact force is greater than 100 MN, what is the probability that there will be at least one damaging iceberg hit in any given year?

**Solution via Simulation**

The number of icebergs (m = 1, 2, 3, ..., or 7) hitting the structure each year follows the discrete probability distribution given in Figure 4.1. On MINITAB, the numbers 1 to 7 are put in column C1 and the corresponding probabilities in C2. To simulate the number of hits (given in C1) for a year, according the probabilities given in C2, we use the RANDOM command and the DISCRETE subcommand. This generated number (m) is then put in the first row of column C3. Next, m normal variates with mean = 80 and standard deviation = 20 are generated using the RANDOM and NORMAL sub-command and put in column C4. Then we check if these generated values in C4 are greater than
100. This check is indicated in C5. Each row of C5 contains either a “0” or a “1” depending on whether the corresponding row values in C4 is less than 100 or greater or equal to 100, respectively. This can be done using the LET command. The sum of C5, which is the number of hits with magnitudes greater than or equal to 100, is then stored in C6. The process is repeated over many years (trials), say 1000, and the number of trials where 0, 1, 2, 3, … damaging hits are observed per year over the total number of trials can be displayed in a histogram. The number of times that that will be at least one damaging hit can then be check from C6 and stored in C7.

Display 4.5 shows the ICEBERG.MAC macro that will perform the simulation and a sample run with the ICEBERG macro with 1000 replications is given in Display 4.6.

Display 4.5: Iceberg macro

ICEBERG.MAC
GMACRO
ICEBERG
ERASE C1-C50
NOTE Number of years (trials) to simulate ? (e.g. 1000)
SET c50;
FILE “terminal”;
NOBS 1.
COPY c50 k50
DO k1=1:k50
RANDOM 1 c3;
DISCRETE c1 c2.
COPY c3 k2
RANDOM k2 c4;
NORMAL 80 20.
LET c5 = (C4 >= 100)
LET c6(k1)=SUM(c5)
ENDDO
LET c7=(c6>=1)
LET k3=SUM(c7)/k50
NOTE Probability of at least one damaging hit is k3
PRINT k3
NOTE
NOTE Histogram of the number of damaging iceberg hits
GSTD
HISTOGRAM c6
ENDMACRO

Display 4.6: Sample MINITAB session with ICEBERG macro

MTB > %ICEBERG

No. of trials to simulate? (e.g. 1000)
DATA> 1000

Probability of at least one damaging hit is k3

Data Display
K3  0.351000

Histogram of the number of damaging iceberg hits

Histogram
Histogram of C6  N = 1000
Each * represents 15 observation(s)

Midpoint    Count
0           649 ########################################################
1           289  *********************
2            52  ****
3             7  *
4             3  *

MTB>
**Analytical Solution to the Beware of Icebergs Problem**

Analytical solution to this problem requires the use of the total probability theorem and a table of the cumulative normal distribution table. When there is a hit, the probability that the magnitude is greater than 100 MN is:

\[
P(X \geq 100) = P\left(z \geq \frac{100 - 80}{20}\right) = P(z \geq 1.0) = 0.1587
\]

Where \(X\) is the magnitude of the iceberg hit in MN. If we let the event \(D\) = at least one damaging iceberg hits in a year, and \(D'\) = no damaging iceberg hits in a year,

\[
P(D') = P(D'|n=1) P(n=1) + P(D'|n=2) P(n=2) + P(D'|n=3) P(n=3) + P(D'|n=4) P(n=4) + \ldots + P(D'|n=7) P(n=7)
\]

Since iceberg arrivals are independent events, the probability of no damaging iceberg hits with \(n\) icebergs per year is \(0.8413^n\). Substituting into the total probability equation above, we get:

\[
P(D') = 0.8413 \times 0.25 + 0.8413^2 \times 0.30 + 0.8413^3 \times 0.20 + 0.8413^4 \times 0.15 + 0.8413^5 \times 0.05 + 0.8413^6 \times 0.03 + 0.8413^7 \times 0.02 = 0.6584
\]

Hence, the probability of at least one damaging iceberg hit in a year is,

\[
P(D) = 1 - P(D') = 1 - 0.6584 = 0.3416
\]

As can be seen, the result from simulation is quite close to the analytical result. The accuracy of the result can be improved by using more replications.

### 4.4 Rainy Weeks and Total Rainfall

This problem is adapted from a similar problem given in Benjamin and Cornell (1970). It is perhaps the most difficult problem in this book to solve analytically. However, it will be shown that the problem is quite easily solved using simulation with MINITAB. In fact
the problem is very similar to the previous problem except that now we are dealing with
the Poisson distribution and an exponential distribution.

**Problem:** In the fair City of St. John’s, the total number of rainy weeks, \( N \), with at least
a trace of rain, can be approximately modeled by the Poisson distribution with a mean of
20 weeks/year; and the distribution of the rainfall amount \( R_i \) (in cm) in the \( i^{th} \) rainy week
can be modeled by an exponential distribution with a mean of 2.0 cm. If the total annual
rainfall, \( T \), is given by:  \[ T = \sum_{i=1}^{N} R_i \], what is the probability that the total annual rainfall
will exceed 60 cm in any year?

**Solution via Simulation**
The number of rainy weeks (\( N = 1, 2, 3, \ldots \)) each year follows the Poisson distribution
with a mean of 20 weeks/year. On MINITAB, to simulate the number of rainy weeks, we
use the RANDOM command and the POISSON subcommand. This generated number
(\( N \)) is then put in the first row of column C1. Next, \( N \) exponential variates with mean =
2.0 are generated using the RANDOM and EXPO sub-command and put in column C2.
Then we sum up these generated values and put the sum in column C3. The process is
repeated over many years (trials), say 1000. Column C3 would now contain 1000 years
of total annual rainfall and can be displayed in a histogram. Total annual rainfall that
exceeds 60 cm can be checked using the LET command and stored in C4. The
probability that the annual rainfall exceeding 60 cm is then the sum of C4 divided by
1000.
Display 4.7 shows the RAINFALL.MAC macro that will perform the simulation and a sample run with the RAINFALL macro with 1000 replications is given in Display 4.8. This macro contains an IF statement to check whether a zero rainy week is generated. If there is no rainy week during the year, then there would be no rainfall amount for the year. This check is necessary because MINITAB must generate one or more variates, otherwise, the macro will stop running and an error message given.

**Display 4.7: Rainfall macro**

```
GMACRO
RAINFALL
ERASE c1-c50
NOTE No. of trials ? (e.g. 1000)
SET c50;
FILE "terminal";
NOBS 1.
COPY c50 k50
DO k1=1:k50
RANDOM 1 c1;
POISSON 20.
COPY c1 k51
IF k51=0
   LET c2(1)=0
   LET c3(k1)=0
   NEXT
ELSE
   RANDOM k51 c2;
EXPO 2.
LET c3(k1)=SUM(c2)
ENDIF
ENDDO
GSTD
HISTOGRAM c3
LET c4=(c3>=60)
LET k2=SUM(c4)/k50
NOTE The probability that the total annual rainfall is greater or equal to 60 cm is k2:
PRINT k2
ENDMACRO
```
Display 4.8: Sample MINITAB session with Rainfall macro

MTB > %RAINFALL

No. of trials ? (e.g. 1000)
DATA> 1000

Histogram

Histogram of C3   N = 1000
Each * represents 10 observation(s)

<table>
<thead>
<tr>
<th>Midpoint</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>20</td>
<td>104</td>
</tr>
<tr>
<td>30</td>
<td>264</td>
</tr>
<tr>
<td>40</td>
<td>289</td>
</tr>
<tr>
<td>50</td>
<td>214</td>
</tr>
<tr>
<td>60</td>
<td>82</td>
</tr>
<tr>
<td>70</td>
<td>28</td>
</tr>
<tr>
<td>80</td>
<td>3</td>
</tr>
<tr>
<td>90</td>
<td>3</td>
</tr>
<tr>
<td>100</td>
<td>1</td>
</tr>
</tbody>
</table>

The probability that the total annual rainfall is greater or equal to 60 cm is k2:

Data Display

k2  0.0640000

MTB>

Analytical Solution

The analytical solution is not straightforward. One would have to know that the sum of n independently and identically distributed exponential random variables is gamma distributed. That is, the distribution of T is gamma. Interested readers should refer to Benjamin and Cornell’s book where the full analytical solution is given.
Exercises

Each problem below can be solved by using simulation on Minitab.

4.1 This problem was discussed in Julian Simon’s book, Resampling: The New Statistics:

If the archer problem in Problem 4.1 of Chapter 4 gets three points for a bullseye and one point for white, and if her probabilities are as those in Problem 4.1, what is the probability that the archer will get eight or more points in seven shots?

4.2 Six women went to a party and each was wearing a distinctive hat. Each of them left their hat at the coat check counter on their way in. Unfortunately, the coat-checker did not issue any of them with a receipt. What is the probability that the coat-checker will be able to match each hat with its rightful owner if the checker has no prior knowledge as to which hat belongs to whom? What is the probability that none of the women will get their hat back? What is the probability that there will be one, two, three, four, five, or six matches?

4.3 In the design of a dam for flood control, the height of the dam to be designed for depends on two factors: the amount of flood waters coming into the reservoir behind the dam (Y), and the water level of the water reservoir when the flood waters arrives (X). The water level in the reservoir during the flood season ranges from 20 to 45 m with equal probability. The amount of flood waters entering the reservoir has a mean of 10 m and follows an exponential distribution. What is the probability that the dam will be overtopped is the dam is 60 m high?
5. REFERENCES


