Statistical Computing

1. Understanding Randomness and Random Numbers

Sudhakar Kunte

Sudhakar Kunte is Professor of Statistics in the Department of Statistics, University of Pune. He obtained his PhD in Statistics from the Purdue University, USA in 1973. His current research interests include Bayesian inference and finite population sampling.

Elements of statistical computing are discussed in this series. We begin with the notion of random numbers and their generation using computers. Using these computer generated random numbers for statistical computing and simulations is discussed in the later parts of the series.

The first step in a cricket match is the toss. It decides as to which captain gets an option to decide whether to field first or bat first. In most tournaments the first step is to make a draw. This is usually done by drawing chits or some other method equivalent to it. In most card games the first step is to shuffle the cards before dealing them. Even in a final match of the world chess championship between two players, who plays the first game with white pieces is decided by a toss. Why leave such decisions to chance? Perhaps there is no better way to decide. Depending upon weather conditions on the particular day, the condition of the pitch and other factors, it may be of advantage for a team to bat first. Should any one team get this advantage? Obviously not. On the other hand, one of the teams has to bat first. What the toss does is to give equal chance for either team to get this advantage. In this sense deciding by toss is an unbiased or fair procedure. The final outcome of the toss may of course not be fair, in the sense that the team which wins the toss would get the advantage. How the team utilizes this advantage is a different matter. In a lottery it is finally only one ticket which wins the big prize. However the lottery is fair or unbiased as long as the procedure of drawing the winning number is such that it gives equal chance for all the numbers to get selected. A number R is said to be a random number between 1 and N, if R is selected in such a way that it is equally likely to be any number between 1 and N. Thus

Probability $(R = r) = 1/N$, for all $r = 1, 2, \ldots, N$.

Consider the following questions:

Q1: Is 4 an even number?

Q2: Is 4 an odd number?

Q3: Is 4 a prime number?

Q4: Is 4 a random number between 1 and 99?

The first three questions have well-known answers as yes, no and no respectively. Q4 cannot be answered unless we know how 4 was selected. In this question *randomness* is the property of the procedure which is used to select the number 4. Thus Q4 cannot be answered unless we know this selection procedure. Quite often in practice the word 'random number' is misused in the following sense:

'A number which is selected without any well-defined rule is a random number.'

This is not the way a random number is defined or a number is selected at random. The procedure used to select a random number should satisfy certain properties in order that its outcome is acceptable as a random number. A toss which is made by a 'shole' coin which has both heads is not a fair toss or a random toss. How does one draw such a proper random number? We describe two methods of drawing a random number between 0 and 99.

Method 1: Make 100 chits with numbers 00 to 99 written on one chit each. Fold these chits. Put them in a large box. Mix them thoroughly and draw one chit. The number on the selected chit is a random number between 0 to 99.

Method 2: Make only ten chits with numbers 0 to 9. Follow the procedure of method one for two draws, putting back the chit of the first draw before second draw is made. Put the digit selected on the first draw at the tens place and the second digit at unit's place to get a two-digit number. The outcome is a random number between 00 to 99.

Suppose a long sequence of digits from 0 to 9 has been obtained by repeated applications of a random mechanism such as the

The procedure used to select a random number should satisfy certain properties in order that its outcome is acceptable as a random number.

ones described above in such a manner that the repeated applications can be considered to be unrelated (*independent* in the technical sense in which it is used in probability theory). Such a sequence is called a random sequence of digits. Such sequences are often presented in the form of a table; such tables are called random-number tables. Random-number tables are included as appendices in most books dealing with statistical methods in the pre-computer era. One can read such a table at an arbitrarily selected starting point and select the next two-digit number and that would serve as a random two-digit number.

It is easy to think of several experiments which can be physically performed to generate random numbers. The following are some examples:

- Tossing a fair coin and observing the face up. This experiment generates digit 0 (tail) or 1 (head) at random.
- Throwing a fair die and observing the face turning up. This experiment generates a random number between 1and 6.
- Designate the cards of a deck of playing cards with numbers from 1 to 52 (in any manner you like). Shuffle the deck thoroughly and select a card. Read the number on the card. This experiment generates a random number between 1 and 52.

In order to generate a random number, we have to perform a physical experiment with a number of equally likely outcomes. Can a computer generate a random number? A computer essentially performs logical and numerical operations on the given inputs according to an algorithm supplied by the program. Thus on a computer, given an input and the program, the outcome is completely deterministic. There is nothing random about the outcome. However there are some numerical algorithms which when performed generate a set of numbers (or a sequence of numbers) which satisfy some of the more important properties of a set (or a sequence) of random numbers. Hence they are called pseudo-random numbers. The algorithms which generate such pseudo-random numbers are called *ran-*

Can a computer generate a random number?

dom-number generators. Such random-number generators are programs and using these programs the computer generates numbers which are essentially pseudo–random numbers.

The test of a digit being random is simply that 'the mechanism generating that number gives equal chances for the ten digits from 0 to 9'. The outcome itself does not tell you anything. To make the use of random numbers valid, it is not only necessary that each digit is generated by a random mechanism as above, but also that the various digits are generated *independently;* for instance, if you want to select one out of 100 (designated 00 to 99) using two adjacent digits of a sequence of random digits, the two digits should be also independent so that the one hundred two-digit numbers each have a 1/100 probability. Further, in many applications several k-digit random numbers are needed and users tend to take consecutive sets of k digits for this purpose. Thus this independence property should hold good for any collection of digits in the set, more importantly for collections of adjacent k digits. Thus the randomness property should be viewed as a property of the entire sequence of digits, not merely the entire set of digits.

What do we mean by saying that a sequence of numbers looks like a random-number sequence? As we said before, they satisfy some of the more important properties of a sequence of numbers generated by a genuinely random experiment like coin-tossing or rolling a die or shuffling a pack of cards. What are these properties? They are:

1. The observed frequencies of the ten digits from 0 to 9 in the sequence are equal, but for chance variation.
2. There are no simple and easily discernible patterns in the sequence – this is not a clearly stated property and in practice one looks for absence of some simple patterns in the sequence.

   (a) There should not be any visible trend – increasing, decreasing or other types – in the sequence.
   (b) There should not be any cyclic pattern in the sequence; that is, there should not be any significant correlation between

What do we mean by saying that a sequence of numbers looks like a random-number sequence?

elements one apart, two apart, ..., k apart.

For each such property we can have a statistical test using which we can decide whether the sequence of numbers satisfies this property or not and thereby decide if it satisfies the properties required of a sequence of pseudo-random numbers. The advantage of having such random-number generators on a computer is that a large number of (pseudo-)random numbers can be generated with ease and in a short time. These can then be used in statistical applications like conducting a sample survey or designing a controlled experiment or in simulation studies. In such studies, often a long sequence of random numbers are to be generated repeatedly, followed by some heavy and complex computational process executed on a high-speed computer – a task which was practically impossible before the advent of high-speed computers.

Random Number Generators: As discussed above a random number generator is a program which generates a sequence of numbers. Such a program starts with a user specified number x_0 called the seed. Using a mathematical formula the program successively generates numbers $x_1, x_2, \ldots$. We give below examples of some random number generators.

Middle Square Algorithm: This is perhaps the first algorithm proposed by Von-Neumann to generate sequences of random numbers. This algorithm generates a sequence of $2a$ digit numbers with base b. Suppose x_n is a $2a$ digit number (possibly having a few zeros in the beginning). We square x_n to get a $4a$ digit number. Remove the a digits to the left and a digits to the right and retain the middle $2a$ digits. This retained $2a$ digit number is x_{n+1}. As a mathematical formula we have

$$x_{n+1} = [x^2{}_n / b^a] - [x^2{}_n / b^{3a}]\, b^{2a}.$$

Here $[x]$ represents the integral part of x.

Linear Congruential Algorithm: This algorithm was proposed in 1948 by D L Lehmer. It generates x_{n+1} from x_n using the

mathematical formula

$$x_{n+1} = (ax_n + c) \bmod m,$$

where a, c and m are user specified integers and (x) mod m represents the remainder in x obtained by taking away the largest multiple of m from x. The numbers generated by the algorithm range from 0 to m–1. The proper choice of a, c, m should be such that

1. c is relatively prime to m,
2. $(a-1)$ is a multiple of q, for each prime factor q of m,
3. $(a-1)$ is a multiple of 4, if m is a multiple of 4.

Several modifications of these algorithms have been proposed over a period of time. Note that these algorithms are based upon integral arithmetic. Whenever a number repeats in the sequence the following sequence will also repeat and so there will be cycles in the sequence. Thus these methods are not useful in generating long sequences of random numbers.

Most computer languages and statistical packages these days have random number generators as a built-in mathematical function. Most of these are still based on modifications of the Linear Congruential Algorithm. Some of them, however, are based on complicated mathematical formulae and non-integral arithmetic, and these functions generate sequences of pseudo-random numbers between 0 and 1 in terms of their decimal expansion having several digits after the decimal point. Multiplying such numbers by 10 and taking the integral part we get a random digit. In general to obtain a random number between 1 and N, we multiply the random number generated by the computer by N, take its integral part and add one. Thus if R represents a random number generated by the computer and we define

$$X = [R * N] + 1,$$

then X is a random number between 1 to N. In the next part of this series we shall address the real use of these random numbers in statistical computing.

Suggested Reading

[1] R L Karandikar, On Randomness and Probability: How to Model Uncertain Events Mathematically, *Resonance*, Vol.1, No.2, 55–68, 1996.

[2] H Ramesh and V Vinay, Who Will Win the Toss? *Resonance*, Vol.3, No.4, 72–87, 1998.

Address for Correspondence
Sudhakar Kunte
Department of Statistics
University of Pune
Pune 411 007, India.
Email:
skunte@stats.unipune.ernet.in