Lecture 2: Generating random numbers #2

Fugo Takasu
Dept. Information and Computer Sciences
Nara Women's University
takasu@ics.nara-wu.ac.jp

26 April 2006

1 Generating exponential random number

If an uniform random variable $0 < x < 1$ is given, exponential random variable is readily derived. We define a new random variable $y = -\frac{\ln x}{\lambda}$. Then y is exponentially distributed with $\gamma = \lambda$. This can be shown as follow. Let $p(x)$ be the pdf of a uniform random variable x and $q(y)$ be the pdf of y.

$$\begin{aligned} p(x) &= 1 \quad \text{for} \quad 0 < x < 1 \\ &= 0 \quad \text{for otherwise} \end{aligned}$$

From the basic rule of variable transform, $|q(y)dy| = |p(x)dx|$, we see

$$\begin{aligned} q(y) &= p(x)\left|\frac{dx}{dy}\right| \\ &= p(x)\lambda\exp[-\lambda y] \\ &= \lambda\exp[-\lambda y] \end{aligned}$$

thus, exponential random variable is easily obtained from an uniform random variable. This transformation has been implemented as `expdev()` which returns an exponential random number with $\gamma = 1$.

2 Generating gaussian random numbers

With the transformation of two uniform random variables x_1, x_2, $(0 < x_1, x_2 < 1)$,

$$\begin{aligned} y_1 &= \sqrt{-2\ln x_1}\cos 2\pi x_2 \\ y_2 &= \sqrt{-2\ln x_1}\sin 2\pi x_2 \end{aligned}$$

y_1 and y_2 both are gaussian random variable with $m = 0$ and $\sigma = 1$. This transformation has been implemented as gasdev() and is available for use.

Gaussian with mean m and variance σ^2 is often written as $N(m, \sigma^2)$. Let x be a random variable of $N(0, 1)$. Then $y = a + bx$ is random variable of $N(a, b^2)$. That is, any gaussian random variable is derived from $N(0, 1)$ with simple transformation.

3 Random numbers generated by C

We have learnt how to generate random numbers using a routine listed in Numerical Recipes in C. Random number generator ran2() returns a double number uniformly distributed between 0.0 and 1.0. The ran2() is used to generate other random numbers, expdev() that returns an exponential distributed number with $\gamma = 1$, and gasdev() a gaussian with $m = 0$ and $\sigma = 1$. Here is a sample program to generate these three random numbers and write them to files.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define STEP 100

extern double ran2(long), expdev(long), gasdev(long) ;

int main(void)
{
  long seed; /* seed is long integer */

  double rand1, rand2, ran3;
  int i;
  FILE *fp_uniform, *fp_exp, *fp_gauss;

  seed = (long)time(NULL);   /* seed should be chosen different at every new  trial */

  /* Initialize the random generator ran2 with NEGATIVE argument */
  ran2(-seed);      /* This must be done only once! */

  printf("Seed is %ld\n", seed);
  fp_uniform = fopen("uniform.out","w");
  fp_exp = fopen("exp.out", "w");
  fp_gauss = fopen("gauss.out", "w");

  for(i=0; i<STEP; i++){

    /* Uniform random variable is generated and returned by calling ran2() */
    rand1 = ran2(seed);
    rand2 = expdev(seed);
    rand3 = gasdev(seed);

    printf("%f %f %f\n", rand1, rand2, rand3)
    fprintf(fp_uniform, "%f\n", rand1);
    fprintf(fp_exp, "%f\n", rand2);
    fprintf(fp_gauss, "%f\n", rand3);

  }

  fclose(fp_uniform);
  fclose(fp_exp);
  fclose(fp_gauss);

  return 0;
}
```

4 Checking generated random numbers

The above sample program generates 100 random numbers (uniform, exponential, and gaussian) and writes these into three separate files. We now want to check if these generated numbers reflect uniform, exponential, and gauusian random numbers. We use *Mathematica* to do this.

4.1 Checking the exponential random numbers

We do the same to see if generated exponential random numbers are really distributed "exponentially".

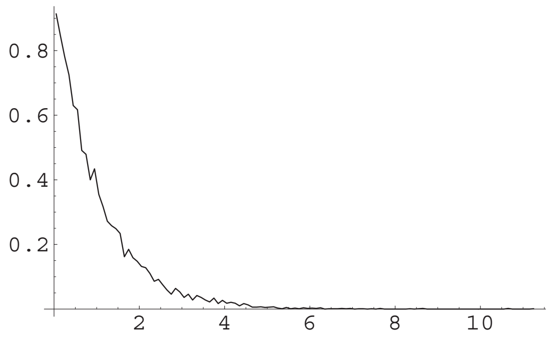
```
data = ReadList["exp.out",Real];
maxdata = Max[data]
len = Length[data]
```

```
11.2286

10000
```

```
dx = 0.1;
categories = 1/dx;
counts = BinCounts[ data,{0,maxdata,dx}]
midpoints = Table[x-dx/2,{x,dx,maxdata+dx,dx}];

dist2 = Transpose[ {midpoints, counts/len*categories}];
```

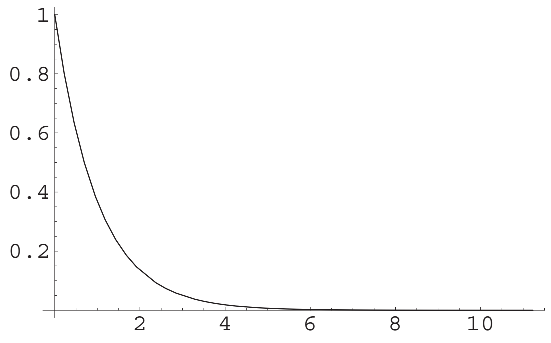
```
914, 846, 781, 725, 630, 617, 491, 479, 400, 434, 355, 317, 272, 258, 249, 234,
162, 185, 159, 148, 132, 128, 110, 86, 92, 75, 59, 46, 64, 53, 36, 46, 28, 42,
36, 28, 22, 34, 17, 27, 18, 21, 18, 10, 17, 13, 6, 6, 7, 5, 6, 7, 3, 1, 5, 1, 3,
1, 4, 2, 3, 2, 4, 0, 1, 1, 1, 2, 1, 2, 0, 1, 1, 0, 1, 0, 2, 0, 0, 0, 0, 0, 0, 1, 0,
1, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 1
```

```
g = ListPlot[ dist2,PlotJoined->True, PlotRange->All]
```



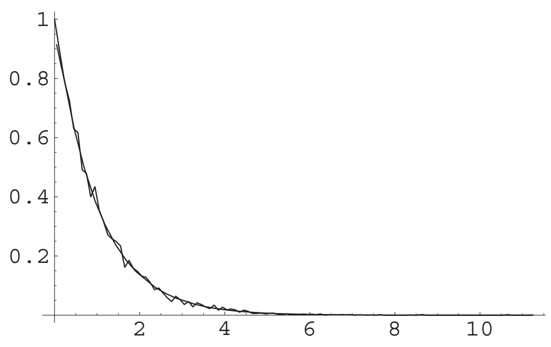
```
Graphics
```

```
g2 = Plot[ Exp[-lambda x]lambda/.lambda->1,{x,0,maxdata},
    PlotRange->All]
```



```
Graphics
```

```
Show[g,g2]
```



```
Graphics
```

```
meanLog = Apply[ Plus, data]/len
```

```
1.0036
```

```
varianceLog = Apply[Plus, (data - meanLog)^2]/len
```

```
1.01242
```

4.2 Exercise

The function expdev() returns an exponential random number of $\gamma = 1$. As already shown simple transformation expdev()$/\gamma$ gives a random number whose pdf is $\gamma \exp[-\gamma x]$. Using this relationship, generate exponential random numbers of $\gamma = 2$ and confirm that the frequency distribution of these numbers coincides with $\gamma \exp[-\gamma x]$ where $\gamma = 2$.

4.3 Checking the gaussian random numbers

Same above.

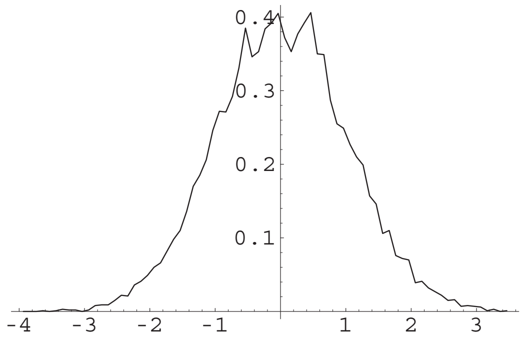
```
data = ReadList["gauss.out",Real];
maxdata = Max[data]
mindata = Min[data]
len = Length[data]
```

```
3.42319

 3.98613

10000
```

```
dx = 0.1;
categories = 1/dx;
counts = BinCounts[ data,{mindata,maxdata,dx}]
midpoints = Table[x-dx/2,{x,mindata+dx,maxdata+dx,dx}];

dist2 = Transpose[ {midpoints, counts/len*categories}];
```

```
 0, 0, 0, 1, 0, 1, 3, 2, 2, 0, 2, 8, 9, 9, 15, 22, 21, 36, 41, 49, 60, 66, 82, 98,
 110, 136, 170, 185, 206, 246, 272, 271, 292, 331, 385, 346, 353, 384, 392,
 405, 372, 353, 377, 392, 406, 350, 349, 287, 255, 249, 227, 210, 199, 157,
 146, 106, 110, 76, 72, 70, 39, 41, 32, 27, 22, 15, 16, 7, 8, 7, 6, 1, 3, 0, 1
```

```
g = ListPlot[ dist2, PlotJoined->True, PlotRange->All]
```

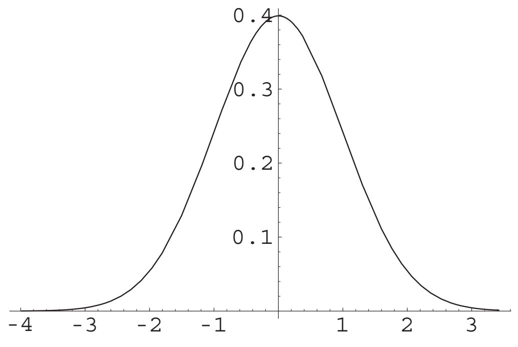


```
  Graphics
```

```
mean=.

gauss = 1/Sqrt[2Pi]/sd Exp[-(x-mean)^2/2/sd^2]/.{mean->0, sd->1}
```

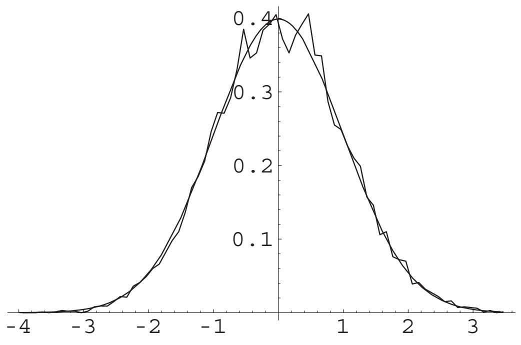
$$\begin{array}{l} x^2 \\ {}_2 \\ 2 \end{array}$$

```
g2 = Plot[ gauss,{x,mindata,maxdata}]
```



```
Graphics
```

```
Show[g,g2]
```



```
Graphics
```

```
meanGauss = Apply[ Plus, data]/len
```

```
0.0171967
```

```
varianceGauss = Apply[Plus, (data - meanGauss)^2]/len
```

```
0.993531
```

4.4 Exercise

The function gasdev() returns a gaussian random number of $m = 0$ and $\sigma = 1$. As already shown, simple transformation $y = ax + b$ where x is $N(0,1)$ and a and b are constant gives a random number of $N(b, a^2)$. Using this relationship, generate gaussian random numbers of, say, $m = 1$ and $\sigma = 2$ and confirm that the frequency distribution of these numbers coincides with $N(1,4)$.

5 Summary

We are now a bit confident that we are good at generating random numbers in C programming and that these random numbers are of good quality (we hope so). We will use these to simulate a variety of stochastic processes in the course.