

# Probability simulations using MINITAB and True Basic: with selected examples<sup>1</sup>

*Xuejun Dong*

North Dakota State University  
Central Grasslands Research Extension Center  
Streeter, ND 58483

e-mail: [xuejun.dong@ndsu.edu](mailto:xuejun.dong@ndsu.edu)

In stochastic modeling, it is useful to be able to simulate a statistical distribution on the computer. In this short note, I will illustrate how to simulate simple probability distributions using MINITAB, an easy-to-use statistical program widely available to college students. The note is based on my study of the relevant materials from Evans and Rosenthal (2006)<sup>2</sup>, Giordano and Nakanishi (2006)<sup>3</sup> and Chorin and Hald (2006)<sup>4</sup>.

## 1 Simulating nonuniform distributions

If we can simulate a random variable with uniform distribution, denoted as Uniform  $[0,1]$ , then we can obtain continuous random variables with different distributions. There two common methods for doing this: one is the *transformation method* and the other is the *rejection method*. These are discussed in Giordano and Nakanishi (2006) and in Evans and Rosenthal (2006). The transformation method relies on the use of the inversion of functions. However, it involves the evaluation of functions, which can be difficult sometimes. The rejection method, on the other hand, can be used for any distribution, even those with arbitrary densities.

Here are some important formulae that can be used to obtain certain nonuniform distributions, given a uniform distribution. These are based on Evans and Rosenthal (2006).

Suppose  $U_1$  is a random variable uniformly distributed on  $[0,1]$ , then we can do the following to obtain several other continuous distributions:

---

<sup>1</sup> Created: April 1, 2009 / Last modified: October 15, 2009

<sup>2</sup>M. J. Evans and J. S. Rosenthal, 2006. Probability and Statistics: The Science of Uncertainty. W. H. Freeman and Company, New York, NY. pp685.

<sup>3</sup>N. J. Giordano and H. Nakanishi, 2006. Computational Physics. Pearson Education, Inc. Upper Saddle River, NJ. 2nd edition, pp544.

<sup>4</sup> A. J. Chorin and O. H. Hald, 2006. Stochastic Tools in Mathematics and Science. Springer Science + Business Media, Inc. New York, NY. pp147.

- The uniform[L,R] distribution: Calculate  $X = (R - L)U_1 + L$ , then  $X \sim \text{Uniform}[L,R]$ , meaning  $X$  is uniformly distributed on interval  $[L,R]$  with a density function of

$$f(x) = \begin{cases} \frac{1}{R-L} & \text{if } L \leq x \leq R \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Apparently, the density for Uniform[0,1] is

$$f(x) = \begin{cases} 1 & \text{if } 0 \leq x \leq 1 \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

- The exponential( $\lambda$ ) distribution: Calculate  $X = \ln(1/U_1)/\lambda$ , then  $X \sim \text{Exponential}(\lambda)$ . In particular, if  $\lambda = 1$ , we will have  $X \sim \text{Exponential}(1)$ . The density for exponential( $\lambda$ ) distribution is

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & \text{if } x \geq 0 \\ 0 & \text{if } x < 0. \end{cases} \quad (3)$$

And the density for exponential(1) is

$$f(x) = \begin{cases} e^{-x} & \text{if } x \geq 0 \\ 0 & \text{if } x < 0. \end{cases} \quad (4)$$

- The  $N(\mu, \sigma^2)$  distribution: Suppose we have two independent uniform distributions,  $U_1 \sim \text{Uniform}[0,1]$  and  $U_2 \sim \text{Uniform}[0,1]$ . Then we can easily obtain two independent normal distributions

$$X = \sqrt{-2\log(U_1)} \cos(2\pi U_2), \quad Y = \sqrt{-2\log(U_1)} \sin(2\pi U_2) \quad (5)$$

where  $\log()$  is for natural logarithm. That is, we have  $X \sim N(0,1)$  and  $Y \sim N(0,1)$ . Further, if we want to get another normally distributed random variable  $Z$  with mean  $\mu$  and standard deviation  $\sigma$ , we simply use  $Z = \sigma X + \mu$ , then  $Z \sim N(\mu, \sigma^2)$ . The density for  $Z \sim N(\mu, \sigma^2)$  is defined as

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/(2\sigma^2)}. \quad (6)$$

A computer example for simulating a normal distribution is shown in Appendix 4.1.

Appendix 4.2 contains an example using the rejection method to get a sample of an arbitrarily specified distribution. Here for simplicity we choose to use the example provided in Giordano and Nakanishi (2006), page 518, to show how to generate a normal distribution with a mean and standard deviation of 1. We know that for random variable  $x$ , the probability  $P(X = x)$  equals zero at any point  $x$  so that the integral of the density from  $-\infty$  to  $\infty$  will yield 1. However, the density of  $f(x)$  has a defined value at

particular point of the defined range of  $x$ . This method first generates a value of density for a chosen  $x$ , then compares this particular density value with a uniformly distributed random number drawn from the range of zero and a maximum value of the density from the theoretical distribution that we are going to generate numerically. In particular it proceeds in the following four steps:

1. Specify the density and range of interest of the distribution. Suppose we decide to sample a distribution with a density function of  $f_X(x)$  for a random variable  $X$  between  $x_{min}$  and  $x_{max}$ .
2. Use a computer (such as RND function in `True Basic` or `Rand()` in `C`) to generate a sequence of random numbers  $\{x_1, x_2, \dots\}$  uniformly distributed in the range of  $x_{min}$  and  $x_{max}$ .
3. Find the maximum value of  $f_X(x)$  in the range of  $x_{min}$  and  $x_{max}$ . Denote this maximum value as  $f_{max}$ .
4. Pick a number  $x_i$  and evaluate  $f_X(x_i)$ . Now generate a new random variable  $X_{test}$  using the uniform distribution in the range of  $[0, f_{max}]$ . If  $f_X(x_i) < X_{test}$ , remove  $x_i$  from the list of the uniformly distributed random number list  $\{x_1, x_2, \dots\}$ ; otherwise keep  $x_i$  in the list. After the process is repeated for all members of  $\{x_1, x_2, \dots\}$ , the numbers that remain in the list will have a distribution according  $f_X(x)$ , instead of the uniform distribution, according to which the list was originally generated.

## 2 Monte Carlo method for approximating a nonrandom quantity

Now we evaluate several definite integrals using the Monte Carlo method. These three examples are due to Evans and Rosenthal (2006), but we will do it with MINITAB.

- Using uniform distribution to evaluate integral  $I = \int_0^1 \cos(x^2) \sin(x^4) dx$ . This is shown in Appendix 4.3. We can see that with the increase of  $n$ , the approximation gets more accurate, as can be seen from the error estimate,  $M_n \pm 3S/\sqrt{n}$ , where  $M_n$  and  $S$  are mean and standard deviation, respectively. Because uniform density has value of  $f(x) = 1$  (see Equation 2), any integral  $I = \int_0^1 g(x) dx$  can be evaluated using the Monte Carlo method by sampling many replicates of  $\eta_i$  from uniform distribution and approximating the integral as  $I = \frac{1}{n} \sum_{i=1}^n g(\eta_i)$ . However, as discussed in Chorin and Hald (2006), some components of  $g(x)$  (such as  $e^x$ ) might induce a large standard error in the Monte Carlo sampling and thus compromise the efficiency of this method. Importance sampling may be used to address this issue. Also, the sampling errors may be reduced by sampling from distributions other than the uniform distribution, provided  $g(x)$  has appropriate terms representing the densities of those other distributions. We will see two examples below.

Table 1: Approximating the value of  $\pi$  using the Monte Carlo method.

$N$	<i>Estimated pi</i>
100	3
500	3.184
1000	3.152
1500	3.2053333
3000	3.1586667
5000	3.1624
7500	3.1669333
10000	3.1392
10500	3.1295238
12000	3.1183333
15000	3.1237333
20000	3.1614
50000	3.13504
100000	3.14912
150000	3.1394133

- Using exponential distribution to evaluate integral  $I = \int_0^{\infty} 25x^2 \cos(x^2) e^{-25x} dx$ . Because the density of exponential(25) is  $25e^{-25x}$  for  $x \geq 0$ , the integral  $I$  can be written as

$$I = \int_0^{\infty} x^2 \cos(x^2) (25e^{-25x}) dx = E[X^2 \cos(x^2)], \quad (7)$$

where  $X \sim \text{Exponential}(25)$ . The MINITAB code for this example is shown in Appendix 4.4. The method for obtaining an exponential distribution based on a sample of uniform distribution is discussed in Section 1.

- Using normal distribution to evaluate integral  $I = \int_0^{\infty} \sin(x) e^{-x^2/2} dx$ . This can be done by sampling two independent uniform distributions using Equation 5. The MINITAB code for implementing this example is shown in Appendix 4.5.

The next two examples (Appendices 4.6 and 4.7) show how to calculate the area and volume of geometric shapes using Monte Carlo method. By calculating the area of a circle with radius 1, we approximate the value of  $\pi$  (see Table 1). The **True Basic** code is listed in Appendix 4.6. The program runs faster in True Basic (15 seconds) than in MINITAB.

The volume of a ball is approximated using the Monte Carlo method. The code is shown in Appendix 4.7. A simulation result is shown in Table 2.

Finally, the value of  $e$  is approximated by evaluating the integral  $I = \int_1^{10} \frac{1}{x} dx$ , which is Exercise E.6 of Giordano and Nakanishi (2006). We give the MINITAB code for the calculation as shown in Appendix 4.8. Some results are shown in Table 3. We see that the simulation is not accurate and it converges very slowly toward the true value even

Table 2: Approximating the volume of a ball with radius 1 using the Monte Carlo method.

$N$	<i>Estimated volume</i>	<i>True volume</i>
5000	4.16160	4.18879
20000	4.20000	4.18879

Table 3: Approximating the value of  $e$  using the Monte Carlo method.

$N$	<i>Estimated</i>	<i>True value</i>
1000	2.48380	2.71828
10000	2.61218	2.71828
20000	2.61856	2.71828

with a large sample size. As emphasized by Giordano and Nakanishi (2006), Monte Carlo method gets more and more efficient for sampling in higher dimensional systems, because the estimation error is not a function of dimension but in the order of  $1/\sqrt{n}$ .

### 3 Bayes' theorem and conditional probability

The Bayes' theorem provides an opportunity for estimating model parameters through the assimilation of new data. This framework sheds a light on proper understanding of some important data assimilation techniques, such as those based on the *Kalman filtering* (see Pages 89–92 of Charin and Hald (2006)). The following equations are based on pages 38–42 of the above text, but we will add some additional comments as well as analysis, so that the strength of the theorem becomes more evident.

The first form of the Bayes' theorem is stated as: Let  $A$  and  $B$  be two events with  $P(A) \neq 0$  and  $P(B) \neq 0$ . Then,

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}. \quad (8)$$

The second form of the Bayes' theorem is stated by Chorin and Hald (2006) as: Let  $Z = \{Z_j\}$ ,  $j = 1, 2, \dots$ , is a finite or countable partition of the sample space  $\Omega$ , then for the probability  $P(A)$  of an event  $A$ , we have

$$P(Z_j|A) = \frac{P(A|Z_j)P(Z_j)}{\sum_j P(A|Z_j)P(Z_j)}. \quad (9)$$

For the application of the second form of the Bayes' theorem, let's consider the sampling space  $\Omega$  being divided into  $j$  disjoint sections, each with a probability. That means that we have a *prior* belief about the likeliness of occurrence of each of the partitioned sections in the *whole* set  $\Omega$ . Now, this second form of the theorem can be used to update

our belief about the probability of the event of section  $j$  of  $\Omega$  to be happening (that is,  $Z_j$ ) based on the result of our measurement data, e.g., that of event  $A$ . In other words, we will see how the probability of the event  $Z_j$  conditioned on event  $A$  can be changed by using information of  $A$ . Here is the example from Chorin and Hald (2006), pages 39-40:

Let  $\eta_1$  and  $\eta_2$  be two independent and identically distributed random variables with

$$\eta_1, \eta_2 = \begin{cases} 1 & \text{with probability } p \\ 0 & \text{with probability } 1 - p. \end{cases}$$

For the sum  $\eta_1 + \eta_2$  we can deduce

$$\eta_1 + \eta_2 = \begin{cases} 2 & \text{with probability } p^2 \\ 1 & \text{with probability } 2p(1 - p) \\ 0 & \text{with probability } (1 - p)^2. \end{cases}$$

Suppose that before the experiment we thought that the parameter  $p$  had the value  $p = 1/4$  with probability  $1/4$  and the value  $p = 1/2$  with probability  $3/4$ . This is the “prior distribution”. Now we make an experiment and find  $\eta_1 + \eta_2 = 1$ . We can use Equation 9 to see how the experiment affects our belief about the distribution of parameter  $p$ . To do that, let  $A$  be the event that  $\eta_1 + \eta_2 = 1$ ,  $Z_1$  be the event that the value of  $p$  is  $1/4$  (which has a prior probability of  $1/4$ ) and let  $Z_2$  be the event that  $p = 1/2$  (which has a prior probability of  $3/4$ ). Then, using Equation 9 we find that  $P(Z_1|A) = \frac{1}{5}$ . This *posterior* probability for  $Z_1$  is lower than its prior, which is  $\frac{1}{4}$ . This means that our belief that  $p = 1/4$  is weakened in light of one experimental result! Actually, if we use this newly obtained (posterior) probability ( $1/5$ ) as the new prior, and conduct a new observation which also yields  $\eta_1 + \eta_2 = 1$ , then our belief will further be weakened. After about 12 runs of new experiment (all having in the same result:  $\eta_1 + \eta_2 = 1$ ), the final posterior probability is less than 0.01! That means repeatedly observing this same result will eventually lead us to believe that the correct value of  $p$  is almost  $1/2$ , instead of  $1/4$ . The MINITAB code for doing this simulation is shown in Appendix 4.9 and the result is shown in Table 4. To understand this result, let’s recall how the sampling space  $\Omega$  is partitioned (Table 5). It is partitioned only in two segments:  $Z_1$  and  $Z_2$ .  $Z_1$  means  $p$  is lower ( $1/4$ ) while  $Z_2$  means a higher  $p$  value ( $1/2$ ). Their respective probabilities ( $1/4$  vs.  $3/4$ ) suggest that  $p = 1/2$  is more likely, if we don’t consider any new data. Now the repeated occurring of event  $\eta_1 + \eta_2 = 1$  suggests that the probability of this event needs to be increased. Because there are only two choices for the value of  $p$  (either  $1/4$  or  $1/2$ ). Given  $6/16 < 1/2$ , we are forced to prefer  $p = 1/2$  (over  $p = 1/4$ ) in order to be compatible with the measurement (repeated occurrence of  $\eta_1 + \eta_2 = 1$ ).

Now, if we assume our measurements contain a repeated occurrence of  $\eta_1 + \eta_2 = 2$  instead, then how would our posterior probability of  $P(Z_1|A)$  end up with, if all other aspects of the problem remain unchanged? The result is shown in Table 6<sup>5</sup>. We see that under such a situation, the posterior probability of  $P(Z_1|A)$  also decreases, but with a fast speed (becoming less than 0.01 after only two iterations). This can be interpreted

<sup>5</sup>The MINITAB code of Appendix 4.9 will need to be modified for these two lines: `let k1 = 1/16 # P(A|Z1)` and `let k2 = 9/16 # P(A|Z2)`.

Table 4: The prior and posterior probabilities for the event  $Z_1$  conditioned on the event  $A$ , which is defined as  $\eta_1 + \eta_2 = 1$ .

<i>Prior</i>	<i>Posterior</i>
0.2500	0.2000
0.2000	0.1579
0.1579	0.1233
0.1233	0.0954
0.0954	0.0733
0.0733	0.056
0.056	0.0426
0.0426	0.0323
0.0323	0.0244
0.0244	0.0184
0.0184	0.0139
0.0139	0.0104
0.0104	0.0079

Table 5: Probabilities of the events  $\eta_1 + \eta_2$  for each of the two partitions of the sampling space  $\Omega$ :  $Z_1 : p = 1/4$  with a probability of  $1/4$  and  $Z_2 : p = 1/2$  with a probability of  $3/4$ .

$\eta_1 + \eta_2$	$p(p = 1/4)$	$p(p = 1/2)$
2	1/16	1/4
1	6/16	1/2
0	9/16	1/4

similarly as when  $A$  is  $\eta_1 + \eta_2 = 1$ , but here because  $1/16 \ll 1/4$ , each measurement has a more significant effect.

Finally, let's consider the case when  $A$  is defined as the event  $\eta_1 + \eta_2 = 0$ . This time, the repeated occurrence of  $\eta_1 + \eta_2 = 0$  suggests an increase of  $p$ , but it is already true that  $9/16 > 1/4$ , so the partition of  $p = 1/4$  is preferred. We can predict that after a few iterations, the posterior probability of  $P(Z_1|A)$  will approach 1.0, and indeed it does, as indicated in Table 7<sup>6</sup>.

<sup>6</sup>The MINITAB code for this problem (assuming  $A$  is defined as  $\eta_1 + \eta_2 = 0$ ) involves modifications of the following two lines: *let k1 = 9/16 ‡ P(A|Z<sub>1</sub>)* and *let k2 = 1/16 ‡ P(A|Z<sub>2</sub>)*.

Table 6: The prior and posterior probabilities for the event  $Z_1$  conditioned on the event  $A$ , which is defined as  $\eta_1 + \eta_2 = 2$ .

<i>Prior</i>	<i>Posterior</i>
0.2500	0.0357
0.0357	0.0041

Table 7: The prior and posterior probabilities for the event  $Z_1$  conditioned on the event  $A$ , which is defined as  $\eta_1 + \eta_2 = 0$ .

<i>Prior</i>	<i>Posterior</i>
0.2500	0.7500
0.7500	0.9643
0.9643	0.9959

## 4 Appendix

### 4.1 An Example for simulating a normal distribution using MINITAB

```

gmacro
transform

# MINITAB macro to generate normal distributions
# using the transformation method
# on page 115 of Evans and Rosenthal (2004)
# (also described in Giordano and Nakanishi (2006), p516.

erase k1-k6
erase c1-c4

let k1= 1000 # N of random number

Random k1 c1;
  Uniform 0.0 1.0.      # Generate a uniform distribution in c1
Random k1 c2;
  Uniform 0.0 1.0.      # Generate a another uniform distribution in c2

# This is to simulate norm(0,1) using the two uniform (0,1)
# already been generated in c1 & c2 (see p215 of ER, 2004)

let c3=sqrt(2*log(1/c1))*cos(2*3.14159*c2)

# if  $X \sim (0,1)$ , then let  $Z = \sigma * x + \mu$ , we will have  $Z \sim (\mu, \sigma)$ 
# where,  $\mu$  is the mean and  $\sigma$  is stdev of this norm distribution

```

```
# here we assume mu=1 and sigma = 1
```

```
let c4=c3+1
```

```
endmacro
```

## 4.2 An Example of rejection sampling for simulating an arbitrary distribution

This MINITAB macro simulates a Gaussian distribution  $N(1,1)$  using the rejection method as described in Giordano and Nakanishi (2006), pages 517-518.

```
gmacro
```

```
rejection
```

```
let k1=10000
```

```
Random k1 c1;
```

```
Uniform -2.0 4.0.
```

```
let k2=1/sqrt(2*3.14159)
```

```
let c2=k2*exp(-(c1-1)**2/2)
```

```
set c3
```

```
100
```

```
end
```

```
let k4=count(c3)+1
```

```
do k3=1:k1
```

```
call ptest
```

```
if c2(k3)>=c10(1)
```

```
let c3(k4)=c1(k3)
```

```
let k4=k4+1
```

```
else
```

```
next
```

```
endif
```

```
enddo
```

```
delete 1 c3
```

```
endmacro
```

```
gmacro
```

```
ptest
```

```
Random 1 c10;
```

```
Uniform 0 k2.
```

```
endmacro
```

### 4.3 Evaluating a definite integral by sampling a uniform distribution

```
gmacro
```

```
uniform
```

```
# MINITAB macro to implement Monte Carlo approximate of  
# integral  $I = \int_0^1 (\cos(x^2) \sin(x^4)) dx$   
# on page 214 of Evans and Rosenthal (2004)  
# problem 4.5.3
```

```
let k1= 1000 # N of random number  
call unif
```

```
let k1= 10000 # N of random number  
call unif
```

```
let k1= 100000 # N of random number  
call unif
```

```
endmacro
```

```
gmacro
```

```
unif
```

```
# to generate uniform distribution in order to simulate and integral  
# as shown in the main.
```

```
Random k1 c1;  
Uniform 0.0 1.0.
```

```
let c2=cos(c1**2)*sin(c1**4)
```

```
let k2=mean(c2)
```

```
let k3=stdev(c2)
```

```
let k4=k2-3*k3/sqrt(k1)
```

```
let k5=k2+3*k3/sqrt(k1)
```

```
name k1 'N' k2 'M_n' k3 'S' k4 'lowerB' k5 'higherB'

name c1 'u' c2 'T_i'

print k1-k5

endmacro
```

#### 4.4 Evaluating a definite integral by sampling an exponential distribution

```
gmacro
exponent

# MINITAB macro to implement Monte Carlo approximation of
# integral  $I = \int_0^{\infty} (25 x^2 \cos(x^2) e^{-25x}) dx$ 
# on page 214 of Evans and Rosenthal (2004)
# problem 4.5.4

erase k1-k5
erase c1-c3

let k1= 1000 # N of random number
call ex

let k1= 10000 # N of random number
call ex

let k1= 100000 # N of random number
call ex

erase k1-k5
erase c1-c3

endmacro

gmacro
ex

# to generate exponential distribution in order to simulate the integral
# as shown in the main

Random k1 c1;
Uniform 0.0 1.0.
```

```

let c2=-loge(c1)/25

let c3=c2**2*cos(c2**2)

let k2=mean(c3)

let k3=stdev(c3)

let k4=k2-3*k3/sqrt(k1)

let k5=k2+3*k3/sqrt(k1)

name k1 'N' k2 'M_n' k3 'S' k4 'lowerB' k5 'higherB'

name c1 'u_i' c2 'x_i' c3 'T_i'

print k1-k5

endmacro

```

#### 4.5 Evaluating a definite integral by sampling a normal distribution

```

gmacro
normal

# MINITAB macro to implement Monte Carlo approximation of
# integral  $I = \int_0^{\infty} \sin(x) e^{-x^2/2} dx$ 
# on page 216 of Evans and Rosenthal (2004)
# problem 4.5.6

erase k1-k6
erase c1-c4

let k1= 1000 # N of random number
call no

let k1= 10000 # N of random number
call no

# N of random number!!!! this part is very slow (needs 15 minutes!)

let k1= 100000
call no

```

```
erase k1-k6
erase c1-c4

endmacro

gmacro
no

# to generate normal(0,1) distribution in order to simulate the integral
# as shown in the main

Random k1 c1;
  Uniform 0.0 1.0.      # Generate a uniform distribution in c1

Random k1 c2;
  Uniform 0.0 1.0.      # Generate a another uniform distribution in c2

# This is to simulate norm(0,1) using the two distributions of uniform (0,1)
# already been generated in c1 & c2 (see p215 of ER, 2004)

let c3=sqrt(2*log(1/c1))*cos(2*3.14159*c2)

do k6 = 1:k1
  if c3(k6) > 0
    let c4(k6) = sin(c3(k6))
  else
    let c4(k6) = 0
  endif
enddo

let k2=mean(c4)*sqrt(2*3.14159)

let k3=3*sqrt(k2*(1-k2)/k1)

let k4=k2-k3

let k5=k2+k3

name k1 'N' k2 'M_n' k3 'S' k4 'lowerB' k5 'higherB'

# same code as in page 216 of ER 2004

name c1 'u_i' c2 'v_i' c3 'X_i' c4 'T_i'

print k1-k5
```

```
endmacro
```

## 4.6 A True Basic program for approximating the value of $\pi$

```
! pai

! This program is to approximate the value of pi
! using the Monte Carlo method

! Assume the radius of a circle is is 1 and the origin is set to the center
! of the circle

randomize

OPEN #1: NAME "DATA", ORG TEXT, CREATE NEWOLD, ACCESS OUTPUT
print #1: "N", "estimated PAI"
print

dim x(1),y(1),samples(15),z(15)

mat read samples
data 100,500,1000,1500,3000,5000,7500,10000,10500,12000,15000,20000,50000,100000,150000

FOR j = 1 to 15

let n = samples(j)
mat redim x(n),y(n)

FOR i = 1 to n
  let x(i) = RND
  let y(i) = RND
NEXT i

let C = 0

FOR i = 1 to n
  let a = (x(i)^2 + y(i)^2)^0.5
  if a < 1 THEN
    let C = C + 1
  end if
NEXT i

let z(j) = 4*C/n

print #1: n,z(j)
```

```
NEXT j
```

```
end
```

#### 4.7 A MINITAB macro for approximating the volume of a ball with radius 1

```
gmacro
```

```
ball
```

```
# This MINITAB macro approximates the volume of a ball
```

```
# using the Monte Carlo method
```

```
# Assume the radius of the ball is 1 and the origin of the 3-d coordinates
```

```
# is set to the center of the ball.
```

```
#let k2 = 1000
```

```
#call b
```

```
let k2=20000
```

```
call b
```

```
#erase k1-k6
```

```
#erase c1-c3
```

```
endmacro
```

```
gmacro
```

```
b
```

```
let k1 = 0
```

```
Random k2 c1;
```

```
Uniform 0.0 1.0.
```

```
Random k2 c2;
```

```
Uniform 0.0 1.0.
```

```
Random k2 c3;
```

```
Uniform 0.0 1.0.
```

```
do k3 = 1:k2
```

```
let k5=c1(k3)**2 + c2(k3)**2 + c3(k3)**2
```

```
if sqrt(k5) < 1
```

```

    let k1 = k1 + 1
  else
    next
  endif
enddo

let k4=8*k1/k2
let k6=4/3*3.14159

name k2 'N' k4 'estimate' k6 'true'

print k2 k4 k6
endmacro

```

#### 4.8 A MINITAB macro for approximating the value of $e$ using the Monte Carlo method

```

gmacro
euler

# This MINITAB macro approximates the universal number e
# using the Monte Carlo method

# As suggested by GN (2006), we start with evaluating the definite integral
#  $\int_1^{10} (1/x) dx$ , which equals  $\log_e(10)$ 

#let k2 = 1000
#call eu

let k2=20000
call eu

erase k1-k5
erase c1-c2

endmacro

gmacro
eu

let k1 = 0

Random k2 c1;          # generate values for x
  Uniform 1.0 10.0.

```

```

Random k2 c2;
  Uniform 0.1 1.0.      # generate values for y=1/x

do k3 = 1:k2
  if c2(k3) <= 1/c1(k3)
    let k1 = k1 + 1
  else
    next
  endif
enddo

let k4 = k2 / (k2 + 8.1 * k1)  # rectangle area is 9*0.9 = 8.1
let k5 = 10**(k4)

name k2 'N' k5 'e-est'

print k2 k5
endmacro

```

#### 4.9 A MINITAB macro for recursively simulating a posterior probability

```

gmacro
bayes1

# For example on page 39 of Chorin and hald (2006)

# A: event that eta1 + eta2 = 1

# Our prior distribution is:

# p = 1/4 with probability of 1/4; p = 1/2 with probability 3/4.

# To do this problem using the second form of the Bayes' theorem,
# we consider the sampling space OMEGA as partitioned into Z1 and Z2
# with Z1 be the event that p = 1/4, and Z2 be the event that p = 1/2.

# We consider the probability of Z1 (that is p = 1/4) given A (that is
# eta1 + eta2 = 1).

let k1=2*(1/4)*(3/4)  # P(A|Z1)

let k2=2*(1/2)*(1/2)  # P(A|Z2)

let k3=1/4  # Prior probability for Z1 (prob that p=0.25 is 1/4)

```

```
set c1
  100
end

let k11=count(c1)+1

let k10=0

while k10=0

  let k4=k1*k3/(k1*k3+k2*(1-k3))
  let c1(k11)=k3
  let c2(k11)=k4
  let k11=k11+1
  let k3=k4

  if k4<0.01 or k4>0.99
    let k10=1
  endif

endwhile

delete 1 c1-c2
name c1 'Prior_Z1' c2 'posterior_Z1'

erase k1-k11

endmacro
```