Simulation: Random Variables & Stochastic processes

Vineet Sahula

Dept. of Electronics & Communication Engineering
MNIT, Jaipur INDIA

July 2013

Outline

Outline

Outline

Outline

Outline

Outline

Outline

Outline

Simulating an RV

Estimating E[g(X)]

- Let $X = (X_1, X_2, \ldots, X_n)$ denote a random vector
  - having density function $f(x_1, x_2 \ldots, x_n)$
- Objective is to find the expected value of $g(X)$
  - $E[g(X)] = \int\int \ldots \int g(x_1, x_2, \ldots x_n) f(x_1, x_2, \ldots, x_n) dx_1 dx_2, \ldots, dx_n$
- Analytical OR numerical integration is tedious
- Alternately, approximate $E[g(X)]$ by means of SIMULATION

Simulating an RV

Estimating E[g(X)]

- Let $X = (X_1, X_2, \ldots, X_n)$ denote a random vector
  - having density function $f(x_1, x_2 \ldots, x_n)$
- Objective is to find the expected value of $g(X)$
  - $E[g(X)] = \int\int \ldots \int g(x_1, x_2, \ldots x_n) f(x_1, x_2, \ldots, x_n) \, dx_1 dx_2, \ldots, dx_n$
- Analytical OR numerical integration is tedious
- Alternately, approximate $E[g(\mathbf{X})]$ by means of SIMULATION

Simulating an RV

Estimating E[g(X)]

- Let $X = (X_1, X_2, \ldots, X_n)$ denote a random vector
  - having density function $f(x_1, x_2 \ldots, x_n)$
- Objective is to find the expected value of $g(X)$
  - $E[g(X)] = \int\int \ldots \int g(x_1, x_2, \ldots x_n) f(x_1, x_2, \ldots, x_n) \, dx_1 dx_2, \ldots, dx_n$
- Analytical OR numerical integration is tedious
- Alternately, approximate $E[g(\mathbf{X})]$ by means of SIMULATION

Simulating an RV

Estimating E[g(X)]

- Let $X = (X_1, X_2, \ldots, X_n)$ denote a random vector
  - having density function $f(x_1, x_2 \ldots, x_n)$
- Objective is to find the expected value of $g(X)$
  - $E[g(X)] = \int\int \ldots \int g(x_1, x_2, \ldots x_n) f(x_1, x_2, \ldots, x_n)\, dx_1 dx_2, \ldots, dx_n$
- Analytical OR numerical integration is tedious
- Alternately, approximate $E[g(\mathbf{X})]$ by means of SIMULATION

Approximating $E\left[g\left(\mathbf{X}\right)\right]$

Monte Carlo Simulation

- Generate a random RV,
  - $\mathbf{X}^{(1)} = \left(\mathbf{X}_1^{(1)}, \mathbf{X}_2^{(1)}, \cdots, \mathbf{X}_n^{(1)}\right)$ having density function $f\left(x_1, x_2 \ldots, x_n\right)$
  - compute $Y^{(1)} = g\left(\mathbf{X}^{(1)}\right)$
- Generate second RV,
  - $\mathbf{X}^{(2)}$
  - compute $Y^{(2)} = g\left(\mathbf{X}^{(2)}\right)$
- Repeat this r number of times, generating i.i.d RVs
  - $Y^{(i)} = g\left(\mathbf{X}^{(i)}\right)$, $i = 1, 2, \ldots, r$ are generated
  - $\Rightarrow \lim \lim_{r \to \infty} \frac{Y^{(1)} + Y^{(2)} + \cdots + Y^{(r)}}{r} = E\left[Y^{(i)}\right] = E\left[g\left(\mathbf{X}\right)\right]$

Approximating $E\left[g\left(\mathbf{X}\right)\right]$

Monte Carlo Simulation

- Generate a random RV,
  - $\mathbf{X}^{(1)} = \left(\mathbf{X}_1^{(1)}, \mathbf{X}_2^{(1)}, \cdots, \mathbf{X}_n^{(1)}\right)$ having density function $f\left(x_1, x_2 \ldots, x_n\right)$
  - compute $Y^{(1)} = g\left(\mathbf{X}^{(1)}\right)$
- Generate second RV,
  - $\mathbf{X}^{(2)}$
  - compute $Y^{(2)} = g\left(\mathbf{X}^{(2)}\right)$
- Repeat this r number of times, generating i.i.d RVs
  - $Y^{(i)} = g\left(\mathbf{X}^{(i)}\right)$, $i = 1, 2, \ldots, r$ are generated
  - $\Rightarrow \lim \lim_{r \to \infty} \frac{Y^{(1)} + Y^{(2)} + \cdots + Y^{(r)}}{r} = E\left[Y^{(i)}\right] = E\left[g\left(\mathbf{X}\right)\right]$

Approximating $E\left[g\left(\mathbf{X}\right)\right]$

Monte Carlo Simulation

- Generate a random RV,
  - $\mathbf{X}^{(1)} = \left(\mathbf{X}_1^{(1)}, \mathbf{X}_2^{(1)}, \cdots, \mathbf{X}_n^{(1)}\right)$ having density function $f\left(x_1, x_2 \ldots, x_n\right)$
  - compute $Y^{(1)} = g\left(\mathbf{X}^{(1)}\right)$
- Generate second RV,
  - $\mathbf{X}^{(2)}$
  - compute $Y^{(2)} = g\left(\mathbf{X}^{(2)}\right)$
- Repeat this r number of times, generating i.i.d RVs
  - $Y^{(i)} = g\left(\mathbf{X}^{(i)}\right)$, $i = 1, 2, \ldots, r$ are generated
  - $\Rightarrow \lim \lim_{r \to \infty} \frac{Y^{(1)} + Y^{(2)} + \cdots + Y^{(r)}}{r} = E\left[Y^{(i)}\right] = E\left[g\left(\mathbf{X}\right)\right]$

How to generate a Random Vectors, $\mathbf{X}^{(r)}$

Generating random vectors X having specified joint distribution?

- Objective-
  - Generate a sequence of random vectors $\mathbf{X}^{(r)}$ having aprticular distribution,
- As first step, we need to be able to generate random variables from uniform distribution (0,1)

How to generate a Random Vectors, $\mathbf{X}^{(r)}$

Generating random vectors X having specified joint distribution?

- Objective-
  - Generate a sequence of random vectors $\mathbf{X}^{(r)}$ having aprticular distribution,
- As first step, we need to be able to generate random variables from uniform distribution (0,1)

Generating random variable on UNIFORM (0,1)

Generating a Uniform RV on (0,1), MANUALLY

- 10 identical slips, containing numbers 0..9
- Successively select, n chits with replacement
- Equivalent to generating a string of n digits with decimal in front, thus
  can be regarded as a value of a uniform (0,1) RV rounded off to nearest $\left(\frac{1}{10}\right)^n$

Generating random variable on UNIFORM (0,1)

Generating a Uniform RV on (0,1), MANUALLY

- 10 identical slips, containing numbers 0..9
- Successively select, n chits with replacement
- Equivalent to generating a string of n digits with decimal in front, thus
  can be regarded as a value of a uniform (0,1) RV rounded off to nearest $\left(\frac{1}{10}\right)^n$

Generating random variable on UNIFORM (0,1)

Generating a Uniform RV on (0,1), MANUALLY

- 10 identical slips, containing numbers 0..9
- Successively select, n chits with replacement
- Equivalent to generating a string of n digits with decimal in front, thus
  can be regarded as a value of a uniform (0,1) RV rounded off to nearest $\left(\frac{1}{10}\right)^n$

Generating random variable on UNIFORM (0,1)

Generating a Uniform RV on (0,1), on COMPUTER

- Pseudo random instead of truly random
- Starts with initial value X_0 called seed
- Then, recursively compute values using a, c and m
    - $X_{n+1} = (aX_n + c)\ modulo\ m$
    - Thus, each X_n is a number $0, 1, \ldots (m-1)$
- $\frac{X_n}{m}$ is taken as an approximation to a UNIFORM (0,1) RV

Generating random variable on UNIFORM (0,1)

Generating a Uniform RV on (0,1), on COMPUTER

- Pseudo random instead of truly random
- Starts with initial value X_0 called seed
- Then, recursively compute values using a, c and m
  - $X_{n+1} = (aX_n + c)$ *modulo m*
  - Thus, each X_n is a number $0, 1, \ldots (m-1)$
- $\frac{X_n}{m}$ is taken as an approximation to a UNIFORM (0,1) RV

Generating random variable on UNIFORM (0,1)

Generating a Uniform RV on (0,1), on COMPUTER

- Pseudo random instead of truly random
- Starts with initial value X_0 called seed
- Then, recursively compute values using a, c and m
  - $X_{n+1} = (aX_n + c)\ modulo\ m$
  - Thus, each X_n is a number $0, 1, \ldots (m-1)$
- $\frac{X_n}{m}$ is taken as an approximation to a UNIFORM (0,1) RV

Generating random variable on UNIFORM (0,1)

Generating a Uniform RV on (0,1), on COMPUTER

- Pseudo random instead of truly random
- Starts with initial value X_0 called seed
- Then, recursively compute values using a, c and m
  - $X_{n+1} = (aX_n + c)\ modulo\ m$
  - Thus, each X_n is a number $0, 1, \ldots (m-1)$
- $\frac{X_n}{m}$ is taken as an approximation to a UNIFORM (0,1) RV

Estimating the number of distinct entries in a large list

Generating a Uniform RV on (0,1), on COMPUTER

- Pseudo random instead of truly random
- Starts with initial value X_0 called seed
- Then, recursively compute values using a, c and m
  - $X_{n+1} = (aX_n + c)$ *modulo* m
  - Thus, each X_n is a number $0, 1, \ldots (m-1)$
- $\frac{X_n}{m}$ is taken as an approximation to a UNIFORM (0,1) RV

Estimating the number of distinct entries in a large list

Generating a Uniform RV on (0,1), on COMPUTER

- Pseudo random instead of truly random
- Starts with initial value X_0 called seed
- Then, recursively compute values using a, c and m
  - $X_{n+1} = (aX_n + c)$ *modulo m*
  - Thus, each X_n is a number $0, 1, \ldots (m-1)$
- $\frac{X_n}{m}$ is taken as an approximation to a UNIFORM (0,1) RV

Estimating the number of distinct entries in a large list

Generating a Uniform RV on (0,1), on COMPUTER

- Pseudo random instead of truly random
- Starts with initial value X_0 called seed
- Then, recursively compute values using a, c and m
  - $X_{n+1} = (aX_n + c)$ *modulo m*
  - Thus, each X_n is a number $0, 1, \ldots (m-1)$
- $\frac{X_n}{m}$ is taken as an approximation to a UNIFORM (0,1) RV

Estimating the number of distinct entries in a large list

Generating a Uniform RV on (0,1), on COMPUTER

- Pseudo random instead of truly random
- Starts with initial value X_0 called seed
- Then, recursively compute values using a, c and m
  - $X_{n+1} = (aX_n + c) \; modulo \; m$
  - Thus, each X_n is a number $0, 1, \ldots (m-1)$
- $\frac{X_n}{m}$ is taken as an approximation to a UNIFORM (0,1) RV

Estimating the SUM of values of distinct entries in a large list

Generating a Uniform RV on (0,1), on COMPUTER

- Pseudo random instead of truly random
- Starts with initial value X_0 called seed
- Then, recursively compute values using a, c and m
  - $X_{n+1} = (aX_n + c)\ modulo\ m$
  - Thus, each X_n is a number $0, 1, \ldots (m-1)$
- $\frac{X_n}{m}$ is taken as an approximation to a UNIFORM (0,1) RV

Estimating the SUM of values of distinct entries in a large list

Generating a Uniform RV on (0,1), on COMPUTER

- Pseudo random instead of truly random
- Starts with initial value X_0 called seed
- Then, recursively compute values using a, c and m
  - $X_{n+1} = (aX_n + c)$ *modulo m*
  - Thus, each X_n is a number $0, 1, \ldots (m-1)$
- $\frac{X_n}{m}$ is taken as an approximation to a UNIFORM (0,1) RV

Estimating the SUM of values of distinct entries in a large list

Generating a Uniform RV on (0,1), on COMPUTER

- Pseudo random instead of truly random
- Starts with initial value X_0 called seed
- Then, recursively compute values using a, c and m
  - $X_{n+1} = (aX_n + c)\ modulo\ m$
  - Thus, each X_n is a number $0, 1, \ldots (m-1)$
- $\frac{X_n}{m}$ is taken as an approximation to a UNIFORM (0,1) RV

Estimating the SUM of values of distinct entries in a large list

Generating a Uniform RV on (0,1), on COMPUTER

- Pseudo random instead of truly random
- Starts with initial value X_0 called seed
- Then, recursively compute values using a, c and m
  - $X_{n+1} = (aX_n + c)\ modulo\ m$
  - Thus, each X_n is a number $0, 1, \ldots (m-1)$
- $\frac{X_n}{m}$ is taken as an approximation to a UNIFORM (0,1) RV

Outline

Inverse transform method

Proposition for inverse transform method

- Let U be a UNIFORM (0,1) random variable
- If we define a random variable X with continuous distribution function F, then
  - $X = F^{-1}(U)$
- $F^{-1}(u)$is defined as to equal that value of x for which $F(x) = u$

Inverse transform method

Proposition for inverse transform method

- Let U be a UNIFORM (0,1) random variable
- If we define a random variable X with continuous distribution function F, then
  - $X = F^{-1}(U)$
- $F^{-1}(u)$is defined as to equal that value of x for which $F(x) = u$

Inverse transform method

Proposition for inverse transform method

- Let U be a UNIFORM (0,1) random variable
- If we define a random variable X with continuous distribution function F, then
  - $X = F^{-1}(U)$
- $F^{-1}(u)$is defined as to equal that value of x for which $F(x) = u$

Inverse transform method- Proof

Proof for inverse transform method

$$F_X(a) = P\{X \leqslant a\} = P\left\{F^{-1}(U) \leqslant a\right\}$$

- Now, $F(x)$ is a monotone function, it follows that $F^{-1}(U) \leqslant a$, iff $U \leqslant F(a)$
- $F_X(a) = P\{U \leqslant F(a)\} == F(a)$
- Hence, we can simulate a RV X from continuos distribution F, When $F^{-1}$is computable
  - by simulating a RV U, and then setting $X = F^{-1}(U)$

Inverse transform method- Proof

Proof for inverse transform method

$$F_X(a) = P\{X \leqslant a\} = P\{F^{-1}(U) \leqslant a\}$$

- Now, $F(x)$ is a monotone function, it follows that $F^{-1}(U) \leqslant a$, iff $U \leqslant F(a)$
- $F_X(a) = P\{U \leqslant F(a)\} == F(a)$
- Hence, we can simulate a RV X from continuos distribution F, When $F^{-1}$is computable
  - by simulating a RV U, and then setting $X = F^{-1}(U)$

Inverse transform method- Proof

Proof for inverse transform method

$$F_X(a) = P\{X \leqslant a\} = P\left\{F^{-1}(U) \leqslant a\right\}$$

- Now, $F(x)$ is a monotone function, it follows that $F^{-1}(U) \leqslant a$, iff $U \leqslant F(a)$
- $F_X(a) = P\{U \leqslant F(a)\} == F(a)$
- Hence, we can simulate a RV X from continuos distribution F, When $F^{-1}$is computable
    - by simulating a RV U, and then setting $X = F^{-1}(U)$

Simulating an EXPONENTIAL RV

F(x)

$F(x) = 1 - e^{-x}$

- For $F(x) = 1 - e^{-x}$,
- then $F^{-1}(u)$ is that value of x such that $F(x) = u$, i.e.

$$\begin{aligned} 1 - e^{-x} &= u \\ x &= -\log(1 - u) \end{aligned}$$

Simulating an EXPONENTIAL RV ...

F(x)

- Hence, if U is UNIFORM (0,1) variable, then following is a random variable EXPONENTIALLY distributed

$$F^{-1}(U) = -\log(1-U)$$

- As $1-U$ is also UNIFORM on (0,1)
  - $-\log U$ is EXPONENTIAL with mean **1**
  - $-c \log U$ is EXPONENTIAL with mean **c**, as cX has mean **c** if X has mean **1**

Simulating an EXPONENTIAL RV ...

F(x)

- Hence, if U is UNIFORM (0,1) variable, then following is a random variable EXPONENTIALLY distributed

$$F^{-1}(U) \quad = \quad -\log(1-U)$$

- As $1-U$ is also UNIFORM on (0,1)
  - $-\log U$ is EXPONENTIAL with mean **1**
  - $-c\log U$ is EXPONENTIAL with mean **c**, as cX has mean **c** if X has mean **1**

Outline

The Rejection Method

Simulating from distribution f(x), knowing method for g(x)

- Suppose we have method for simulating an RV having density function $g(x)$
- Using this basis, let's simulate from $f(x)$
- First, we simulate Y from g
- Then, ACCEPT this siulated value with a probability proportional to $\frac{f(Y)}{g(Y)}$

The Rejection Method

Simulating from distribution f(x), knowing method for g(x)

- Suppose we have method for simulating an RV having density function $g(x)$
- Using this basis, let's simulate from $f(x)$
- First, we simulate Y from g
- Then, ACCEPT this siulated value with a probability proportional to $\frac{f(Y)}{g(Y)}$

The Rejection Method

Simulating from distribution f(x), knowing method for g(x)

- Suppose we have method for simulating an RV having density function $g(x)$
- Using this basis, let's simulate from $f(x)$
- First, we simulate Y from g
- Then, ACCEPT this siulated value with a probability proportional to $\frac{f(Y)}{g(Y)}$

The Rejection Method

Simulating from distribution f(x), knowing method for g(x)

- Suppose we have method for simulating an RV having density function $g(x)$
- Using this basis, let's simulate from $f(x)$
- First, we simulate Y from g
- Then, ACCEPT this siulated value with a probability proportional to $\frac{f(Y)}{g(Y)}$

The Rejection Method ...

Technique for simulating from distribution f(x)

- If

$$\frac{f(y)}{g(y)} \leqslant c \quad for\ all\ y$$

Step_1 Simulate Y having density $g(Y)$

Step_2 Simulate U

Step_3 if $U \leqslant \frac{f(Y)}{cg(Y)}$, set $X = Y$; Otherwise RETURN to **Step_1**

Simulating a Normal RV

Simulating from Normal distribution

- A normal RV Z has mean $(\mu)= 0$ and variance $(\sigma^2)=1$; absolute value of Z will have density-

$$f(x) = \frac{2}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$

- Given that X with $g(x) = e^{-x}$ is available
- $\frac{f(x)}{g(x)} = \sqrt{\frac{2e}{\pi}} e^{-\frac{(x-1)^2}{2}} \leq \sqrt{\frac{2e}{\pi}}$

Simulating a Normal RV

Simulating from Normal distribution

- A normal RV Z has mean $(\mu)=0$ and variance $(\sigma^2)=1$; absolute value of Z will have density-

$$f(x) = \frac{2}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$

- Given that X with $g(x) = e^{-x}$ is available
- $\frac{f(x)}{g(x)} = \sqrt{\frac{2e}{\pi}} e^{-\frac{(x-1)^2}{2}} \leq \sqrt{\frac{2e}{\pi}}$

Simulating a Normal RV

Simulating from Normal distribution

- A normal RV Z has mean $(\mu)= 0$ and variance $(\sigma^2)=1$; absolute value of Z will have density-

$$f(x) = \frac{2}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$

- Given that X with $g(x) = e^{-x}$ is available
- $\frac{f(x)}{g(x)} = \sqrt{\frac{2e}{\pi}} e^{-\frac{(x-1)^2}{2}} \leq \sqrt{\frac{2e}{\pi}}$

Simulating a Normal RV ...

Simulating from Normal distribution- procedure

1. Generate independent random variables Y and U, Y being exponential with rate 1 and U being uniform on (0,1)
2. If $U \leq e^{-\frac{(x-1)^2}{2}}$, or equivalantly, if $-\log U \geq \frac{(Y-1)^2}{2}$; set $X = Y$ else return to 1.

- Having X, we can generate Z, by letting Z be equally likely to be either X or $-X$

Simulating a Normal RV ...

Simulating from Normal distribution- procedure

1. Generate independent random variables Y and U, Y being exponential with rate 1 and U being uniform on (0,1)
2. If $U \leq e^{-\frac{(x-1)^2}{2}}$, or equivalantly, if $-\log U \geq \frac{(Y-1)^2}{2}$; set $X = Y$ else return to 1.

- Having X, we can generate Z, by letting Z be equally likely to be either X or $-X$

Simulating a Normal RV ...

Simulating from Normal distribution- procedure

1. Generate independent random variables Y and U, Y being exponential with rate 1 and U being uniform on (0,1)
2. If $U \leq e^{-\frac{(x-1)^2}{2}}$, or equivalantly, if $-\log U \geq \frac{(Y-1)^2}{2}$; set $X = Y$ else return to 1.

- Having X, we can generate Z, by letting Z be equally likely to be either X or $-X$

Outline

Hazard rate method

Simulatign an RV, S having hazard rate $\lambda(t)$

- Let F be a CDF with $\bar{F}(0) = 1$
- let λ be hazaerd rate, given by $\lambda(t) = \frac{f(t)}{\bar{F}(t)}$, $t > 0$
- That is, λ represents the instantaneous intensity
  - that an item having life distribution F will fail at time t given it has survived to that time

Hazard rate method

Simulatign an RV, S having hazard rate $\lambda(t)$

- Let F be a CDF with $\bar{F}(0) = 1$
- let λ be hazaerd rate, given by $\lambda(t) = \frac{f(t)}{\bar{F}(t)}$, $t > 0$
- That is, λ represents the instantaneous intensity
  - that an item having life distribution F will fail at time t given it has survived to that time

Hazard rate method

Simulatign an RV, S having hazard rate $\lambda(t)$

- Let F be a CDF with $\bar{F}(0) = 1$
- let λ be hazaerd rate, given by $\lambda(t) = \frac{f(t)}{\bar{F}(t)}$, $t > 0$
- That is, λ represents the instantaneous intensity
  - that an item having life distribution F will fail at time t given it has survived to that time

Hazard rate method ...

Simulating an RV, S having hazard rate $\lambda(t)$

- Let's simulate S having hazard rate function $\lambda(t)$
- To do so, let λ be such that
  - $\lambda(t) \leq \lambda$ for all $t \geq 0$ & $\int_0^\infty \lambda(t)dt = \infty$
- Then, to simulate from $\lambda(t)$ we will
  - simulate a Poisson process, with rate λ &
    - 'count' or 'accept' ONLY certain of these Poisson events
  - COUNT an event that occurs at time t, independently of all else, with probability $\frac{\lambda(t)}{\lambda}$

Hazard rate method ...

Simulating an RV, S having hazard rate $\lambda(t)$

- Let's simulate S having hazard rate function $\lambda(t)$
- To do so, let λ be such that
  - $\lambda(t) \leq \lambda$ for all $t \geq 0$ & $\int_0^\infty \lambda(t)dt = \infty$
- Then, to simulate from $\lambda(t)$ we will
  - simulate a Poisson process, with rate λ &
    - 'count' or 'accept' ONLY certain of these Poisson events
  - COUNT an event that occurs at time t, independently of all else, with probability $\frac{\lambda(t)}{\lambda}$

Hazard rate method ...

Simulating an RV, S having hazard rate $\lambda(t)$

- Let's simulate S having hazard rate function $\lambda(t)$
- To do so, let λ be such that
    - $\lambda(t) \leq \lambda$ for all $t \geq 0$ & $\int_0^\infty \lambda(t)dt = \infty$
- Then, to simulate from $\lambda(t)$ we will
    - simulate a Poisson process, with rate λ &
        - 'count' or 'accept' ONLY certain of these Poisson events
    - COUNT an event that occurs at time t, independently of all else, with probability $\frac{\lambda(t)}{\lambda}$

Hazard rate method ...

Simulating an RV, S having hazard rate $\lambda(t)$

- Let's simulate S having hazard rate function $\lambda(t)$
- To do so, let λ be such that
  - $\lambda(t) \leq \lambda$ for all $t \geq 0$ & $\int_0^\infty \lambda(t)dt = \infty$
- Then, to simulate from $\lambda(t)$ we will
  - simulate a Poisson process, with rate λ &
    - 'count' or 'accept' ONLY certain of these Poisson events
  - COUNT an event that occurs at time t, independently of all else, with probability $\frac{\lambda(t)}{\lambda}$

Hazard rate method: Technique

Generating S having hazard rate

$S : \lambda_S(t) = \lambda(t)$

- Let $\lambda(t) \leq \lambda$ for all $t \geq 0$
- Generate pairs of RVs, U_i, X_i for $i \geq 1$; Where $X_i$being exponential with rate λ & $U_i$being uniform (0,1)
- STOP at

$$N = \min \left\{ n : U_n \leq \frac{\lambda\left(\sum_{i=1}^{n} X_i\right)}{\lambda} \right\}$$

- Set $S = \sum_{i=1}^{N} X_i$

Hazard rate method: Technique

Generating S having hazard rate

$S : \lambda_S(t) = \lambda(t)$

- Let $\lambda(t) \leq \lambda$ for all $t \geq 0$
- Generate pairs of RVs, U_i, X_i for $i \geq 1$; Where $X_i$being exponential with rate λ & $U_i$being uniform (0,1)
- STOP at

$$N = \min \left\{ n : U_n \leq \frac{\lambda\left(\sum_{i=1}^{n} X_i\right)}{\lambda} \right\}$$

- Set $S = \sum_{i=1}^{N} X_i$

Hazard rate method: Technique

Generating S having hazard rate

$S : \lambda_S(t) = \lambda(t)$

- Let $\lambda(t) \leq \lambda$ for all $t \geq 0$
- Generate pairs of RVs, U_i, X_i for $i \geq 1$; Where $X_i$being exponential with rate λ & $U_i$being uniform (0,1)
- STOP at

$$N = \min \left\{ n : U_n \leq \frac{\lambda\left(\sum_{i=1}^{n} X_i\right)}{\lambda} \right\}$$

- Set $S = \sum_{i=1}^{N} X_i$

Hazard rate method: Technique

Generating S having hazard rate

$S : \lambda_S(t) = \lambda(t)$

- Let $\lambda(t) \leq \lambda$ for all $t \geq 0$
- Generate pairs of RVs, U_i, X_i for $i \geq 1$; Where $X_i$being exponential with rate λ & $U_i$being uniform (0,1)
- STOP at

$$N = \min \left\{ n : U_n \leq \frac{\lambda\left(\sum_{i=1}^{n} X_i\right)}{\lambda} \right\}$$

- Set $S = \sum_{i=1}^{N} X_i$

[Hazard rate method:] Definition

Stopping Time

Definition

An integer RV N, is said to be *stopping time* for the sequence $X_1, X_2, \ldots$, if the event $\{N = n\}$ is independent of $X_{n+1}, X_{n+2}, \ldots$ for all $n = 1, 2, \ldots$

[Hazard rate method:] Example

No. of finite successes

- Let X_n, $n = 1, 2, \ldots$ be independent and such that $P\{X_n = 0\} = P\{X_n = 1\} = \frac{1}{2}$, $n = 1, 2, \ldots$
- If we let $N = \min\{n : X_1 + \cdots + X_n = 10\}$
- Then, N is a stopping time.
- e.g. N may be regarded as being the stopping time of EXPERIMENT that successively flips a fair coin and then stops when the number of HEADs reaches 10.

[Hazard rate method:] Example

No. of finite successes

- Let X_n, $n = 1, 2, \ldots$ be independent and such that $P\{X_n = 0\} = P\{X_n = 1\} = \frac{1}{2}$, $n = 1, 2, \ldots$
- If we let $N = \min\{n : X_1 + \cdots + X_n = 10\}$
- Then, N is a stopping time.
- e.g. N may be regarded as being the stopping time of EXPERIMENT that successively flips a fair coin and then stops when the number of HEADs reaches 10.

[Hazard rate method:] Example

No. of finite successes

- Let $X_n, n = 1, 2, \ldots$ be independent and such that $P\{X_n = 0\} = P\{X_n = 1\} = \frac{1}{2},\ n = 1, 2, \ldots$
- If we let $N = \min\{n : X_1 + \cdots + X_n = 10\}$
- Then, N is a stopping time.
- e.g. N may be regarded as being the stopping time of EXPERIMENT that successively flips a fair coin and then stops when the number of HEADs reaches 10.

[Hazard rate method:] Example

No. of finite successes

- Let X_n, $n = 1, 2, \ldots$ be independent and such that $P\{X_n = 0\} = P\{X_n = 1\} = \frac{1}{2}$, $n = 1, 2, \ldots$
- If we let $N = \min\{n : X_1 + \cdots + X_n = 10\}$
- Then, N is a stopping time.
- e.g. N may be regarded as being the stopping time of EXPERIMENT that successively flips a fair coin and then stops when the number of HEADs reaches 10.

Simulating from discrete distributions

Simulating X having probability mass function Pj

- Let X be a discrete RV i.e. $P\{X = x_j\} = P_j$ for all $j = 0, 1 \ldots$, where $\sum_j P_j = 1$
- Inverse transform analogue; Let U be uniform (0,1)

Simulating from discrete distributions

Simulating X having probability mass function Pj

- Let X be a discrete RV i.e. $P\{X = x_j\} = P_j$ for all $j = 0, 1 \ldots$, where $\sum_j P_j = 1$
- Inverse transform analogue; Let U be uniform (0,1)

Simulating from discrete distributions ...

Simulating X having probability mass function Pj

- X can be set as following

$$X = \begin{cases} x_1 & \text{if} \quad U < P_1 \\ x_2 & \text{if} \quad P_1 < U < P_1 + P_2 \\ \vdots & \\ x_j & \text{if} \quad \sum_1^{j-1} P_i < U < \sum_1^{j} P_i \\ \vdots & \end{cases}$$

- As

$$P\{X = x_j\} = P\left\{\sum_1^{j-1} P_i < U < \sum_1^{j} P_i\right\} = P_j$$

Simulating from discrete distributions ...

Simulating X having probability mass function Pj

- X can be set as following

$$X = \begin{cases} x_1 & if \quad U < P_1 \\ x_2 & if \quad P_1 < U < P_1 + P_2 \\ \vdots & \\ x_j & if \quad \sum_1^{j-1} P_i < U < \sum_1^{j} P_i \\ \vdots & \end{cases}$$

- As

$$P\{X = x_j\} = P\left\{\sum_1^{j-1} P_i < U < \sum_1^{j} P_i\right\} = P_j$$

Simulating from Geometric distribution

Simulating X having GEOMETRIC probability mass function

- Let's simulate X having Geometric *pmf*, i.e. $P\{X=i\} = p(1-p)^{i-1}$
- $\sum_{i=1}^{j-1} P\{X=i\} = 1 - P\{X > j-1\} = 1-(1-p)^{i-1}$
- We start with generating U, and then setting X to that value of j for which

$$1-(1-p)^{i-1} < U < 1-(1-p)^{i}$$

$$(1-p)^{j-1} > 1-U > (1-p)^{i}$$

Simulating from Geometric distribution

Simulating X having GEOMETRIC probability mass function

- Let's simulate X having Geometric *pmf*, i.e. $P\{X=i\}=p(1-p)^{i-1}$
- $\sum_{i=1}^{j-1} P\{X=i\}=1-P\{X>j-1\}=1-(1-p)^{i-1}$
- We start with generating U, and then setting X to that value of j for which

$$1-(1-p)^{i-1}<U<1-(1-p)^{i}$$

$$(1-p)^{j-1}>1-U>(1-p)^{i}$$

Simulating from Geometric distribution

Simulating X having GEOMETRIC probability mass function

- Let's simulate X having Geometric *pmf*, i.e. $P\{X=i\}=p(1-p)^{i-1}$
- $\sum_{i=1}^{j-1} P\{X=i\}=1-P\{X>j-1\}=1-(1-p)^{i-1}$
- We start with generating U, and then setting X to that value of j for which

$$1-(1-p)^{i-1}<U<1-(1-p)^{i}$$
$$(1-p)^{j-1}>1-U>(1-p)^{i}$$

Simulating from Geometric distribution ...

Simulating X having GEOMETRIC probability mass function

- As U has same distribution as $1-U$, X can also be defined as

$$X = \min\left\{j : (1-p)^j < U\right\} = \min\left\{j : j > \frac{\log U}{\log(1-p)}\right\}$$
$$= 1 + \frac{\log U}{\log(1-p)}$$

Simulating a Binomial RV

Binomial discrete RV

- A binomial RV (n, p) can be most easily simulated considering that
  - it can be expressed as the sum of n independent Bernoulli RV
- e.g., if $U_1, U_2, \ldots U_n$ are independent Uniform (0,1) RVs then letting

$$X_i = \begin{cases} 1 & \text{if } U_i < p \\ 0 & \text{otherwise} \end{cases}$$

- then $X = \sum_{i=1}^{n} X_i$ is a binomial RV with (n, p)

Simulating a Binomial RV

Binomial discrete RV

- A binomial RV (n, p) can be most easily simulated considering that
  - it can be expressed as the sum of n independent Bernoulli RV
- e.g., if $U_1, U_2, \ldots U_n$ are independent Uniform (0,1) RVs then letting

$$X_i = \begin{cases} 1 & \mathit{if}\ U_i < p \\ 0 & \mathit{otherwise} \end{cases}$$

- then $X = \sum_{i=1}^{n} X_i$ is a binomial RV with (n, p)

Simulating a Binomial RV

Binomial discrete RV

- A binomial RV (n, p) can be most easily simulated considering that
  - it can be expressed as the sum of n independent Bernoulli RV
- e.g., if $U_1, U_2, \ldots U_n$ are independent Uniform (0,1) RVs then letting

$$X_i = \begin{cases} 1 & \mathit{if}\ U_i < p \\ 0 & \mathit{otherwise} \end{cases}$$

- then $X = \sum_{i=1}^{n} X_i$ is a binomial RV with (n, p)

Simulating a Binomial RV: Procedure

Improvements

- <1->The previous method requires generation of n random numbers
- <2->Instead of using value of U, the previous algorithm uses the fact $U_i < p$ or not
- <3->Conditional distribution of U given that $U < p$ is uniform in $(0, p)$ & conditional distribution of U given that $U > p$ is uniform in $(p, 1)$

Procedure

- Step_1 Let $\alpha = 1/p$ and $\beta = 1/(1-p)$
- Step_2 Set $k = 0$
- Step_3 Generate a uniform RV U
- Step_4 If $k = n$ stop, else RESET $k = k+1$
- Step_5 If $U \leq p$ set $X_k = 1$ and RESET U to αU. If $U > p$ set $X_k = 0$ and RESET U to $\beta(U-p)$. RETURN to Step_4

Outline

Alias Method

Alias Method

- Let quantities, $\mathbf{P}$, $\mathbf{P}^{(k)}$, $\mathbf{Q}^{(k)}$, $k \leq n-1$ represent *pmf* on $1, 2, \ldots, n$ i.e. they are n-vectors of non-negative numbers summing to 1
- Additionally, each of $\mathbf{P}^{(k)}$will have at most k non-zero components
- Each of $\mathbf{Q}^{(k)}$will have at most 2 non-zero components

Corollary

Let $\mathbf{P} = \{P_i, i = 1, 2, \ldots, n\}$ *denote pmf, then*
(a) there exists and i*,* $1 \leq i \leq n$*, such that* $P_i < \frac{1}{n-1}$*, and*
(b) for this i*, there exists a* j*,* $j \neq i$*, such that* $P_i + P_j \geq \frac{1}{n-1}$

Alias Method

Alias Method

- Let quantities, $\mathbf{P}$, $\mathbf{P}^{(k)}$, $\mathbf{Q}^{(k)}$, $k \leq n-1$ represent *pmf* on $1, 2, \ldots, n$ i.e. they are n-vectors of non-negative numbers summing to 1
- Additionally, each of $\mathbf{P}^{(k)}$ will have at most k non-zero components
- Each of $\mathbf{Q}^{(k)}$ will have at most 2 non-zero components

Corollary

Let $\mathbf{P} = \{P_i, i = 1, 2, \ldots, n\}$ *denote pmf, then*
(a) there exists and i, $1 \leq i \leq n$, *such that* $P_i < \frac{1}{n-1}$, *and*
(b) for this i, *there exists a* j, $j \neq i$, *such that* $P_i + P_j \geq \frac{1}{n-1}$

Alias Method

Alias Method

- Let quantities, $\mathbf{P}$, $\mathbf{P}^{(k)}$, $\mathbf{Q}^{(k)}$, $k \leq n-1$ represent *pmf* on $1, 2, \ldots, n$ i.e. they are n-vectors of non-negative numbers summing to 1
- Additionally, each of $\mathbf{P}^{(k)}$will have at most k non-zero components
- Each of $\mathbf{Q}^{(k)}$will have at most 2 non-zero components

Corollary

Let $\mathbf{P} = \{P_i, i = 1, 2, \ldots, n\}$ *denote pmf, then*
(a) there exists and i, $1 \leq i \leq n$, *such that* $P_i < \frac{1}{n-1}$, *and*
(b) for this i, *there exists a* j, $j \neq i$, *such that* $P_i + P_j \geq \frac{1}{n-1}$

Alias Method

Alias Method

- Let quantities, $\mathbf{P}$, $\mathbf{P}^{(k)}$, $\mathbf{Q}^{(k)}$, $k \leq n-1$ represent *pmf* on $1, 2, \ldots, n$ i.e. they are n-vectors of non-negative numbers summing to 1
- Additionally, each of $\mathbf{P}^{(k)}$ will have at most k non-zero components
- Each of $\mathbf{Q}^{(k)}$ will have at most 2 non-zero components

Corollary

Let $\mathbf{P} = \{P_i, i = 1, 2, \ldots, n\}$ *denote pmf, then*
(a) there exists and i, $1 \leq i \leq n$, *such that* $P_i < \frac{1}{n-1}$, *and*
(b) for this i, *there exists a* $j, j \neq i$, *such that* $P_i + P_j \geq \frac{1}{n-1}$

Alias Method ...

Alias Method ...

- Any *pmf* **P** can be represented as an equally weighted mixture of $n-1$ *pmf* **Q**, i.e. for suitably defined $Q^{(1)}, Q^{(2)}, \dots Q^{(n-1)}$

$$\mathbf{P} = \frac{1}{n-1}\sum_{k=1}^{n-1} \mathbf{Q}^{(k)}$$

- Alongwith example illustration, a general technique is described

Example- Alias method ... (1)

Example: Alias Method ...

- Let $P_1 = \frac{7}{16}, P_2 = \frac{1}{2}, P_3 = \frac{1}{16}$ stand for **P**. here $n = 3$ and hence $k = 1, 2$
- lets' presume following two 2-point mass-functions $Q^{(1)} \& Q^{(2)}$ to constitute **P**.
  - $Q^{(1)}$: all weight on 3 and 2
  - $Q^{(2)}$: may be derived using $Q^{(1)}$ and $P_j = \frac{1}{2}\left(Q_j^{(1)} + Q_j^{(2)}\right)$ $j = 1, 2, 3$

Example- Alias method ... (1)

Example: Alias Method ...

- Let $P_1 = \frac{7}{16}, P_2 = \frac{1}{2}, P_3 = \frac{1}{16}$ stand for **P**. here $n = 3$ and hence $k = 1, 2$
- lets' presume following two 2-point mass-functions $Q^{(1)}$ & $Q^{(2)}$ to constitute **P**.
  - $Q^{(1)}$: all weight on 3 and 2
  - $Q^{(2)}$: may be derived using $Q^{(1)}$ and $P_j = \frac{1}{2}\left(Q_j^{(1)} + Q_j^{(2)}\right) \quad j = 1, 2, 3$

Example- Alias method ...(2)

Example: Alias Method ...

- $\Rightarrow Q_3^{(2)} = 0$; and
  - $Q_3^{(1)} = 2P_3; Q_2^{(1)} = 1 - Q_3^{(1)} = \frac{1}{8}; Q_1^{(1)} = 0$
  - $Q_3^{(2)} = 0; Q_2^{(2)} = 2P_2 - \frac{7}{8} = \frac{1}{8}; Q_1^{(2)} = 2P_1 = \frac{7}{8}$
- Similar procedure may be followed for case of 4-point mass-function P; $n = 4, k = 3$

Example- Alias method ...(2)

Example: Alias Method ...

- $\Rightarrow Q_3^{(2)} = 0$; and
    - $Q_3^{(1)} = 2P_3; Q_2^{(1)} = 1 - Q_3^{(1)} = \frac{1}{8}; Q_1^{(1)} = 0$
    - $Q_3^{(2)} = 0; Q_2^{(2)} = 2P_2 - \frac{7}{8} = \frac{1}{8}; Q_1^{(2)} = 2P_1 = \frac{7}{8}$
- Similar procedure may be followed for case of 4-point mass-function $\mathbf{P}$; $n = 4\,, k = 3$

Alias Method: General rocedure

General procedure: Alias Method

- This method outlines proceure for any n-point *pmf* **P** can be written as follows; We presume corollary/Lemma above for i and j;

$$\mathbf{P} = \frac{1}{n-1}\sum_{k=1}^{n-1}\mathbf{Q}^{(k)}$$

- Lets define $Q^{(1)}$ concentrating on points i and j
  - which will contain all of the mass for point i by noting that in representtion above, $Q_i^{(k)} = 0$ for $k = 2, 3, \ldots, n-1$
  - i.e. $Q_i^{(1)} = (n-1)P_i$ and so $Q_j^{(1)} = 1 - (n-1)P_i$

Alias Method: General rocedure ... (2)

General procedure : Alias Method

- Writing

$$\mathbf{P} = \frac{1}{n-1} Q^{(1)} + \frac{n-2}{n-1} \mathbf{P}^{(n-1)}$$

  - here, $\mathbf{P}^{(n-1)}$ represents the remaining mass, hence
  -

$$\begin{aligned} P_i^{(n-1)} &= 0 \\ P_j^{(n-1)} &= \frac{n-1}{n-2} \left(P_j - \frac{1}{n-1} Q_j^{(1)} \right) = \frac{n-1}{n-2} \left(P_i + P_j - \frac{1}{n-1} \right) \\ P_k^{(n-1)} &= \frac{n-1}{n-2} P_k, \qquad k \neq i \, or \, j \end{aligned}$$

Alias Method: General rocedure ... (3)

General procedure : Alias Method

- Repeating to expand $\mathbf{P}^{(n-1)}$ likewise

$$\mathbf{P}^{(n-1)} = \frac{1}{n-2} Q^{(2)} + \frac{n-3}{n-2} \mathbf{P}^{(n-2)}$$

- Hence, full expansion,

$$\mathbf{P} = \frac{1}{n-1} Q^{(1)} + \frac{1}{n-1} Q^{(2)} + \frac{n-3}{n-2} \mathbf{P}^{(n-2)}$$

$$\mathbf{P} = \frac{1}{n-1} \left(Q^{(1)} + Q^{(2)} + \ldots + Q^{(n-1)} \right)$$

Algorithm for Alias method

Algorithm

The **P** can now be simulated as follows

- Generating random integer N equally likely to be either $1, 2, \ldots (n-1)$
- If N is such that $Q^{(N)}$ puts positive weight only on point i_N and j_N,
  - then we can set X equal to i_N, if second random number is less that $Q^{(N)}_{i_N}$; equal to j_N, otherwise,

Step_1 Generate $U_1$and set $N = 1 + [(n-1)U_1]$
Step_2 Generate U_2 and set

$$X = \begin{cases} i_N & \textit{if } U_2 < Q^{(N)}_{i_N} \\ j_N & \textit{Otherwise} \end{cases}$$

Error minimization

Minimizing error while generating X

- Let $X_1, X_2, \ldots, X_n$ have given distribution,
  - having density function $f(x_1, x_2 \ldots, x_n)$
- Objective is to find the expected value of $E[g(X)]$
  - $\theta = E[g(X_1, X_2, \ldots X_n)]$

Error minimization

Minimizing error while generating X

- Let $X_1, X_2, \ldots, X_n$ have given distribution,
  - having density function $f(x_1, x_2 \ldots, x_n)$
- Objective is to find the expected value of $E[g(X)]$
  - $\theta = E[g(X_1, X_2, \ldots X_n)]$

Error minimization ...

Minimizing error while generating X

- Generate $X_1^{(1)}, X_2^{(1)}, \ldots X_n^{(1)}$ and then $Y^{(1)} = g\left(X_1^{(1)}, X_2^{(1)}, \ldots X_n^{(1)}\right)$ &
  - Generate $Y^{(2)} = g\left(X_1^{(2)}, X_2^{(2)}, \ldots X_n^{(2)}\right)$
- $\bar{Y} = \sum_{i=1}^{k} Y_i/k$ and $E\left[\overline{Y}\right] = \theta$, $E\left[\left(\overline{Y} - \theta\right)^2\right] = Var\left(\overline{Y}\right)$
- $\overline{Y}$ is an estimator of θ; we wish to minimize $E\left[\left(\overline{Y} - \theta\right)^2\right]$

Error minimization ...

Minimizing error while generating X

- Generate $X_1^{(1)}, X_2^{(1)}, \ldots X_n^{(1)}$ and then $Y^{(1)} = g\left(X_1^{(1)}, X_2^{(1)}, \ldots X_n^{(1)}\right)$ &
  - Generate $Y^{(2)} = g\left(X_1^{(2)}, X_2^{(2)}, \ldots X_n^{(2)}\right)$
- $\bar{Y} = \sum_{i=1}^{k} Y_i/k$ and $E\left[\overline{Y}\right] = \theta,\ E\left[\left(\overline{Y} - \theta\right)^2\right] = Var\left(\overline{Y}\right)$
- $\overline{Y}$ is an estimator of θ; we wish to minimize $E\left[\left(\overline{Y} - \theta\right)^2\right]$

Error minimization ...

Minimizing error while generating X

- Generate $X_1^{(1)}, X_2^{(1)}, \ldots X_n^{(1)}$ and then $Y^{(1)} = g\left(X_1^{(1)}, X_2^{(1)}, \ldots X_n^{(1)}\right)$ &
    - Generate $Y^{(2)} = g\left(X_1^{(2)}, X_2^{(2)}, \ldots X_n^{(2)}\right)$
- $\bar{Y} = \sum_{i=1}^{k} Y_i/k$ and $E\left[\overline{Y}\right] = \theta$, $E\left[\left(\overline{Y} - \theta\right)^2\right] = Var\left(\overline{Y}\right)$
- $\overline{Y}$ is an estimator of θ; we wish to minimize $E\left[\left(\overline{Y} - \theta\right)^2\right]$

Use of Anti-thetic variables

Let $Y_1$and Y_2 be idetically distributed RVs, with mean θ

$$Var\left(\frac{Y_1+Y_2}{2}\right) = \frac{1}{4}(Var(Y_1)+Var(Y_1)+2\,Cov(Y_1,Y_2))$$
$$= \frac{Var(Y_1)}{2}+\frac{Cov(Y_1,Y_2)}{2}$$

Variance reduction- Using Antithetic variables

- It would be more advantageous for $Y_1$and Y_2 NOT to be idetically distributed, BUT negatively correlated
- $X_1, X_2, \ldots, X_n$ are independent and simulated by inverse method from U
  - X_i is simulated from $F_i^{-1}(U)$

Use of Anti-thetic variables

Let Y_1 and Y_2 be idetically distributed RVs, with mean θ

$$Var\left(\frac{Y_1+Y_2}{2}\right) = \frac{1}{4}\left(Var\left(Y_1\right)+Var\left(Y_1\right)+2\,Cov\left(Y_1,Y_2\right)\right)$$
$$= \frac{Var\left(Y_1\right)}{2}+\frac{Cov\left(Y_1,Y_2\right)}{2}$$

Variance reduction- Using Antithetic variables

- It would be more advantageous for Y_1 and Y_2 NOT to be idetically distributed, BUT negatively correlated
- $X_1, X_2, \ldots, X_n$ are independent and simulated by inverse method from U
  - X_i is simulated from $F_i^{-1}(U)$

Use of Anti-thetic variables ...

Variance reduction- Using Antithetic variables

- $Y_1 = g\left(F_1^{-1}(U_1), F_2^{-1}(U_2), \ldots F^{-n}(U_n)\right)$
- Since, $91 - U)$ is also uniform $(0,1)$ and is negatively correlated with U
  - $Y_2 = g\left(F_1^{-1}(1-U_1), F_2^{-1}(1-U_2), \ldots F^{-n}(1-U_n)\right)$
- Hence, if Y_1 and Y_2 are negatively correlated, then generating Y_2 by this method would lead to smaller variance

Use of Anti-thetic variables ...

Variance reduction- Using Antithetic variables

- $Y_1 = g\left(F_1^{-1}(U_1), F_2^{-1}(U_2), \ldots F^{-n}(U_n)\right)$
- Since, $91 - U)$ is also uniform $(0, 1)$ and is negatively correlated with U
  - $Y_2 = g\left(F_1^{-1}(1 - U_1), F_2^{-1}(1 - U_2), \ldots F^{-n}(1 - U_n)\right)$
- Hence, if Y_1 and Y_2 are negatively correlated, then generating Y_2 by this method would lead to smaller variance

Use of Anti-thetic variables ...

Variance reduction- Using Antithetic variables

- $Y_1 = g\left(F_1^{-1}(U_1), F_2^{-1}(U_2), \ldots F^{-n}(U_n)\right)$
- Since, $91 - U)$ is also uniform $(0, 1)$ and is negatively correlated with U
  - $Y_2 = g\left(F_1^{-1}(1 - U_1), F_2^{-1}(1 - U_2), \ldots F^{-n}(1 - U_n)\right)$
- Hence, if Y_1 and Y_2 are negatively correlated, then generating Y_2 by this method would lead to smaller variance

Determining the number of runs

Number of RUNs

- Let use simulation to generate i.i.d. (independent idetically distributed) $Y_1, Y_2, \ldots, Y_r$ having mean μ and variance σ^2
- $\overline{Y}_r = \frac{Y^{(1)} + Y^{(2)} + + Y^{(r)}}{r}$, we use $\overline{Y}_r$ as an estimate of μ
- The precision of this estimate is $Var(\overline{Y}_r) = E\left[(\overline{Y}_r - \mu)^2\right] = \frac{\sigma^2}{r}$

Determining the number of runs

Number of RUNs

- Let use simulation to generate i.i.d. (independent idetically distributed) $Y_1, Y_2, \ldots, Y_r$ having mean μ and variance σ^2
- $\overline{Y}_r = \frac{Y^{(1)}+Y^{(2)}++Y^{(r)}}{r}$, we use $\overline{Y}_r$ as an estimate of μ
- The precision of this estimate is $Var(\overline{Y}_r) = E\left[(\overline{Y}_r - \mu)^2\right] = \frac{\sigma^2}{r}$

Determining the number of runs

Number of RUNs

- Let use simulation to generate i.i.d. (independent idetically distributed) $Y_1, Y_2, \ldots, Y_r$ having mean μ and variance σ^2
- $\overline{Y}_r = \frac{Y^{(1)}+Y^{(2)}++Y^{(r)}}{r}$, we use $\overline{Y}_r$ as an estimate of μ
- The precision of this estimate is $Var(\overline{Y}_r) = E\left[(\overline{Y}_r - \mu)^2\right] = \frac{\sigma^2}{r}$

Determining the number of runs ...

Number of RUNs

- We wish to choose r sufficiently large so that $\frac{\sigma^2}{r}$ is acceptably small
  - BUT σ^2 is not known in advance
- To get around this, we initially simulate k times to evalaute σ^2, and use simulated $Y^{(1)}, Y^{(2)}, \ldots Y^{(k)}$ to estimate σ^2 by sample variance
  - $\sum \frac{(Y^{(k)} - \overline{Y}_k)^2}{k-1}$
- Based on this estimate, rest of the $r - k$ values can be simulated

Determining the number of runs ...

Number of RUNs

- We wish to choose r sufficiently large so that $\frac{\sigma^2}{r}$ is acceptably small
  - BUT σ^2 is not known in advance
- To get around this, we initially simulate k times to evalaute σ^2, and use simulated $Y^{(1)}, Y^{(2)}, \ldots Y^{(k)}$ to estimate σ^2 by sample variance
  - $\sum \frac{\left(Y^{(k)} - \overline{Y}_k\right)^2}{k-1}$
- Based on this estimate, rest of the $r - k$ values can be simulated

Determining the number of runs ...

Number of RUNs

- We wish to choose r sufficiently large so that $\frac{\sigma^2}{r}$ is acceptably small
  - BUT σ^2 is not known in advance
- To get around this, we initially simulate k times to evalaute σ^2, and use simulated $Y^{(1)}, Y^{(2)}, \ldots Y^{(k)}$ to estimate σ^2 by sample variance
  - $\sum \frac{\left(Y^{(k)} - \overline{Y}_k\right)^2}{k-1}$
- Based on this estimate, rest of the $r-k$ values can be simulated

Completed!

END

- Thanks!