0405324: Stochastic System Simulation

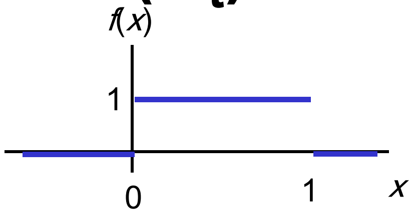
Lecture 6: Generating random numbers and variates; Designing and executing simulation experiments

Content

- **Random-number generation**
- **Generating random variates**
- **Designing and executing simulation experiments**

Random-Number Generators (RNGs)

- **Algorithm to generate independent, identically distributed (iid) random number draws (R_t) from continuous UNIF (0, 1) distribution**
  - These are called *random numbers* in simulation



- **Basis for generating observations from all other distributions and random processes**
  - Transform random numbers in a way that depends on desired distribution or process (variates: later in this chapter)
- **It's essential to have a good RNG (statistically)**
- **There is a lot of *bad* RNGs — this is *very* tricky**
  - Methods and coding are both tricky

Nature of Random Number Generator (RNGs)

- **Recursive formula (algorithm)**
  - Starts with a seed (or seed vector)
  - Does something *weird* to the seed to get the next one
  - Repeats … generates same sequence (cycle) → pseudo-random number (known in advance)
- **Want to "design" RNGs**
  - Long cycle length (much more than the required number of events in a simulation)
  - Good statistical properties: imitate the ideal properties of random numbers (uniformity, independence) -- tests
  - Fast (computationally efficient in terms of time)
  - Streams – repeatable sequences given a seed (starting point) – they should be many and long (for variance reduction … later)
- **This is hard! Doing something *weird* isn't enough**

Techniques for Generating Random Number

- **Middle-square method**--is a method of generating pseudo-randomized numbers. In practice it is not a good method, since its period is usually very short and it has some severe weaknesses, such as the output sequence almost always converges to zero.
- **Linear Congruential Generator (LCG)**--is an algorithm that yields a sequence of pseudo-randomized numbers that are calculated with a discontinuous piecewise linear equation.

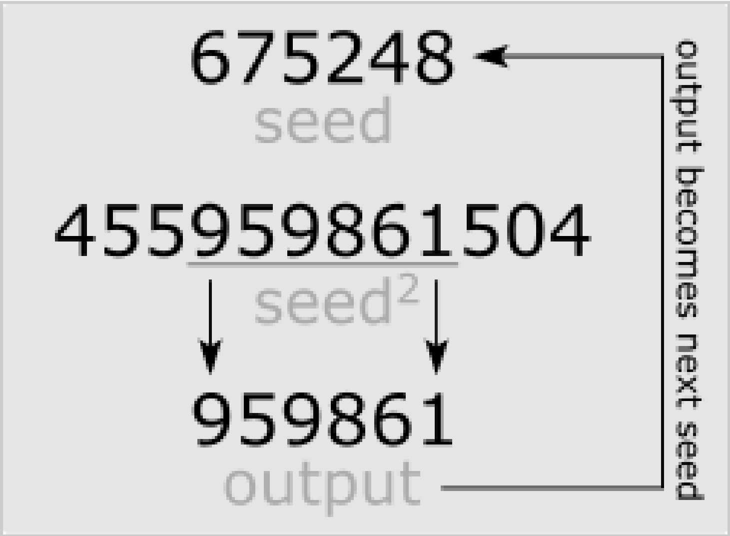
Techniques for Generating Random Number—Middle-square method

To generate a sequence of 4-digit pseudorandom numbers:

- Generally a 4-digit starting value is created
- Square and produce an 8-digit number (if the result is less than 8 digits, leading zeros are added to compensate).
- The middle 4 digits of the result would be the next number in the sequence, and returned as the result. This process is then repeated to generate more numbers.
- For a generator of n-digit numbers, the period can be no longer than 8^n. If the middle 4 digits are all zeroes, the generator then outputs zeroes forever. If the first half of a number in the sequence is zeroes, the subsequent numbers will be decreasing to zero
- In practice it is not a good method, since its period is usually very short and it has some severe weaknesses, such as the output sequence almost always converges to zero.

Techniques for Generating Random Number—Middle-square method

Example-



One iteration of the middle-square method, showing a six digit seed, which is then squared, and the resulting value has its middle six digits as the output value (and also as the next seed for the sequence).

$X_0 = 675248$ (seed)

$X_0^2 = 455\underline{\mathbf{959861}}504$

$\Rightarrow R_1 = 0.959861$

$X_0^2 = (959861)^2 = 921\underline{\mathbf{333139}}300$

$\Rightarrow R_2 = 0.333139$

…

Techniques for Generating Random Number—Middle-square method

As mentioned before, a seed does not guarantee that the sequence will not degenerate and will have a long period. Also zeros, once they appear, are carried in subsequent numbers.

Ex: $X_0 = 5197$ (seed) $X_0^2 = 27\underline{\mathbf{0088}}09$

=> $R_1 = 0.0088$ $X_1^2 = 00\underline{\mathbf{0077}}44$

=> $R_2 = \mathbf{0.0077}$

Ex: $X_0 = 4500$ (seed) $X_0^2 = 20\underline{\mathbf{2500}}00$

==> $R_1 = \mathbf{0.2500}$ $X_1^2 = 06\underline{\mathbf{2500}}00$

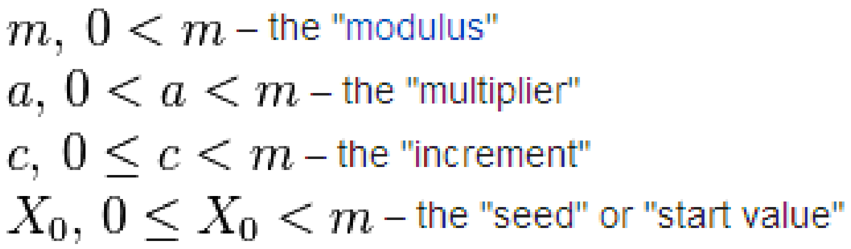
==> $R_2 = \mathbf{0.2500}$

Techniques for Generating Random Number—Linear Congruential Generator (LCG)

- Produces a sequence of integers X_1, X_2, X_3, …between 0 and (m - 1), via recursion

$$X_i = (a\, X_{i-1} + c) \pmod{m},\ i = 0, 1, 2,\ldots$$

- a, c, and m are carefully chosen constants



$m,\ 0 < m$ – the "modulus"

$a,\ 0 < a < m$ – the "multiplier"

$c,\ 0 \le c < m$ – the "increment"

$X_0,\ 0 \le X_0 < m$ – the "seed" or "start value"

- Specify a seed X_0 to start off
- "mod m" means take remainder of dividing by m as next X_i
- All X_i's are between 0 and m – 1
- Return ith "random number" as $R_i = \frac{X_i}{m}$

Techniques for Generating Random Number— Linear Congruential Generator (LCG)

- **Example-** Use the linear congruential method to generate a sequence of four degit random numbers with $m = 63$, $a = 22$, $c = 4$, $X_0 = 19$.

Solution: we get,

$$X_i = (22*X_{i-1} + 4) \pmod{63}, \text{ seed with } X_0 = 19$$

i	$22*X_{i-1}+4$	X_i	$R_i = Xi/m$
0		19	
1	422	44	0.6984
2	972	27	0.4286
3	598	31	0.4921
4	686	56	0.8889
:	::	:	
61	158	32	0.5079
62	708	15	0.2381
63	334	19	0.3016
64	422	44	0.6984
65	972	27	0.4286
66	598	31	0.4921
:	::	:	

- *Cycling* — will repeat forever
- Cycle length ≤ m (could be << m depending on parameters)
- Pick m ***BIG***
- But that might not be enough for good statistical properties

Issues with Linear Congruential Generator (LCG)

- **Cycle length: $\leq m$**
  - Typically, $m = 2.1$ billion ($= 2^{31} - 1$) or more
  - Other parameters chosen so that cycle length = m or $m - 1$
  - In the original LCG of Arena (starting in the 80th) was with m= $2^{31} - 1$, a=7^5=16807, and c=0 with a cycle length of 2.1 billion.
  - The current Arena RNG is based on LCG but based on two component generators that are then combined.
- **Statistical properties**
  - Uniformity, independence
  - There are many tests of RNGs
    - Empirical tests
    - Theoretical tests
- **Must be *carefully*, *cleverly* coded — *BIG* integers**
- ***Reproducibility* — streams (long internal subsequences) with fixed seeds**
  - $X_0, \underbrace{X_1, X_2 \ldots X_P, X_0}_{\text{Stream with seed } X_1}, X_1, \ldots.$

It is convenient that the seeds are far apart in the sequence $X_0, X_1, X_2 \ldots X_P$

Test for Random Number Generator (RNG)

- **When a random number generator is devised, one needs to test its property. The two properties we are concerned most are uniformity and independence. A list of tests are used. The first one tests for uniformity and the second to fifth ones test independence.**
  - **Frequency test** → **for uniformity**
  - Runs test
  - **Autocorrelation test**
  - Gap test
  - Poker test

Runs test, Autocorrelation test, Gap test, Poker test → **Independence**

Test for Random Number Generator (RNG)

1. **Frequency test (for uniformity).** Uses the *Kolmogorov-Smirnov (K-S)* test or the *chi-square* test to compare the distribution of the set of numbers generated to *a uniform distribution.*

Hypothesis:

$H_0: R_i \sim U[0,1]$

$H_1: R_i \neq U[0,1]$

H_0: the numbers are distributed uniformly on the interval [0,1].

2. **Autocorrelation test (for independence).** Tests the *correlation* between numbers and compares the sample correlation to the expected correlation of zero.

Hypothesis:

$H_0: R_i \sim$ independent

$H_1: R_i \neq$ independent

H_0: the numbers are independently distributed.

Test for Random Number Generator (RNG)—Uniformity– frequency test

- **Kolmogorov-Smirnov (K-S) test**—This test compares the continuous cdf, $F_T(x)$ of the uniform distribution, U(0, 1) with empirical cdf, $F_S(x)$ of the sample of N observations. By definition,

$$F_T(x) = \frac{x-a}{b-a} = \frac{x-0}{1-0} = x, \quad 0 \le x \le 1$$

- If the sample from the RNG is $R_1, R_2, \ldots, R_N$, then the empirical cdf, $F_S(X)$ is defined by,

$$F_S(x) = \frac{\#(i : x_i \le x)}{N} = \frac{Cumulative\ Frequency}{N}$$

- **Test statistic**--The difference between $F_T(x)$ and $F_S(x)$ is measured by the statistic *D* which is the greatest vertical distance between $F_S(x)$ and $F_T(x)$

$$D = \max | F_T(x) - F_S(x) |$$

The critical value of D_α is obtained from table for Kolmogorov-Smirnov test

Test for Random Number Generator (RNG)— Uniformity– frequency test

Steps of Kolmogorov-Smirnov (K-S) test for uniformity

- Step 1: Rank the data from smallest to largest. Let $R_{(i)}$ denote the i^{th} smallest observation, so that

$$R_{(1)} \leq R_{(2)} \leq \ldots \leq R_{(N)}$$

- Step 2: Compute

$$D^{+} = \max_{1 \leq i \leq N} \{ i/N - R_{(i)} \}$$

$$F_T(R_{(i)}) = R_{(i)}$$

$$D^{-} = \max_{1 \leq i \leq N} \{ R_{(i)} - ((i-1)/N) \}$$

- Step 3: Compute $D = \max(D^{+}, D^{-})$
- Step 4: Determine critical value D_{α} for given α level
- Step 5: If sample statistic $D > D_{\alpha}$ null hypothesis that the data are a sample from a uniform distribution is rejected.

  IF $D \leq D_{\alpha}$ conclude that no difference is detected.

Test for Random Number Generator (RNG)— Uniformity– frequency test

Example.

For the five random numbers 0.44, 0.81, 0.14, 0.05, and 0.93, perform a test for uniformity by using the Kolmogorov-Smirnov test with the level of significance of 0.05.

Test for Random Number Generator (RNG)— Uniformity– frequency test

Solution: DON'T USE THIS TECHNIQUE. FOLLOW THE NEXT SLIDE

H0: Ri ~ U[0,1]
H1: Ri ≠ U[0,1]

i	xi (Data)	R(i) (Sorted)	i/N	(i-1)/N	$D^+ =$ i/N - R(i)	$D^- =$ R(i) - (i-1)/N
1	0.44	0.05	0.2	0	0.15	0.05
2	0.81	0.14	0.4	0.2	0.26	-
3	0.14	0.44	0.6	0.4	0.16	0.04
4	0.05	0.81	0.8	0.6	-	0.21
5	0.93	0.93	1	0.8	0.07	0.13

N = 5

$D = max(D^+, D^-) = 0.26$

For $\alpha = 0.05$,

$D_\alpha = 0.565 > D$ (D_α is from K-S table)

Hence, H_0 is not rejected. i.e. the random numbers are form a uniform distribution

Test for Random Number Generator (RNG)—Uniformity– frequency test

Solution (OR).

H0: Ri ~ U[0,1]
H1: Ri ≠ U[0,1]

i	xi (Sorted)	F	C.F.	$F_s(x)$	$F_T(x) = x$	$\lvert F_S(x) - F_T(x)\rvert$
1	0.05	1	1	0.2	0.05	0.15
2	0.14	1	2	0.4	0.14	0.26
3	0.44	1	3	0.6	0.44	0.16
4	0.81	1	4	0.8	0.81	0.01
5	0.93	1	5	1.0	0.93	0.07

N = 5

$$D = \max \lvert F_T(x) - F_S(x) \rvert = 0.26$$

For $\alpha = 0.05$,

$D_\alpha = 0.565 > D$ (D$_\alpha$ is from K-S table)

Hence, H_0 is not rejected. i.e. the random numbers are form a uniform distribution

Test for Random Number Generator (RNG)— Uniformity– frequency test

- **Chi-square test: the test statistic is**

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i} \quad (\text{where } \mathrm{df} = k - p - 1)$$

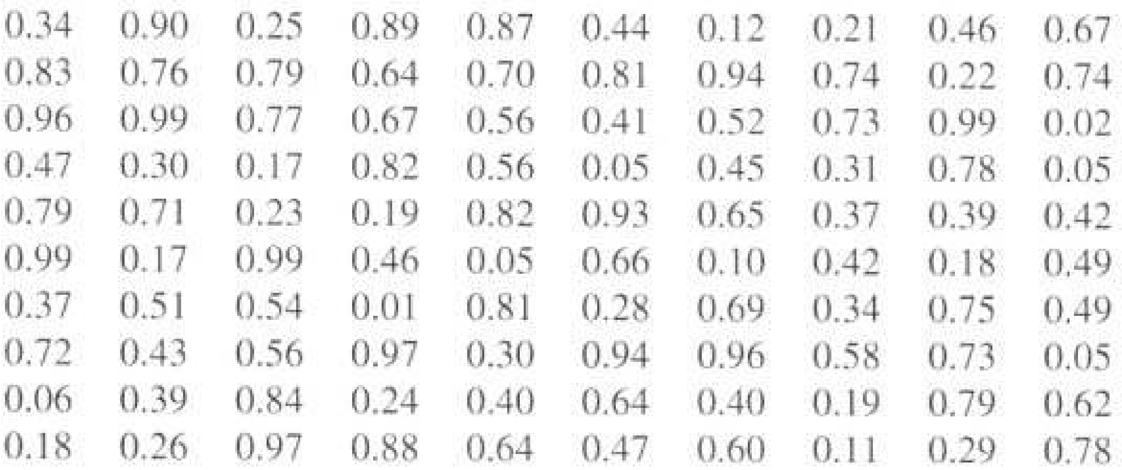
Where:
k = number of categories or cells
p = # of parameters of the hypothesized distribution estimated by sample statistics
o_i = observed frequency for category or cell *i*
E_i = expected frequency for category or cell *I* *(should be at least five)*

For the uniform distribution, E_i is given by, $E_i = p_i N = \frac{N}{k}$, for equally spaced classes. N = total # of observation,

Valid only for large samples, e.g. N >= 50

Test for Random Number Generator (RNG)— Uniformity– frequency test

- **Example-** **Test whether the 100 random numbers (N=100) shown below are uniformly distributed using chi-square test with level of significance of 5%.**

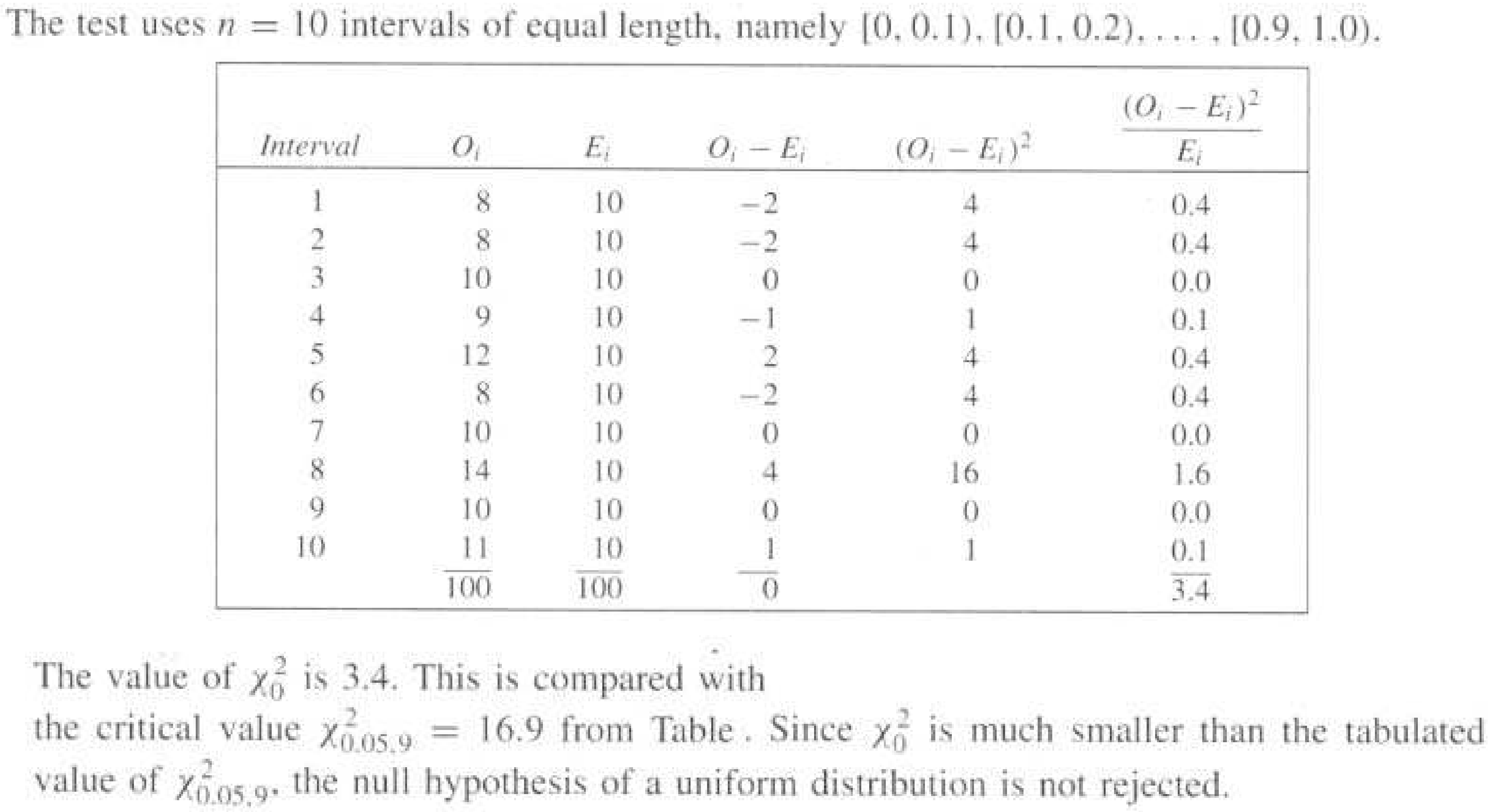


0.34	0.90	0.25	0.89	0.87	0.44	0.12	0.21	0.46	0.67
0.83	0.76	0.79	0.64	0.70	0.81	0.94	0.74	0.22	0.74
0.96	0.99	0.77	0.67	0.56	0.41	0.52	0.73	0.99	0.02
0.47	0.30	0.17	0.82	0.56	0.05	0.45	0.31	0.78	0.05
0.79	0.71	0.23	0.19	0.82	0.93	0.65	0.37	0.39	0.42
0.99	0.17	0.99	0.46	0.05	0.66	0.10	0.42	0.18	0.49
0.37	0.51	0.54	0.01	0.81	0.28	0.69	0.34	0.75	0.49
0.72	0.43	0.56	0.97	0.30	0.94	0.96	0.58	0.73	0.05
0.06	0.39	0.84	0.24	0.40	0.64	0.40	0.19	0.79	0.62
0.18	0.26	0.97	0.88	0.64	0.47	0.60	0.11	0.29	0.78

Test for Random Number Generator (RNG)— Uniformity– frequency test

Solution:

H0: Ri ~ U[0,1]
H1: Ri ≠ U[0,1]



The test uses $n = 10$ intervals of equal length, namely $[0, 0.1), [0.1, 0.2), \ldots, [0.9, 1.0)$.

Interval	O_i	E_i	$O_i - E_i$	$(O_i - E_i)^2$	$\frac{(O_i - E_i)^2}{E_i}$
1	8	10	−2	4	0.4
2	8	10	−2	4	0.4
3	10	10	0	0	0.0
4	9	10	−1	1	0.1
5	12	10	2	4	0.4
6	8	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
	100	100	0		3.4

The value of χ_0^2 is 3.4. This is compared with the critical value $\chi^2_{0.05,9} = 16.9$ from Table. Since χ_0^2 is much smaller than the tabulated value of $\chi^2_{0.05,9}$, the null hypothesis of a uniform distribution is not rejected.

Test for Random Number Generator (RNG)— Independence– autocorrelation test

- Testing the autocorrelation between every m numbers (m is the lag), starting with the i^{th} number
  - ☐ The autocorrelation ρ_{im} between numbers: R_i, R_{i+m}, R_{i+2m}, $R_{i+(M+1)m}$
  - ☐ M is the largest integer such that $i+(M+1)m \leq N$
- Hypothesis:

$$H_0: \quad \rho_{im} = 0, \quad \text{if numbers are independent}$$

$$H_1: \quad \rho_{im} \neq 0, \quad \text{if numbers are dependent}$$

- If the values are uncorrelated:
  - ☐ For large values of M, the distribution of the estimator of ρ_{im}, denoted $\hat{\rho}_{im}$ is approximately normal.

Test for Random Number Generator (RNG)—Independence– autocorrelation test

Look at these number from the left to the right:

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

They look independent and completely random. However, the 5th, 10th, 15th … (every 5 numbers starting from the fifth), the value is large in that position → maybe not independent

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**

→ Need to test the correlation between the random numbers.

→ Compute the autocorrelation between every m numbers (m is the lag) starting with the i^{th} number.

→ The autocorrelation ρ_{im} between $R_i, R_{i+m}, R_{i+2m}, \ldots, R_{i+(M+1)m}$ would be of interest.

→ M is the largest integer such as $i+(M+1)m \leq N$

Test for Random Number Generator (RNG)—Independence– autocorrelation test

→ A nonzero autocorrelation implies a lack of independence

→ Formulate the following (two-tailed) hypothesis:

H0: $\rho_{im} = 0$
H1: $\rho_{im} \neq 0$

→ For large M, the distribution of the estimator of ρ_{im}, $\hat{\rho}_{im}$, is approximately normal if the values R_i, R_{i+m}, R_{i+2m}, …, $R_{i+(M+1)m}$ are not correlated.

Test for Random Number Generator (RNG)—Independence– autocorrelation test

- Test statistics is:
$$Z_0 = \frac{\hat{\rho}_{im}}{\hat{\sigma}_{\hat{\rho}_{im}}}$$

Under the assumption of independence for large M

  - Z_0 is distributed normally with mean = *0* and variance = *1*, and:

$$\hat{\rho}_{im} = \frac{1}{M+1}\left[\sum_{k=0}^{M} R_{i+km} R_{i+(k+1)m}\right] - 0.25$$

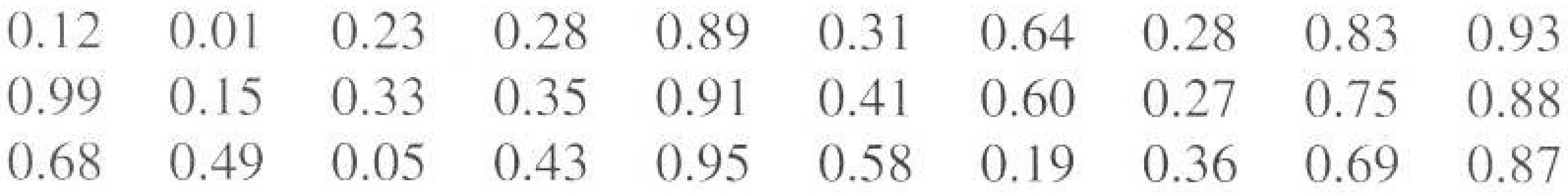
$$\hat{\sigma}_{\rho_{im}} = \frac{\sqrt{13M+7}}{12(M+1)}$$

$$i+(M+1)m \leq N$$

- If $\rho_{im} > 0$, the subsequence has positive autocorrelation
  - High random numbers tend to be followed by high ones, and vice versa.
- If $\rho_{im} < 0$, the subsequence has negative autocorrelation
  - Low random numbers tend to be followed by high ones, and vice versa.

Test for Random Number Generator (RNG)—Independence– autocorrelation test

Example-Test for whether the 3rd, 8th, 13th and so on numbers in the following random numbers are independence using 5% level of significance.



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

Test for Random Number Generator (RNG)— Independence– autocorrelation test

Solution: $H_0: \ \rho_{im} = 0,$ if numbers are independent

$H_1: \ \rho_{im} \neq 0,$ if numbers are dependent

Here, $\alpha = 0.05$, $i = 3$, $m = 5$, $N = 30$, and M = 4

$$\hat{\rho}_{35} = \frac{1}{4+1}[(0.23)(0.28) + (0.28)(0.33) + (0.33)(0.27) + (0.27)(0.05)$$

$$+(0.05)(0.36)] - 0.25$$

$$= -0.1945$$

$$\sigma_{\hat{\rho}_{35}} = \frac{\sqrt{13(4)+7}}{12(4+1)} = 0.1280$$

$$\text{Then,} \quad Z_0 = -\frac{0.1945}{0.1280} = -1.516$$

0.025=α/2=0.05/2 → in Z-Table the probability = 1-α/2=0.975 →z=1.96

- From Z-Table , $z_{0.025} = 1.96$. Hence, the hypothesis is not rejected.

i.e., the RNG is independent.

Test for RNG Independence– shortcomings of autocorrelation test

- The test is not very sensitive for small values of M, particularly when the numbers being tests are on the low side.
- Problem when performing numerous tests:
  - ☐ If $\alpha = 0.05$, there is a probability of 0.05 of rejecting a true hypothesis. (i.e. of **detecting autocorrelation by chance** (rejecting the null hypothesis) while there is no autocorrelation (the null hypothesis is true)
  - ☐ If 10 independence sequences are examined,
    - The probability of finding no significant autocorrelation, by chance alone, is $0.95^{10} = 0.60$.
    - Hence, the probability of detecting significant autocorrelation when it does not exist = *40%*

Conclusion: when conducting the autocorrelation test many times, then autocorrelation can be detected only by chance while no autocorrelation exists.

Generating Random Variates

- **Assume we have desired input distribution for the simulation model (fitted or specified in some way), and RNG (UNIF (0, 1)) (whose generation has been discussed previously)**
- **We wish to generate samples (variates) from this distribution (using random numbers) as input to a simulation model.**

- **Illustrate some widely-used techniques for generating random variates.**
  - Inverse-transform technique
  - Acceptance-rejection technique
  - Special properties

Random Variate Generation

All these techniques assume that a source of uniform, U(0, 1) random numbers is available; R_1, R_2,..., where each R_i has:

$$\text{pdf: } f_R(x) = \begin{cases} 1, & 0 \le x \le 1 \\ 0, & \text{otherwise} \end{cases}$$

and

$$\text{cdf: } F_R(x) = \begin{cases} 0, & x < 0 \\ x, & 0 \le x \le 1 \\ 1, & x > 1 \end{cases}$$

Note: The random variable may be either discrete or continuous.

Random Variate Generation

- **If the r.v. is discrete**, ==>

  x takes on a specific value, and F(x) is a step F^{n}

- **If r.v. is continuous over domain x,** ==>

  f(x) = dF(x) / dx and

  the derivative f(x) is called the pdf.

Mathematically, the cdf is:

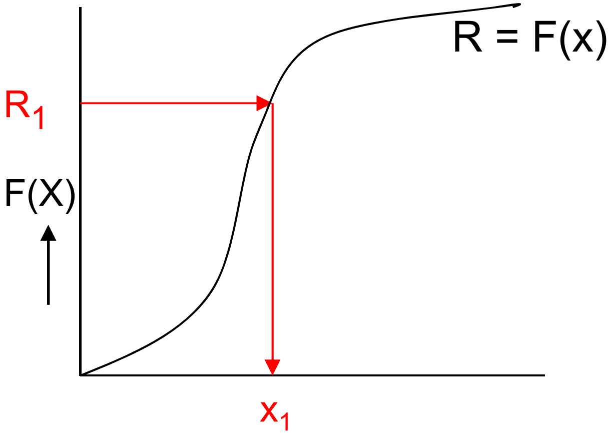
$F(x) = P(X \leq x) = \int_{-\infty}^{x} f(t)dt$, where F(x) is defined over the range $0 \leq F(x) \leq 1$, and f(t) represents the value of the pdf of the variable x, when $X = t$.

Inverse-transform Technique

Continuous distribution

- **The concept:**
  - For cdf function: R = F(x)
  - Generate R_i from uniform (0,1)
  - Find x_i:

$$\boxed{x_i = F^{-1}(R_i)}$$



Steps in inverse-transform technique

assuming x has exponential distribution

- **Exponential Distribution:**

Step 1. Compute the cdf of the desired random variable X: $F(x) = 1 - e^{-\lambda x}$
$x \geq 0$

Step 2. Set $F(x) = R$ on the range of X

Step 3. Solve the equation $F(x) = R$ for X in terms of R.

$$1 - e^{-\lambda x} = R$$

$$e^{-\lambda x} = 1 - R$$

$$-\lambda x = \ln(1 - R)$$

$$x = -\frac{1}{\lambda}\ln(1 - R)$$

Step 4. Generate (as needed) uniform random numbers $R_1, R_2, R_3, \ldots$ and compute the desired random variates $x_i = -\frac{1}{\lambda}\ln(1 - R_i)$

Continuous Uniform Distribution [Inverse-transform for continuous dist]

- **Uniform Distribution:**

  Consider a random variable X that is uniformly distributed on the interval [a, b]

$$F(X) = \frac{X - a}{b - a} = R, \quad a \le X \le b$$

$$\Longrightarrow \quad X = a + (b - a) \,.\, R$$

  - To generate $X_1, X_2, X_3 \ldots$

$$X_i = a + (b - a) \cdot R_i \ , \ i = 1,2,\ldots$$

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{for } x \in [a, b] \\ 0 & \text{otherwise} \end{cases}$$

$$F(x) = \begin{cases} 0 & \text{for } x < a \\ \frac{x-a}{b-a} & \text{for } x \in [a, b) \\ 1 & \text{for } x \ge b \end{cases}$$

Weibull Distribution [Inverse-transform for continuous dist]

- **Weibull Distribution:**

  Consider a random variable X that has weibull distribution (model for time to failure) with location parameter ν set to zero:

$$F(X) = 1 - e^{-(X/\alpha)^{\beta}} = R$$

$$\Rightarrow X = \alpha \cdot [-\ln(1-R)]^{1/\beta}$$

- To generate $X_1, X_2, X_3 \ldots$

$$X_i = \alpha \cdot [-\ln(1-R_i)]^{1/\beta}$$

$$f(x) = \frac{\beta}{\alpha^{\beta}} x^{\beta-1} e^{-(x/\alpha)^{\beta}}, \ x \geq 0$$

Where, α>0 and β>0 are the scale and shape parameters.

$$F(x) = 1 - e^{-(x/\alpha)^{\beta}}, \ x \geq 0$$

Triangular Distribution [Inverse-transform for continuous dist]

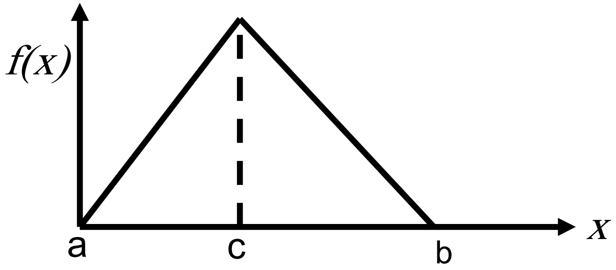
- **Triangular Distribution:**

Example:

Consider a random variable X that has triangular distribution with pdf as follows:

$$f(x)=\begin{cases} x, & 0\le x\le 1 \\ 2-x, & 1<x\le 2 \\ 0, & \text{otherwise} \end{cases}$$

Develop a random variate generation scheme.



$a:\ a\in(-\infty,\infty)$

$b:\ a<b$

$c:\ a\le c\le b$

$a\le x\le b$

$$f(x)=\begin{cases} 0 & \text{for } x<a, \\ \frac{2(x-a)}{(b-a)(c-a)} & \text{for } a\le x<c, \\ \frac{2}{b-a} & \text{for } x=c, \\ \frac{2(b-x)}{(b-a)(b-c)} & \text{for } c<x\le b, \\ 0 & \text{for } b<x. \end{cases}$$

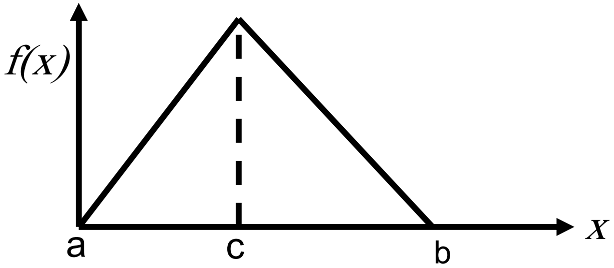
$$F(x)=\begin{cases} 0 & \text{for } x\le a, \\ \frac{(x-a)^2}{(b-a)(c-a)} & \text{for } a<x\le c, \\ 1-\frac{(b-x)^2}{(b-a)(b-c)} & \text{for } c<x<b, \\ 1 & \text{for } b\le x. \end{cases}$$

Triangular Distribution [Inverse-transform for continuous dist]

- **Solution**

From pdf, a = 0, c = 1, b = 2, and from cdf

$$F(x)=\begin{cases}0, & x\le 0\\ \dfrac{x^2}{2}, & 0<x\le 1\\ 1-\dfrac{(2-x)^2}{2}, & 1<x\le 2\\ 1, & x>2\end{cases}$$



For 0≤ X ≤1:

$$F(x)=x^2/2=R\Rightarrow x=\sqrt{2R}$$

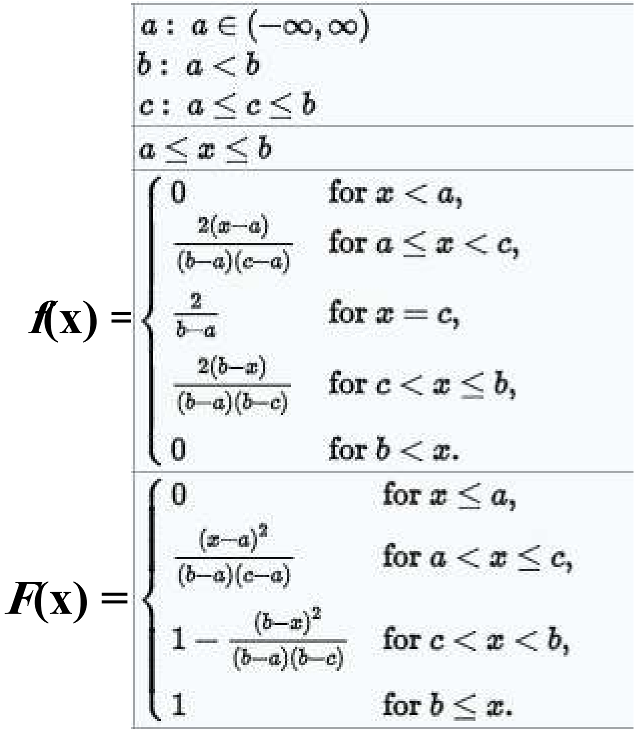
For 1< X ≤2:

$$F(x)=1-\frac{(2-x)^2}{2}=R\Rightarrow 2-(2-x)^2=2R$$

$$\Rightarrow 2-x=\sqrt{2-2R}\Rightarrow x=2-\sqrt{2(1-R)}$$

- To generate $x_1, x_2, x_3 \ldots$ (random variate generation scheme)

$$x_i=\begin{cases}\sqrt{2R_i}, & 0\le R\le 1/2\\ 2-\sqrt{2(1-R_i)}, & 1/2<R\le 1\end{cases}$$



$a:\ a\in(-\infty,\infty)$
$b:\ a<b$
$c:\ a\le c\le b$

$a\le x\le b$

$$f(x)=\begin{cases}0 & \text{for } x<a,\\ \dfrac{2(x-a)}{(b-a)(c-a)} & \text{for } a\le x<c,\\ \dfrac{2}{b-a} & \text{for } x=c,\\ \dfrac{2(b-x)}{(b-a)(b-c)} & \text{for } c<x\le b,\\ 0 & \text{for } b<x.\end{cases}$$

$$F(x)=\begin{cases}0 & \text{for } x\le a,\\ \dfrac{(x-a)^2}{(b-a)(c-a)} & \text{for } a<x\le c,\\ 1-\dfrac{(b-x)^2}{(b-a)(b-c)} & \text{for } c<x<b,\\ 1 & \text{for } b\le x.\end{cases}$$

Random variates with density function

[Inverse-transform for continuous dist.]

Example:

Generate random variates *x* with density function $f(x) = 2x$, $0 \le x \le 1$

Solution:

- $F(x) = \int_0^x f(x)dx = \int_0^x 2x\,dx = x^2, \quad 0 \le x \le 1$
- Now set F(x) = R ==> $R = x^2$
- Next, solve for x, ==> $x = F^{-1}(R) = \sqrt{R}$, $0 \le r \le 1$

Thus, values of x with pdf, $f(x) = 2x$ can be generated by taking the square root of the random number, R.

Empirical Continuous Dist’n with few data [Inverse-transform]

- **When theoretical distribution is not applicable**
- **To collect empirical data:**
  - Resample the observed data (i.e. use the data for the distribution)
  - Interpolate between observed data points to fill in the gaps if the data are drawn from (what is believed is) a continuous input process.
- **For a small sample set (size *n*):**
  - Arrange the data from smallest to largest

$$\mathrm{x}_{(1)} \le \mathrm{x}_{(2)} \le \ldots \le \mathrm{x}_{(n)}$$

  - Assign the probability 1/n to each interval $\mathrm{x}_{(i-1)} \le \mathrm{x} \le \mathrm{x}_{(i)}$

$$\boxed{X_j = \hat{F}^{-1}(R_j) = x_{(i-1)} + a_i\left(R_j - \frac{(i-1)}{n}\right)} \quad \text{When, } (i-1)/n < R_j \le i/n$$

where a_i is the slope of the *i*th segment

$$a_i = \frac{x_{(i)} - x_{(i-1)}}{i/n - (i-1)/n} = \frac{x_{(i)} - x_{(i-1)}}{1/n}$$

Empirical Continuous Dist’n with few data [Inverse-transform]

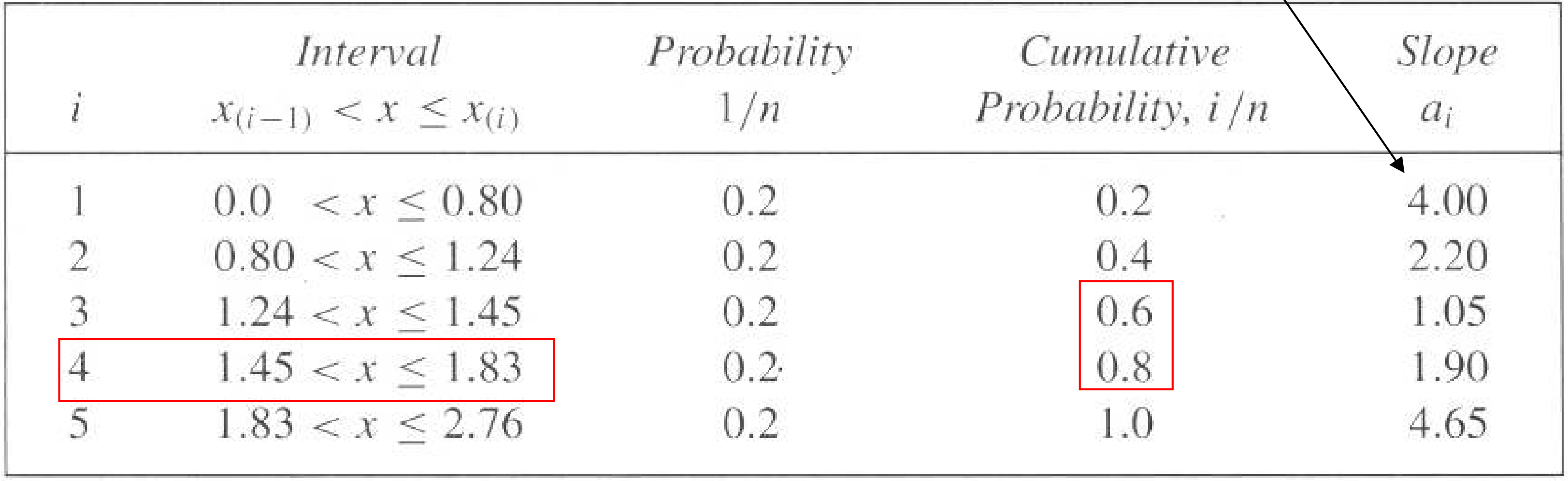
- **Example:** Five observations of fire-crew response times (in minute) to incoming alarms have been collected to be used in a simulation investigating possible alternative staffing and crew-scheduling polices. The data are: 2.76, 1.83, 0.80, 1.45, 1.24. Since the number of data is not enough to define a theoretical distribution, an **empirical distribution** has to be used to produce the input data for the model. Determine the input value to the model for a random number of $R_1 = 0.71$.

Empirical Continuous Dist’n with few data [Inverse-transform]

Solution:

Assume response time X have a range 0 ≤ X ≤ c, where c is unknown but estimated by $\hat{c} = \max\{X_i : i = 1,\dots,n\} = 2.76$

$$a_i = \frac{x_{(i)} - x_{(i-1)}}{1/n} = \frac{0.8-0}{0.2} = 4$$

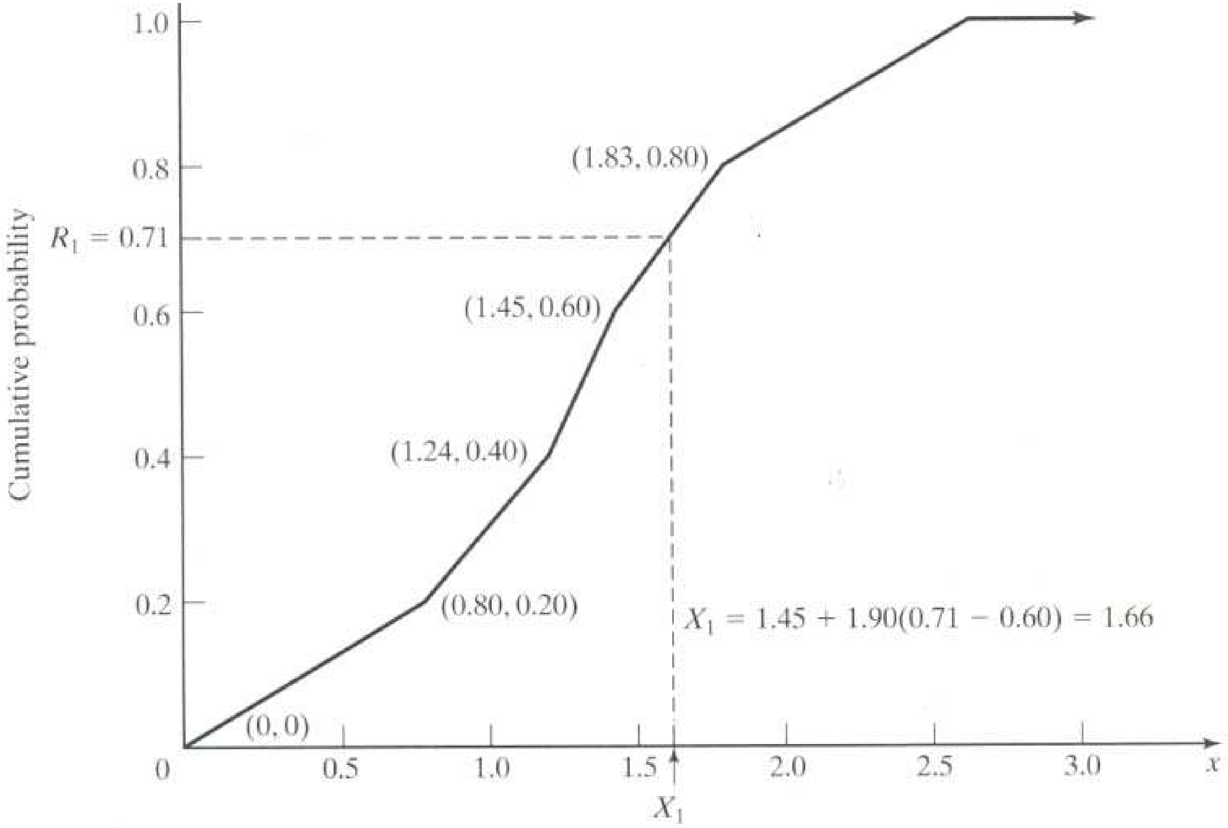


i	*Interval* $x_{(i-1)} < x \leq x_{(i)}$	*Probability* $1/n$	*Cumulative Probability,* i/n	*Slope* a_i
1	$0.0 < x \leq 0.80$	0.2	0.2	4.00
2	$0.80 < x \leq 1.24$	0.2	0.4	2.20
3	$1.24 < x \leq 1.45$	0.2	0.6	1.05
4	$1.45 < x \leq 1.83$	0.2	0.8	1.90
5	$1.83 < x \leq 2.76$	0.2	1.0	4.65

$$\boxed{X_1 = x_{(i-1)} + a_i\left(R_1 - \frac{(i-1)}{n}\right) = x_{(4-1)} + a_4\left(0.71 - \frac{(4-1)}{5}\right) = 1.45 + 1.9(0.71 - 3/5) = 1.66}$$

Empirical Continuous Dist'n with few data [Inverse-transform]

Illustration:



Response times

Figure Empirical cdf of fire-crew response times.

Empirical Continuous Dist’n with Frequency Dist’n [Inverse-transform]

If a large sample of data is available (lets’ say a sample size of about hundred), then it is more convenient to first summarize the data into a frequency distribution with a much smaller number of intervals (rather than representing all data points individually) and then fit a continuous empirical cdf to the frequency distribution.

In this case, the equation for slope a_i is modified to $a_i = \dfrac{x_{(i)} - x_{(i-1)}}{c_i - c_{i-1}}$

Where, c_i is the cumulative probability of the first i intervals of the frequency distribution and $x_{(i-1)} \leq x \leq x_{(i)}$ is the ith interval. The inverse cdf is calculated as

$$\boxed{X_j = \hat{F}^{-1}(R_j) = x_{(i-1)} + a_i\left(R_j - c_{i-1}\right)} \quad \text{When } c_{i-1} < R_j \leq c_i$$

Empirical Continuous Dist’n with Frequency Dist’n [Inverse-transform]

Example:

Suppose that 100 broken widget repair times have been collected. The data are summarized below in terms of the number of observations in various intervals. For example, there were 31 observations between 0 and 0.5 hour, 10 between 0.5 and 1 hour, and so on. It is known that all repairs take at least 15 minutes, so that $X \geq 0.25$ hour always, thus it is set $x_{(0)} = 0.25$. An **empirical distribution** has to be used to produce the input data for the model. Determine the input values to the simulation model for random number of $R_1 = 0.83$ and $R_2 = 0.33$.

i	*Interval (Hours)*	*Frequency*
1	$0.25 \leq x < 0.5$	31
2	$0.5 \leq x < 1.0$	10
3	$1.0 \leq x < 1.5$	25
4	$1.5 \leq x < 2.0$	34

Empirical Continuous Dist'n with Frequency Dist'n [Inverse-transform]

Solution: (n = 100)

i	*Interval (Hours)*	*Frequency*	*Relative Frequency*	*Cumulative Frequency, c_i*	*Slope, a_i*
1	0.25 ≤ x < 0.5	31	0.31	0.31	0.81
2	0.5 ≤ x < 1.0	10	0.10	0.41	5.0
3	1.0 ≤ x < 1.5	25	0.25	0.66	2.0
4	1.5 ≤ x < 2.0	34	0.34	1.00	1.47

Consider $R_1 = 0.83$:

$c_3 = 0.66 < R_1 < c_4 = 1.00$

$\mathbf{X_1} = x_{(4\text{-}1)} + a_4(R_1 - c_{(4\text{-}1)})$

$= 1.5 + 1.47(0.83 - 0.66)$

$= 1.75$

Consider $R_2 = 0.33$:

$c_1 = 0.31 < R_2 < c_2 = 0.41$

$\mathbf{X_2} = x_{(2\text{-}1)} + a_2(R_2 - c_{(2\text{-}1)})$

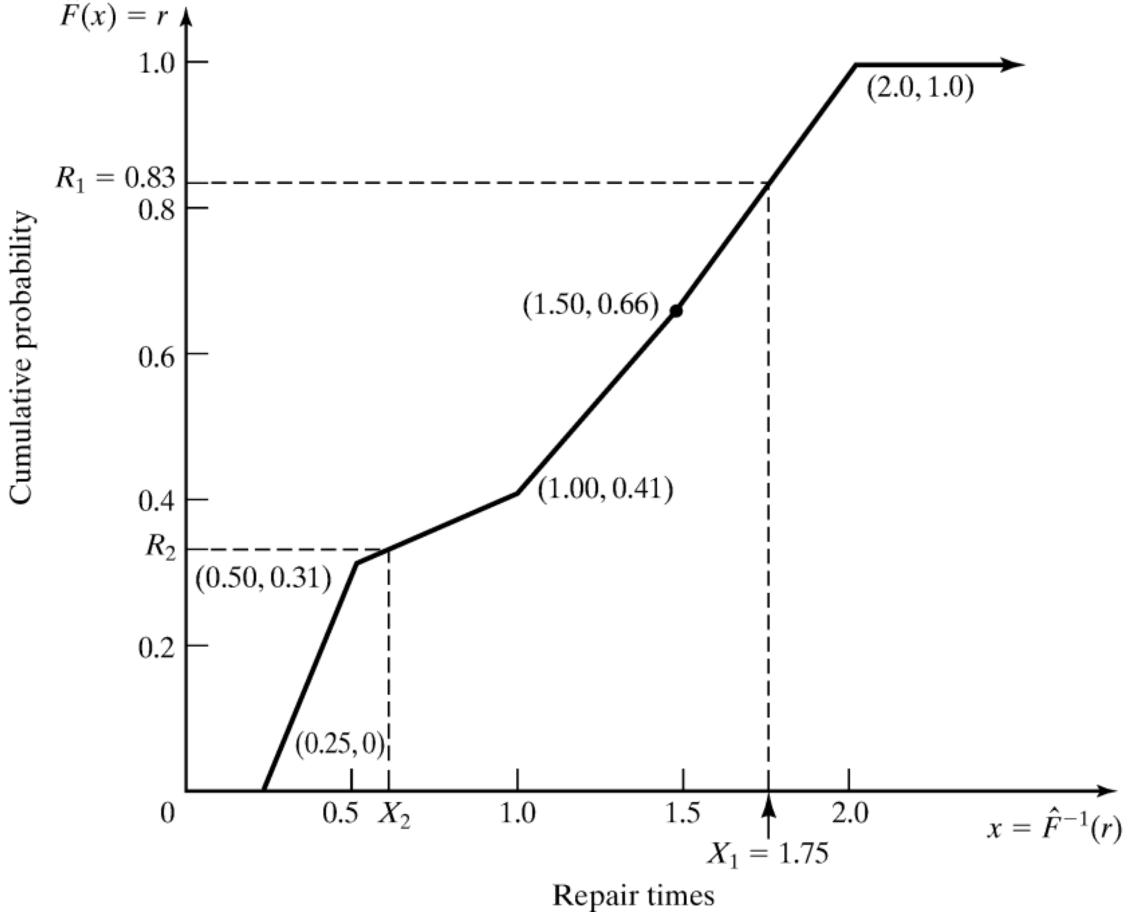
$= 0.5 + 5.0(0.33 - 0.0.31)$

$= 0.6$

$$X = \hat{F}^{-1}(R) = x_{(i-1)} + a_i\left(R - c_{i-1}\right)$$

Empirical Continuous Dist’n with Frequency Dist’n [Inverse-transform]

Illustration:



Continuous distributions without a nice closed-form expression for their cdf

- **There are continuous distributions without a nice closed-form expression for their cdf or its inverse.**
  - Normal distribution
  - Gamma
  - Beta
- **Must approximate (numerically) in these cases**
  - This is what is done anyway to calculate the logarithm function for the exponential CDF
  - For instance, an approximation for the standard normal distribution inverse CDF:

$$X = F^{-1}(R) \approx \frac{R^{0.135} - (1-R)^{0.135}}{0.1975} \qquad 0.0013499 \leq R \leq 0.9986501$$

Discrete Distribution [Inverse-transform]

- **All discrete distributions can be generated via inverse-transform technique**
- **Method: numerically through a table-lookup procedure or, algebraically, the final generation scheme being in-terms of a formula**
- **Does not need interpolation like the continuous variables**
- **Examples of application:**
  - Discrete uniform
  - Geometric
  - Empirical
  -

Discrete Uniform Distributions

[Algebraic procedure]

- **Discrete Uniform Distribution on {1,2,…,k} with pmf and cpf**

$$p(x) = \frac{1}{k}, \quad x = 1, 2, \ldots, k$$

$$F(x) = \begin{cases} 0, & x < 1 \\ \frac{i}{k}, & i \le x < i+1, \; for \quad i \in [1, k-1] \\ 1, & k \le x \end{cases}$$

If the generated random number R satisfies:

$$\frac{i-1}{k} < R \le \frac{i}{k},$$

Then, $X = i$

or, $\boxed{X_i = \text{rounds up}\,(R_i \cdot k) = \lceil R_i \cdot k \rceil}$

Discrete Uniform Distributions

[Algebraic procedure]

- **Example:** **Consider a random variate X is uniformly distributed on {1,2,…,10}, where X represent the number of pallets to be loaded onto a truck. For a random number of 0.78, what should be the input value (random variate) to the simulation model?**

Solution:

$$X_i = \lceil R_i \cdot k \rceil = \lceil 0.78 \cdot 10 \rceil = \lceil 7.8 \rceil = 8$$

Geometric Distributions [Algebraic procedure]

- **Geometric distribution with pmf and cdf**

$$p(x) = p(1-p)^x, \quad x = 0,1,2, \ldots$$
$$F(x) = 1-(1-p)^{x+1}, \quad x = 0,1,2, \ldots$$

$$\mu = \frac{1}{p}; \quad \sigma^2 = \frac{1-p}{p^2}$$

The probability of success on each trial *p* is the parameter of the geometric distribution

If *R* is a generated random number assumed 0 < *R* < 1, then (after some algebraic manipulation),

$$\boxed{X_j \;=\; \text{rounds up}\left(\frac{\ln(1-R_j)}{\ln(1-p)} - 1\right) = \left\lceil \frac{\ln(1-R_j)}{\ln(1-p)} - 1 \right\rceil}$$

Occasionally, there is needed a geometric variate X that can assume values {q, q+1, q+2,…..}. such a variate X can be generated by,

$$\boxed{X_j = q + \left\lceil \frac{\ln(1-R_j)}{\ln(1-p)} - 1 \right\rceil}$$

Geometric Distributions [Algebraic procedure]

- **Example: Generate three values from the geometric distribution on the range {X≥1} (with mean 2) for random numbers of R_1 = 0.932, R_2 = 0.105, and R_3 = 0.687.**

Solution:

For geometric distribution, μ = 1/p ==> p = 1/2, q = 1:

$$X_1 = q + \left\lceil \frac{\ln(1-R_1)}{\ln(1-p)} - 1 \right\rceil = 1 + \left\lceil \frac{\ln(1-0.932)}{\ln(1-0.5)} - 1 \right\rceil = 4,$$

Similarly,

$$X_2 = 1,$$

$$X_3 = 2$$

Empirical Discrete Distribution

[table-lookup procedure]

- **Table-lookup procedure:**

  Let $x_0 = -\infty$, and $x_1, x_2, \ldots, x_n$, be the ordered probability mass points for the random variable X

  Let R be a random number

$$\text{if} \quad F(x_{i-1}) < R \leq F(x_i) \quad \Rightarrow \quad X = x_i$$

An Empirical Discrete Distribution

[table-lookup procedure]

- **Example:** At the end of any day, the number of shipments, X, on the loading dock of IHW company is either 0, 1, or 2 with observed frequency (probability) of occurrence 0.50, 0.30, and 0.20. Internal consultants have been asked to develop a model to improve the efficiency of the loading and hauling operation; as a part of this model, they will need to be able to generate values X, to represent the number of shipments on the loading dock at the end of each day. The consultants decide to model X as a discrete random variable with the distribution given above. Generate a value of the random variate X for a random number of 0.73.

An Empirical Discrete Distribution

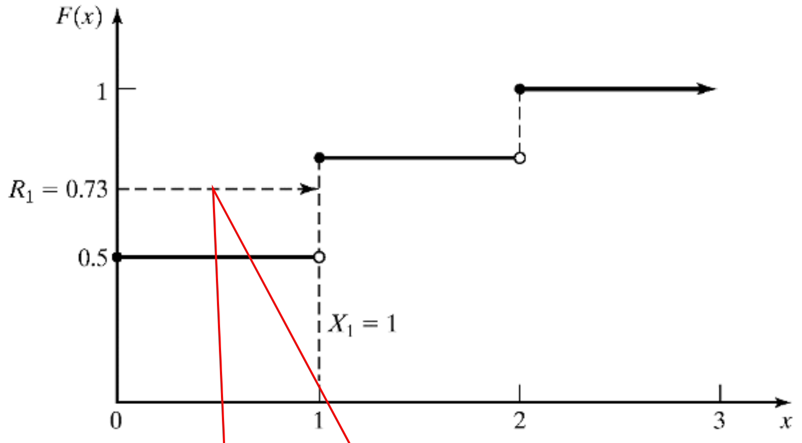
[table-lookup procedure]

Solution: Data - Probability distribution:

x	*p(x)*	*F(x)*
0	0.50	0.50
1	0.30	0.80
2	0.20	1.00

The cdf F(x) = P(X ≤ x), is given by

$$F(x) = \begin{cases} 0, & x < 0 \\ 0.5, & 0 \le x < 1 \\ 0.8, & 1 \le x < 2 \\ 1.0, & 2 \le x \end{cases}$$



Consider $R_1 = 0.73$:
$F(x_{i-1}) < R <= F(x_i)$
$F(x_0) < 0.73 <= F(x_1)$
Hence, $x_1 = 1$

Designing and Executing Simulation Experiments

- **Think of a simulation model as a convenient “testbed” or laboratory for experimentation**
  - Look at different output responses
  - Look at effects, interaction of different input factors
- **Apply classical experimental-design techniques**
  - Factorial experiments — main effects, interactions
  - Fractional-factorial experiments
  - Factor-screening designs
  - Response-surface methods, “metamodels”
  - CRN is “blocking” in experimental-design terminology
  - Process Analyzer (PAN) provides a convenient way to carry out a designed experiment

Tutorial-3 (Lecture 6)

Continued in Lecture-7