**Chapter three**

**Random-Number Generation**

Random numbers are a necessary basic ingredient in the simulation of almost all discrete systems. Most computer languages have a subroutine, object, or function that will generate a random number. Similarly simulation languages generate random numbers that are used to generate event limes and other random variables.

**Properties of Random Numbers**

A sequence of random numbers, R1, R2, , must have two important statistical properties, uniformity and independence. Each random number *Ri,* is an independent sample drawn from a continuous uniform distribution between zero and 1. That is, the pdf is given by

**F(x) =** {**1, 0** ≤**x**≤**1**

**0, otherwise**

This density function is shown in Figure 3.1. The expected value of each *Ri,* is:

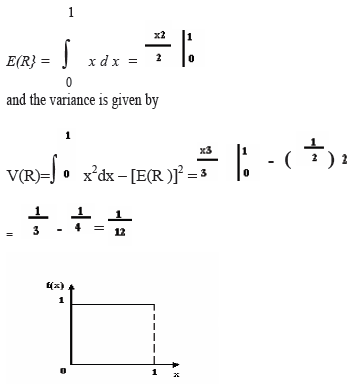


Figure 3*.1\** the: pdf for random numbers

**Some consequences of the uniformity and independence properties are the following:**

1. If the interval (0,1) is divided into *n* classes, or subintervals of equal length, the expected number of observations m each interval is *N/n* where A' is the total number of observations.
2. The probability of observing a value in a particular interval is of the previous values drawn

**Generation of Pseudo-Random Numbers**

Notice that the title of this section has the word “pseudo” in it.”Pseudo” means false, so false random numbers are being generated. The goal of any generation scheme is to produce a sequence of numbers between zero and 1 which simulates, or initiates, the ideal properties of uniform distribution and independence as closely as possible.

When generating pseudo-random numbers, certain problems or errors can occur. These errors, or departures from ideal randomness, are all related to the properties stated previously. Some examples include the following

1. The generated numbers may not be uniformly distributed.
2. The generated numbers may be discrete -valued instead continuous valued
3. The mean of the generated numbers may be too high or too low.
4. The variance of the generated numbers may be too high or low
5. There may be dependence. The following are examples:

(a) Autocorrelation between numbers.

(b) Numbers successively higher or lower than adjacent numbers.

(c) Several numbers above the mean followed by several numbers below the mean

Usually, random numbers are generated by a digital computer as part of the simulation.

Numerous methods can be used to generate the values. In selecting among these methods, or routines, there are a number of important considerations.

1. The routine should be fast. . The total cost can be managed by selecting a computationally efficient method of random-number generation.
2. The routine should be portable to different computers, and ideally to different programming languages .This is desirable so that the simulation program produces the same results wherever it is executed.
3. The routine should have a sufficiently long cycle. The cycle length, or period, represents the length of the random-number sequence before previous numbers begin to repeat themselves in an earlier order. Thus, if 10,000 events are to be generated, the period should be many times that long. A special case cycling is degenerating. A routine degenerates when the same random numbers appear repeatedly. Such an occurrence is certainly unacceptable. This can happen rapidly with some methods.
4. The random numbers should be replicable. Given the starting point (or conditions), it should be possible to generate the same set of random numbers, completely independent of the system that is being simulated. This is helpful for debugging purpose and is a means of facilitating comparisons between systems.
5. Most important, and as indicated previously, the generated random numbers should closely approximate the ideal statistical properties of uniformity and independences.

**Techniques for Generating Random Numbers**

The linear congruential method is the most widely used technique for generating random numbers, so we describe in detail. We also report an extension of this method that yields sequences with a longer period.

**Linear Congruential Method**

The linear congruential method, initially proposed by Lehmer [1951], produces a sequence of integers, *X1*, *X2,...* between zero and *m —* 1 according to the following recursive relationship:

**Xi+1 = (aXi + c) mod m, i = 0, 1, 2,... (3.1)**

The initial value **X0** is called the seed, **a** *is* called the constant multiplier, **c** is the increment, and **m**is the modulus. If ***c* ≠0**, then the form is called the mixed **congruential method*.***When *c =* 0, the form is known as the ***multiplicative congruential method****.* The selection of the values for **a, *c,*****m** and **Xo** drastically affects the statistical properties and the cycle length. . An example will illustrate how this technique operates.

**EXAMPLE 3.1**

Use the linear congruential method to generate a sequence of random numbers with ***X0 =* 27, *a=*****17, *c =* 43**, and ***m =* 100**. Here, the integer values generated will all be between zero and 99 because of the value of the modulus. These random integers should appear to be uniformly distributed the integers zero to 99.Random numbers between zero and 1 can be generated by

***Ri =Xi/m, i= 1,2,…… (3.2)***

The sequence of Xi and subsequent Ri values is computed as follows:

X0 = 27

X1 = (17**.**27 + 43) mod 100 = 502 mod 100 = 2

R1=2⁄100=0. 02

*X2* = (17 • 2 + 43) mod 100 *=* 77 mod 100 = 77

R2=77 ⁄100=0. 77

X3 = (17•77+ 43) mod 100 = 1352 mod 100 = 52

R3=52 ⁄100=0. 52

First, notice that the numbers generated from Equation (3.2) can only assume values from the set

i = {0,1 /m, 2/m,..., (m — l)/m), since each *Xi* is an integer in the set {0,1,2,..., m —1}. Thus, each *Ri* is discrete on i, instead of continuous on the interval [0, 1], This approximation appears to be of little consequence, provided that the modulus **m** is a very large integer. (Values such as *m =* 231 — 1 and *m =* 248 are in common use in generators appearing in many simulation languages.) By maximum density is meant that the values assumed by *Ri =* 1,2,..., leave no large gaps on [0,1]

Second, to help achieve maximum density, and to avoid cycling (i.e., recurrence of the same sequence of generated numbers) in practical applications, the generator should have the largest possible period. Maximal period can be achieved by the proper choice of *a,* c, m, and *X0* .

* For m a power of 2, say *m =2b* and *c* ≠ 0, the longest possible period is *P* = *m = 2*b*,* which is achieved provided that **c**is relatively prime to m (that is, the greatest common factor of c and m i s l), and =a = l+4k, where *k* is an integer.
* For m a power of 2, say *m =2b* and *c =* 0, the longest possible period is *P* = *m*⁄*4 = 2b-2* which is achieved provided that the seed X0 is odd and the multiplier, a, is given by =3+8K , for some K=0,1,..
* For m a prime number and c=0, the longest possible period is P=m-1, which is achieved provided that the multiplier, a, has the property that the smallest integer k such that a k-1 is divisible by m is k= m-1.

Example 3.2

Using the multiplicative congruential method, find the period of the generator for a=13, m=26=64, and X0= 1, 2, 3, and 4. The solution is given in table 3.1. When the seed is 1 and 3, the sequence has period 16. However, a period of length eight is achieved when the seed is 2 and a period of length four occurs when the seed is 4.

**In example 3.2,** m=26 =64 and c=0. The maximal period is therefore P=m/4=16. Notice that this period is achieved using odd seeds X0=1 and X0=3, but even seeds X0=2 and X0=4, yield periods of eight and four, both less than maximum. Notice that a=13 is of the form 5+8k with k=1, as required to achieve maximal period.

When X0=1, the generated sequence assumes values from the set (1, 5, 9, 13,…..53,57,61). The “gaps” in the sequence of generated random number Ri, are quite large (i.e. the gap is 5/64-1/64 or 0.0625). Such a gap gives rise to concern about the density of the generated sequence.

Table 3.1 period determination using various seeds

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| i | Xi | Xi | Xi | Xi |
| 0 | 1 | 2 | 3 | 4 |
| 1 | 13 | 26 | 39 | 52 |
| 2 | 41 | 18 | 59 | 36 |
| 3 | 21 | 42 | 63 | 20 |
| 4 | 17 | 34 | 51 | 4 |
| 5 | 29 | 58 | 23 |  |
| 6 | 57 | 50 | 43 |  |
| 7 | 37 | 10 | 47 |  |
| 8 | 33 | 2 | 35 |  |
| 9 | 45 |  | 7 |  |
| 10 | 9 |  | 27 |  |
| 11 | 53 |  | 31 |  |
| 12 | 49 |  | 19 |  |
| 13 | 61 |  | 55 |  |
| 14 | 25 |  | 11 |  |
| 15 | 5 |  | 15 |  |
| 16 | 1 |  | 3 |  |

**EXAMPLE 3.3**

Let *m* = 102 = 100, *a* = 19, *c* = 0, and *X0 =* 63, and generate a sequence c random integers using

Equation (3.1).

*X0 =* 63

*X1* = (19) (63) mod 100 = 1197 mod 100 *=* 97

X2 = (19) (97) mod 100 = 1843 mod 100 = 43

X3 = (19) (43) mod 100 = 817 mod 100 = 17

**.**

**.**

When m is a power of 10, say *m* = 10b*,* the modulo operation is accomplished by saving the *b* rightmost (decimal) digits

**EXAMPLE 3.4**

The values for a, c and m have been selected to ensure that the characteristic desired in a generator are most likely to be achieved. By changing X0, the user can control the repeatability of the stream.

Let a = 75 = 16,807, *m* = 231-1 = 2,147,483,647 (a prime number), and c= 0. These choices satisfy the conditions that insure a period of **P = m— 1**. Further, specify a seed, *XQ =* 123,457.

The first few numbers generated are as follows:

X1= 75(123,457) mod (231 - 1) = 2,074,941,799 mod (231 - 1)

*X1 =* 2,074,941,799

R1*=* X1 ⁄231

*X2 =* 75(2,074,941,799) mod (231 - 1) = 559,872,160

*R2 =* X2 ⁄231= 0.2607

X3 = 75(559,872,160) mod (231 - 1) = 1,645,535,613

*R3 =* X3 ⁄231= 0.7662

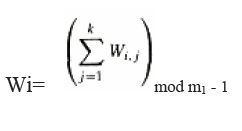
Notice that this routine divides by *m* + 1 instead of *m* ; however, for such a large value of *m* ,the effect is negligible.

**Combined Linear Congruential Generators**

As computing power has increased, the complexity of the systems that we are able to simulate has also increased.

One fruitful approach is to combine two or more multiplicative congruential generators in such a way that the combined generator has good statistical properties and a longer period. The following result from L'Ecuyer [1988] suggests how this can be done:

If Wi,1 , Wi , 2. . . , *W i,k* are any independent, discrete-valued random variables (not necessarily identically distributed), but one of them, say Wi, 1, is uniformly distributed on the integers 0 to mi — 2, then



is uniformly distributed on the integers 0 to m1-2.

To see how this result can be used to form combined generators, let Xi,1, X i,2,..., X i,k be the i th output from *k* different multiplicative congruential generators, where the *j* th generator has prime modulus ***mj*,** and the multiplier ***aj***is chosen so that the period is **m*j —* 1*.***Then the j'th generator is producing integers *Xi,j* that are approximately uniformly distributed on 1 to *mj -* 1, and *Wi,j = X i,j* — 1 is approximately uniformly distributed on 0 to m*j - 2.* L'Ecuyer [1988] therefore suggests combined generators of the form

k

Xi = Σ (-1) j-1 *X i,j* mod m1 – *1*

j=1

with

R i = Xi/m1, Xi > 0

m1 -1 ⁄m1 Xi=0

Notice that the " (-1)j-1 "coefficient implicitly performs the subtraction X i,1-1; for example, if *k =* 2, then (-1)°(X *i 1* - 1) - *( - l ) l ( X i 2* - 1)=Σ2

j=1( -1)

j-1 *X i,j*

The maximum possible period for such a generator is

P= (m1 -1)(m2 - l ) - - - (mk - 1)

2 k-1

which is achieved by the following generator:

**EXAMPLE 3.5**

For 32-bit computers, L'Ecuyer [1988] suggests combining *k = 2* generators with m1 =

2147483563, *a*1*=* 40014, *m2 =* 2147483399, and *a2 =*40692. This leads to the following algorithm:

1. Select seed X1, 0 in the range [1, 2147483562] for the first generator, and seed X2.0 in the range [1, 2147483398]. Set j=0.
2. Evaluate each individual generator.

X 1, j+1 = 40014X 1, j mod 2147483563

X2, j+i = 40692X2, j mod 2147483399

1. Set

Xj+1 = *(X* 1, j+1 - X 2 j+1) mod 2147483562

1. Return

R j+1 = Xj+1 ⁄ 2147483563, X j+1 > 0

2147483563/2147483563. X j +1= 0

1. . Set *j = j +* 1 and go to step 2.

This combined generator has period (m1-1)(m2-1)/2≈2\*1018

**Tests for Random Numbers**

The desirable properties of random numbers are uniformity and independence. To insure that these desirable properties are achieved, a number of tests can be performed (fortunately, the appropriate tests have already been conducted for most commercial simulation software). The tests can be placed in two categories according to the properties of interest. The first entry in the list below concerns testing for uniformity. The second through fifth entries concern testing for independence. The five types of tests

1. **Frequency test: -** Uses the Kolmogorov-Smirnov or the chi- square test to compare the distribution of the set of numbers generated to a uniform distribution.

2. **Runs test:-**. Tests the runs up and down or the runs above, and below the mean by comparing the actual values to expected values. The statistic for comparison is the chi-square.

3*.* **Autocorrelation test: -** Tests the correlation between numbers and compares the sample correlation to the expected correlation of zero.

4. **Gap test:-***.* Counts the number of digits that appear between repetitions of particular digit and then uses the Kolmogorov-Smirnov test to compare with the expected size of gaps,

5 **Poker test: -**Treats numbers grouped together as a poker hand. Then the hands obtained are compared to what is expected using the chi-square test.

The null hypothesis, H**0** reads that the numbers are distributed uniformly on the interval (0,1). Failure to reject the null hypothesis means that no evidence of non-uniformity has been detected on the basis of this test. This does not imply that further testing of the generator for uniformity is unnecessary.

In testing for independence, the hypotheses are as follows:

H**0**: R**i** ~ independently

H**1**:R**i**~independently

This null hypothesis H**0** reads that the numbers are independent. Failure to reject the null hypothesis means that no evidence of dependence has been detected on the basis of this test. This does not imply that further testing of the generator for independence is unnecessary.

For each test, a level of significance a must be stated. The level a is the probability of rejecting the null hypothesis given that the null hypothesis is true, or

a = P (reject H**0** |H**0** true)

The decision maker sets the value of & for any test. Frequently, ***a*** is set to 0.01 or 0.05. If several tests are conducted on the same set of numbers, the probability of rejecting the null hypothesis on at least one test, by chance alone (i.e., making a Type I (a) error), increases. Say that a= 0.05 and that five different tests are conducted on a sequence of numbers. The probability of rejecting the null hypothesis on at least one test, by chance alone, may be as large as 0.25.

**Frequency Tests**

A basic test that should always be performed to validate a new generator is the test of uniformity. Two different methods of testing are available. They are the Kolmogorov-Smirnov and the chi-square test. Both of these tests measure the degree of agreement between the distribution of a sample of generated random numbers and the theoretical uniform distribution. Both tests are based on the null hypothesis of no significant difference between the sample distribution and the theoretical distribution.

The **Kolmogorov-Smirnov test: -** This test compares the continuous cdf, F(X), of the uniform distribution to the empirical cdf, S**N**(x), of the sample of N observations. By definition,

F(x) = x, 0 <= x <= 1

If the sample from the random-number generator is R1 R2, ,• • •, R**N**, then the empirical cdf (*cumulative distribution function)*, S**N**(X), is defined by

S**N**(X) = number of R1 R2, ,• • •, Rn which are <= x

N

As N becomes larger, S**N**(X) should become a better approximation to F(X), provided that the null hypothesis is true.

The Kolmogorov-Smirnov test is based on the largest absolute deviation between F(x) and S**N**(X) over the range of the random variable. That is. it is based on the statistic

D = max | F(x) - S**N**(x)| ………………………………. 3**.3**

For testing against a uniform cdf, the test procedure follows these steps:

**Step-1:** Rank the data from smallest to largest. Let R (i) denote the ***i*th** smallest observation, so that

R (1) <= R (2) <= • • • <= R (N)

**Step-2:** Compute D+ = max \_\_i\_ - R(i)

1<=i<=N N

D- = max R(i) - \_i - 1\_

1<=i<=N N

**Step-3:** Compute D = max (D+, D-).

**Step-4:** Determine the critical value, Dα, Let a=0.05, (The critical value D is given as D**0.05** =1.36/√100= 0.136, since D=max |F(x)-S**N**(x)| =0.0224 is less than D**0.05** do not reject the hypothesis of independence Table-3.1)for the specified significance level **α and** the given sample size **N**.

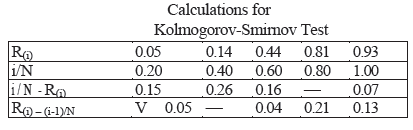
**Step-5:** If the sample statistic D is greater than the critical value, Dα, the null hypothesis that the data are a sample from a uniform distribution is rejected.

If D <= Dα, conclude that no difference has been detected between the true distribution of { R1, R2, • • •, Rn } and the uniform distribution.

**EXAMPLE 3.4**

Suppose that the five numbers 0.44, 0.81, 0.14, 0.05, 0.93 were generated, and it is desired to perform a test for uniformity using the Kolmogorov-Smirnov test with a level of significance a of 0.05.

First, the numbers must be ranked from smallest to largest. The calculations can be facilitated by use of Table 3.2. The top row lists the numbers from smallest (R(1) ) to largest (R(n) ) .The computations for D+, namely i /N -R(i} and for D-, namely R(i ) - ( i - l ) / N, are easily accomplished using Table 3.2. The statistics are computed as D+ = 0.26 and D- = 0.21. Therefore, D = max {0.26, 0.21} = 0.26. The critical value of **D,** obtained from Table A.8 for a = 0.05 and N =5, is 0.565. Since the computed value, 0.26, is less than the tabulated critical value, 0.565, the hypothesis of no difference between the distribution of the generated numbers and the uniform distribution is not rejected.



The calculations in the above table are illustrated in Figure 3.2, where the empirical cdf, SN(X), is compared to the uniform cdf, F(x). It can be seen that D+ is the largest deviation of SN(x) above F(x), and that D- is the largest deviation of SN(X) below F(x). For example, at R(3) the value of D+ is given by 3/5 - R(3) = 0.60 - 0.44 =0.16 and of D- is given by R(3) = 2/5 = 0.44 - 0.40 = 0.04. Although the test statistic D is defined by Equation (3.3) as the maximum deviation over all x, it can be seen from

Figure 3.1 that the maximum deviation will always occur at one of the jump points R(1) , R(2) . . . , and thus the deviation at other values of x need not be considered.

**The chi-square test**

The chi-square test uses the sample statistic.



Where Oi; is the observed number in the *i*th class, *Ei* is the expected number in the ith class, and *n* is the number of classes. For the uniform distribution, *Ei* the expected number in each class is given by

Ei = N/n

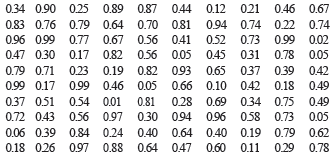
For equally spaced classes, where *N* is the total number of observations. It can be shown that the sampling distribution of X02 is approximately the chi-square distribution with *n -* 1 degrees of freedom

**EXAMPLE 3.6**

Use the chi-square test with *a =* 0.05 to test whether the data shown below are uniformly distributed. Table .3 contains the essential computations. The test uses n = 10 intervals of equal length, namely

[0, 0.1), [0.1, 0.2), . . . , [0.9, 1.0). The value of X02 is 3.4. This is compared with the critical value

X20.05,9 =16.9.Since X02 is much smaller than the tabulated value of X2 0.05,9 , the null hypothesis of a uniform distribution is not rejected.



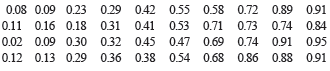
Both the Kolmogorov-Smirnov and the chi-square test are acceptable for testing the uniformity of a sample of data, provided that the sample size is large. However, the Kolmogorov-Smirnov test is the more powerful of the two and is recommended. Furthermore, the Kolmogorov-Smirnov test can be applied to small sample sizes, whereas the chi-square is valid only for large samples, say N>=50.

Imagine a set of 100 numbers which are being tested for independence where the first 10 values are in the range 0.01-0.10, the second 10 values are in the range 0.11-0.20, and so on. This set of numbers would pass the frequency tests with ease, but the ordering of the numbers produced by the generator would not be random. The tests in the remainder of this chapter are concerned with the independence of random numbers which are generated. The presentation of the tests is similar to that by Schmidt and Taylor [1970].

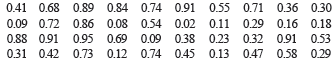
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Interval** | **Oi** | **Ei** | **Oi -Ei** | **( Oi -Ei )2** | **( Oi -Ei )2 / Ei** |
| 1 | 08 | 10 | -2 | 4 | 0.4 |
| 2 | 08 | 10 | -2 | 4 | 0.4 |
| 3 | 10 | 10 | 0 | 0 | 0.0 |
| 4 | 09 | 10 | -1 | 1 | 0.1 |
| 5 | 12 | 10 | 2 | 4 | 0.4 |
| 6 | 08 | 10 | -2 | 4 | 0.4 |
| 7 | 10 | 10 | 0 | 0 | 0.0 |
| 8 | 14 | 10 | 4 | 16 | 1.6 |
| 9 | 10 | 10 | 0 | 0 | 0.0 |
| 10 | 11 | 10 | 1 | 1 | 0.1 |

***Runs Tests***

**Runs up and runs down***.* Consider a generator that provided a set of 40 numbers in the following sequence:



Both the Kolmogorov-Smirnov test and the chi-square test would indicate that the numbers are uniformly distributed. However, a glance at the ordering shows that the numbers are successively larger in blocks of 10 values. If these numbers are rearranged as follows, there is far less reason to doubt their independence



The runs test examines the arrangement of numbers in a sequence to test the hypothesis of independence.

Before defining a run, a look at a sequence of coin tosses will help with some terminology. Consider the following sequence generated by tossing a coin 10 times:

***H T T H H T T T H T***

There are three mutually exclusive outcomes, or events, with respect to the sequence. Two of the possibilities are rather obvious. That is, the toss can result in a head or a tail. The third possibility is "no event." The first head is preceded by no event and the last tail is succeeded by no event. Every sequence begins and ends with no event.

A run is defined as a succession of similar events preceded and followed by a different event. The length of the run is the number of events that occur in the run. In the coin-flipping example above there are six runs. The first run is of length one, the second and third of length two, the fourth of length three. and the fifth and sixth of length one.

There are two possible concerns in a runs test for a sequence of number. The number of runs is the first concern and the length of runs is a second concern. The types of runs counted in the first case might be runs up and runs down. An up run is a sequence of numbers each of which is succeeded by a larger number. Similarly, a down run is a sequence of numbers each of which is succeeded by a smaller number. To illustrate the concept, consider the following sequence of 15 numbers:

-0.87 +0.15 +0.23 +0.45 -0.69 -0.32 -0.30 +0.19 -.24 +0.18 +0.65 +0.82 -0.93 +0.22 0.81

The numbers are given a “+” or a “— “depending on whether they are followed by a larger number or a smaller number. Since there are 15 numbers, and they are all different, there will be 14 +'s and —’s. The last number is followed by "no event" and hence will get neither a + nor a —. The sequence of 14 +s and —’s as follows:

- + + + - - - + - + + - +

Each succession of +’s and —’s forms a run. There are eight runs. The first run is of length one. The second and third are of length three, and so on. Further, there are four runs up and four runs down.

There can be too few runs or too many runs. Consider the following sequence of numbers:

0.08 0.18 0.23 0.36 0.42 0.55 0.63 0.72 0.89 0.91

This sequence has one run, a run up. It is unlikely that a valid random-number generator would produce such a sequence. Next, consider the following sequence

0,08 0.93 0.15 0.96 0.26 0.84 0.28 0.79 0.36 0.57

This sequence has nine runs, five up and four down. It is unlikely that a sequence of l0 numbers would have this many runs. What is more likely is that the number of runs will be somewhere between the two extremes. These two extremes can be formalized as follows: if *N* is the number of numbers in a sequence, the maximum number of runs is N —1and the minimum number of runs is one.

If *a* is the total number of runs in a truly random sequence, the mean and variance of *a* are given by

μa = 2N – 1 / 3

and

σ2a = 16N – 29 / 90

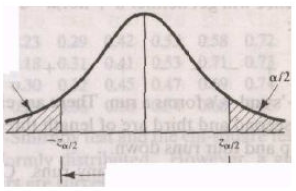
For *N >* 20, the distribution of *a* is reasonably approximated by a normal distribution, N ( μa , σ2 a) This approximation can be used to test the independence of numbers from a generator. In that case the standardized normal test statistic is developed by subtracting the mean from the observed number of runs, *a*, and dividing by the standard deviation. That is, the test statistic is

**Z0 = *a -* μa / σa**

Substituting Equation (3,4) for μa and the square root of Equation (3.5) for σa yields

**Z0 = a - [( 2N - 1)/3] / sqrt((16N – 29) / 90 )**

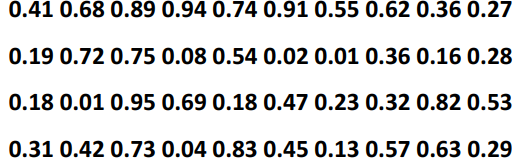
Where Z0 ~ *N(0* 1). Failures to reject the hypothesis of independence occur when — *za / 2* <=

Z0 <= *za / 2* where *a* is the level of significance. The critical values and rejection region are shown in Figure 3.3.

Failure to reject Figure 3.3

**Example 3.8**

Based on run up and run down determine whether the following sequence of 40 numbers is such that the hypothesis of independence can be rejected where σ= 0.05



The sequence of runs up and down is as follows

+ + + - + - + - - - + + - + - - + - + - - + - - + - + + - - + + - + - - + + -

There are 26 run in this sequence. N = 40 and a = 26 so

μa = 2(40) – 1 / 3 = 26.33

and

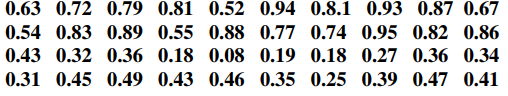
σ2a = 16(40) – 29 / 90 = 6.79

Then

**Z0 = *a -* μa / σa. Z0 = 26 *– 26.33* / sqrt 6.79 = -13**

Now, the critical value is z0 0.025=1.96, so the independence of the numbers cannot be rejected on the base of this test.

1. **Runs above and below the mean**. The test for runs up and runs down is not completely adequate to assess the independence of a group of numbers. Consider the following 40 numbers:



The sequence of runs up and runs down is as follows:

+ + + - + - + - - - + + - + - - + - + - - + - - + - + + - - + + - + - - + + -

This sequence is exactly the same as that in Example 3.8. Thus, the numbers would pass the runs-up and runs-down test. However, it can be observed that the first 20 numbers are all above the mean [(0.99 + O.OO)/2 = 0.495] and the last 20 numbers are all below the mean. Such an occurrence is highly unlikely. The previous runs analysis can be used to test for this condition, if the definition of a run is changed. Runs will be described as being above the mean or below the mean. A " + " sign will be used to denote an observation above the mean, and a "-" sign will denote an observation below the mean.

For example, consider the following sequence of 20 -digit random numbers;

**0.40 0.84 0.75 0.18 0.13 0.92 0.57 0.77 0.30 0.71**

**0.42 0.05 0.78 0.74 0.68 0.03 0.18 0.51 0.10 0.37**

The pluses and minuses are as follows:

- + + - - + + + - + - - + + + - - + - -

In this case, there is a run of length one below the mean followed by a run of length two above the mean, and so on. In all there are 11 runs, five of which are above the mean and six of which are below the mean. Let **n1** and **n2** be the number of individual observations above and below the mean and let **b** is the total number of runs. Notice that the maximum number of runs is N = n1 + n2 and the minimum number of runs is one.

Given **n1** and **n2** the mean with a continuity correction suggested by Swed and Eisenhart [1943] —and the variance of b for a truly independent sequence are given by

µb =2 n1 n2 / N +1/2 (3.6)

and

α b2 = 2n1n2 (2n1n2-N) / N2 (N – 1) (3.7)

For either **n1** or **n2** greater than 20, b is approximately normally distributed. The test statistic can be formed by subtracting the mean from the number of runs and dividing by the standard deviation, or

Z0 = (b-(2n1n2/N)-1/2) / [2n1n2 (2n1n2-N/N2 (N-1)] ½

Failure to reject the hypothesis of independence occurs when —za/2 <= Z0 < =za/2-, where a is the level of significance

EXAMPLE 4.9

Determine whether there are an excessive number of runs above or below the mean for the sequence of numbers given in Example 3.8. The assignment of + 's and — 's results in the following:

- + + + + + + + - - - + + - + - - - - - - - + + - - - - + + - - + - + - - + + -

The values of n1, n2, and b are as follows:

n1= 18

n2= 22

N = n1 +n2 = 40

b = 17

Equations (3.6) and (3.7) are used to determine b and b2 as follows:

µb = 2(18) (22)/40 + 1/2 = 20.3

α b2 = 2(18) (22) [(2) (18) (22)-40] / (40)2 (40-1) =9.54

Since n2 is greater than 20, the normal approximation is acceptable, resulting in a Z0 value of

Z0 = 17-20.3/ 9.54 = -1.07

Since Z0.025 = 1.96 the hypothesis of independence cannot be rejected on the basis of this test

1. **Runs test:** length of runs. Yet another concern is the length of runs. As an example of what might occur, consider the following sequence of numbers

**0.16, 0.27, 0.58, 0.63, 0.45, 0.21, 0.72, 0.87, 0.27, 0.15, 0.92, 0.85**

Assume that this sequence continues in a like fashion: two numbers below the mean followed by two numbers above the mean. A test of runs above and below the mean would detect no departure from independence. However, it is to be expected that runs other than of length two should occur.

Let Y, be the number of runs of length i in a sequence of N numbers. For an independent sequence, the expected value of Yj for runs up and down is given by

E(Yi) = 2/(i+3)![ N(i2+3i+1)-(i3+3i2-i-4) ] ,i <= N – 2 (3.8)

E(Yi) =2/N! i = N – 1 (3.9)

For runs above and below the mean, the expected value of y, is approximately given by

E(Yi) = N wi / E(I ), N>20 (3.10)

Where wi the approximate probability that a run has length i, is given by

wi=( n1 /N)I (n2/N) + (n1/N)(n2/N)I N>20 (3.11)

and where E(I ) , the approximate expected length of a run, is given by

E(I )= n1 / n2 + n2/ n1 N>20 (3.12)

The approximate expected total number of runs (of all lengths) in a sequence of length N, E(A), is given by

E(A)=N/ E(I) N>20 (3.13)

The appropriate test is the chi-square test with Oi, being the observed number of runs of length i.

Then the test statistic is

X02= i = 1 L [ Oi - E(Yi)2 ] / E(Yi)

Where L = N - 1 for runs up and down and L = N for runs above and below the mean. If the null hypothesis of independence is true, then X02 is approximately chi-square distributed with L — 1 degrees of freedom.

EXAMPLE 4.10

Given the following sequence of numbers, can the hypothesis that the numbers are independent be rejected on the basis of the length of runs up and down at a =0.05?

0.30 0.48 0.36 0.01 0.54 0.34 0.96 0.06 0.61 0.85 0.48 0.86 0.14 0.86 0.89 0.37 0.49 0.60 0.04 0.83

0.42 0.83 0.37 0.21 0.90 0.89 0.91 0.79 0.57 0.99 0.95 0.27 0.41 0.81 0.96 0.31 0.09 0.06 0.23 0.77

0.73 0.47 0.13 0.55 0.11 0.75 0.36 0.25 0.23 0.72 0.60 0.84 0.70 0.30 0.26 0.38 0.05 0.19 0.73 0.44

For this sequence the +'s and —'s are as follows

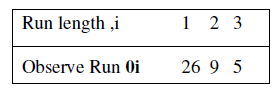
**+ - - + - + - + + - + - + + - + + - + - + - - + - + - - + - - + + + - - - + + - - -**

**+ - + - - - + - + - - - + - + + -**

The length of runs in the sequence is as follows:

1,2,1,1,1,1,2,1,1,1,2,1,2,1,1,1,1,2,1,1, 1,2,1,2,3,3,2,3,1,1,1,3,1,1,1, 3,1,1,2,1

The number of observed runs of each length is as follows:



The expected numbers of runs of lengths one, two, and three are computed from Equation µ(3.8) as

E(Yi) = 2/4![60(1 + 3 + 1) - (1 + 3 - 1 - 4)]= 25.08

E(Y2) = 2/5![60(4 + 6 + 1) - (8 + 12 - 2- 4)] = 10.77

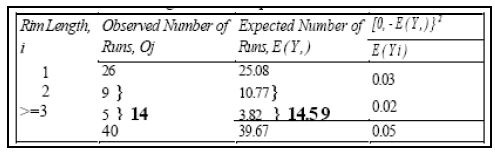
E(Y3) =2/6![60(9 + 9 + 1) – (27 + 27 - 3 – 4)] =3.04

The mean total number of runs (up and down) is given by Equation (3.4) as

µ a = 2(60)-1/3=39.67

Thus far, the E(Yi) for i = 1, 2, and 3 total 38.89. The expected number of runs of length 4 or more is the difference i=1 E(Yi) or 0.78

Table 3.4. Length of Runs Up and Down: x2 Test



The tests for autocorrelation are concerned with the dependence between numbers in a sequence.

As an example, consider the following sequence of numbers:

**0.12 0.01 0.23 0.28 0.89 0.31 0.64 0.28 0.83 0.93**

**0.99 0.15 0.33 0.35 0.91 0.41 0.60 0.27 0.75 0.88**

**0.68 0.49 0.05 0.43 0.95 0.58 0.19 0.36 0.69 0.87**

From a visual inspection, these numbers appear random, and they would probably pass all the tests presented to this point. However, an examination of the 5th, 10th, 15th (every five numbers beginning with the fifth), and so on. indicates a very large number in that position. Now, 30 numbers is a rather small sample size to reject a random-number generator, but the notion is that numbers in the sequence might be related. In this particular section, a method for determining whether such a relationship exists is described. The relationship would not have to be all high numbers. It is possible to have all low numbers in the locations being examined, or the numbers may alternately shift from very high to very low.

The test to be described below requires the computation of the autocorrelation between every m numbers (m is also known as the lag) starting with the ith number. Thus, the autocorrelation p, m between the following numbers would be of interest: /?,-, Rj+m, Ri+2m, • •

Ri+(M+\)m- The value M is the largest integer such that / + (M + l)m < N, where N is the total number of values in the sequence. (Thus, a subsequence of length M + 2 is being tested.) Since a nonzero autocorrelation implies a lack of independence, the following two tailed test is appropriate:



For large values of M, the distribution of the estimator of im denoted ^im is approximately normal if the values Ri, Ri+m, Ri+2m,…….Ri+(M+1)m are un-correlated. Then the test statistic can be formed as follows:



Which is distributed normally with a mean of zero and a variance of 1, under the assumption of independence, for large M.

The formula for ^im in a slightly different form, and the standard deviation of the estimatorα

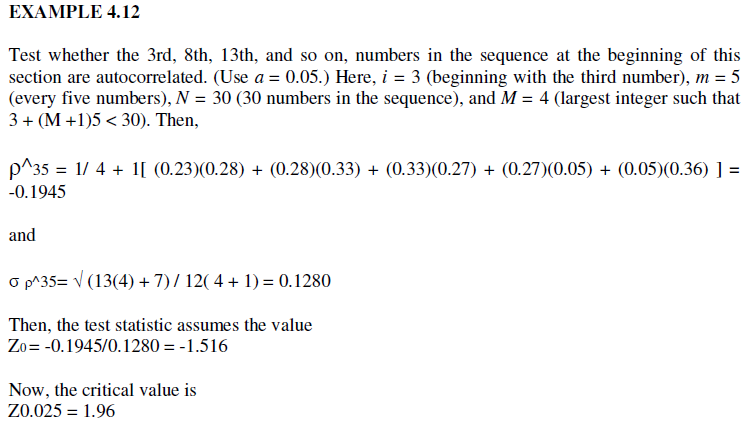
^im are given by Schmidt and Taylor [1970] as follows:





After computing Z0, do not reject the null hypothesis of independence if - za/2 <= Z0 <= za/2, where α is the level of significance.

If im > 0, the subsequence is said to exhibit positive autocorrelation. In this case, successive values at lag m have a higher probability than expected of being close in value (i.e., high random numbers in the subsequence followed by high, and low followed by low). On the other hand, if im < 0, the subsequence is exhibiting negative autocorrelation, which means that low random numbers tend to be followed by high ones, and vice versa. The desired property of independence, which implies zero autocorrelation, means that there is no discernible relationship of the nature discussed here between successive, random numbers at lag m.



Therefore, the hypothesis of independence cannot be rejected on the basis of this test. It can be observed that this test is not very sensitive for small values of M, particularly when the numbers being tested are on the low side. Imagine what would happen if each of the entries in the foregoing computation of^im were equal to zero. Then, ^im would be equal to —0.25 and the calculate would have the value of —1.95, not quite enough to reject the hypothesis of independence.

Many sequences can be formed in a set of data, given a large value of N. For example, beginning with the first number in the sequence, possibilities include

1. The sequence of all numbers,
2. The sequence formed from the first. third, fifth,..., numbers,
3. The sequence formed from the first, fourth, numbers, and so on. If α — 0.05, there is a

Probability of 0.05 of rejecting a true hypothesis. If 10 independent sequences are examined, the probability of finding no significant autocorrelation, by chance alone, is (0.95)10 or 0.60. Thus, 40% of the time significant autocorrelation would be detected when it does not exist. If α is 0.10 and 10 tests are conducted, there is a 65% chance of finding autocorrelation by chance alone. In conclusion, when fishing" for autocorrelation, upon performing numerous tests, autocorrelation may eventually be detected, perhaps by chance alone, even when no autocorrelation is present.

* + 1. Gap Test

The gap test is used to determine the significance of the interval between the recurrences of the same digit. A gap of length x occurs between the recurrences of some specified digit. The following example illustrates the length of gaps associated with the digit 3:

**4, 1, 3, 5, 1, 7. 2. 8. 2. 0, 7. 9. 1. 3. 5, 2, 7, 9. 4. 1. 6. 3**

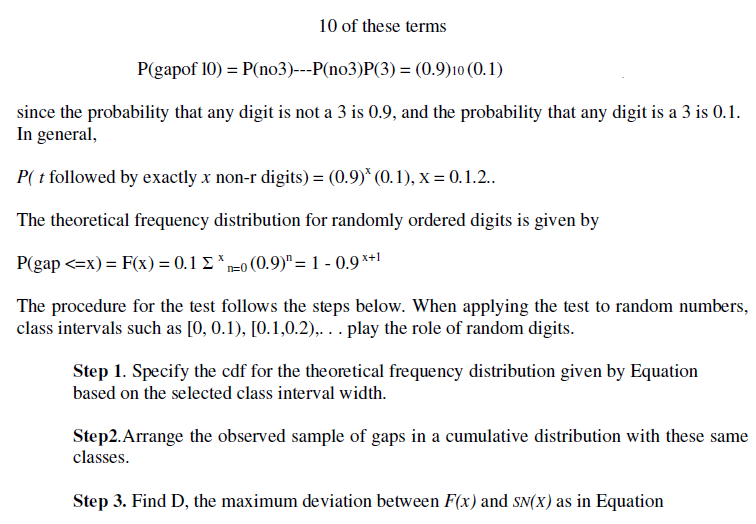
**3, 9. 6, 3, 4. 8, 2. 3, 1, 9, 4. 4, 6. 8. 4, 1, 3. 8. 9. 5. 5. 7**

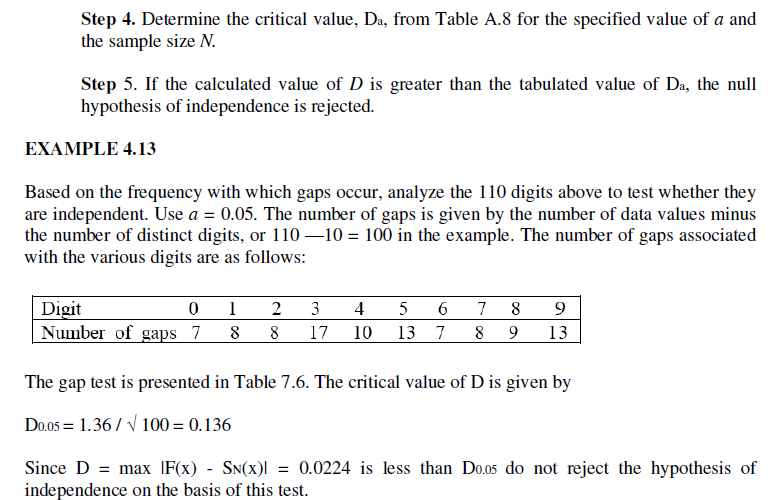
**3, 9, 5, 9. 8, 5. 3. 2, 2, 3, 7. 4, 7. 0. 3. 6. 3, 5, 9. 9. 5. 5**

**5, 0, 4. 6. 8. 0, 4. 7. 0, 3. 3, 0, 9, 5. 7, 9. 5. 1. 6. 6. 3. 8**

**8, 8, 9, 2, 9. 1. 8. 5, 4. 4. 5, 0, 2. 3, 9, 7. 1. 2. 0. 3, 6. 3**

To facilitate the analysis, the digit 3 has been underlined. There are eighteen 3's in the list. Thus, only 17 gaps can occur. The first gap is of length 10. the second gap is of length 7, and so on. The frequency of the gaps is of interest. The probability of the first gap is determined as follows:





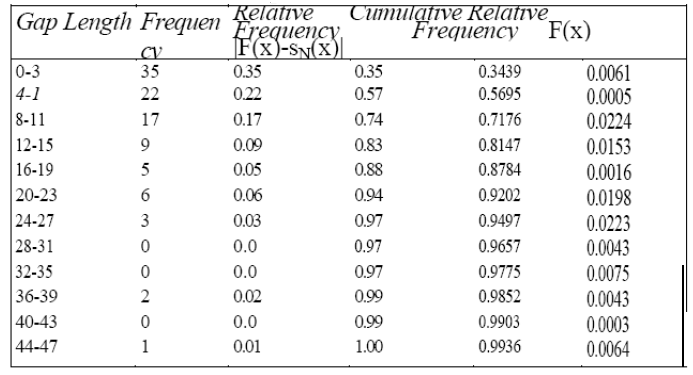


Table 3.6 Gap-Test Example

**3.4.5 Poker Test**

