

# JavaプログラミングのTip集

## 第12回

2008年2月24日  
アイティエス



## 乱数(Random Number)を考える

ある区間内で表現できる数の出現頻度が同じ割合になる → **一様乱数**

実際には擬似乱数という有限の周期で表される乱数列を扱う

↓ コンピュータで発生させる

**pseudo random number**

<用途>  
シミュレーション分野  
暗号分野



## 乱数の生成法

### ●線形合同法(linear congruential method)

1951年D. H. Lehmer 提唱『レーマー法』

- ①混合合同法  $X_{n+1} = kX_n + c \pmod{M}$
- ②乗算合同法  $X_{n+1} = kX_n \pmod{M}$  ← **今回採用**  
(c=0の場合が乗算合同法)

### ●M系列乱数(M sequence random number)

### ●メルセンヌツイスター(Mersenne Twister)

1997年松本眞と西村拓士の両氏により提唱  
『R』で採用



## 乗算合同法で乱数を発生する

レーマー(Lehmer)の漸化式

$$X_{n+1} = kX_n \pmod{M}$$

- ①k、 $X_0$ 、Mの値はなんでもよいが、乱数らしき値を得るには値を選定する必要がある
- ②mod Mとはある式の値をMで割った余りを表現する記号である
- ③乱数がどの程度の周期で現れるかはMの値で決まる



## 例 漸化式の扱い

- k=23
- $X_0 = 50249347$
- $M = 2^{15}$

$$X_{n+1} = kX_n \pmod{M}$$

$$\begin{aligned} X_0 &= 50249347 \\ X_1 &= 23 \times 50249347 \pmod{2^{15}} \\ X_2 &= 23 \times 7621 \pmod{2^{15}} \\ X_3 &= 23 \times 11443 \pmod{2^{15}} \\ X_4 &= 23 \times 1045 \pmod{2^{15}} \\ &\vdots \end{aligned}$$

$X_0 = 50249347$  ← 乱数の種  
 $X_1 = 7621$  ← Seedという  
 $X_2 = 11443$   
 $X_3 = 1045$   
 $X_4 = 24035$   
 $X_5 = 28517$   
 $X_6 = 531$   
 $X_7 = 12213$   
 $X_8 = 18755$   
 $X_9 = 5381$   
 $X_{10} = 25459$



## 一様乱数のプログラム

- 「RandTest.java」で作成する
- 3つの引数を設ける

下限值 上限値 乱数個数

```
>java RandTest 10 99 10  
###乱数データ10件###  
30  
41  
12  
76  
88  
11  
43  
61  
24  
79
```

2桁の乱数を10個作成した例



## RandTest.java その1

- 乱数の下限、上限をセットできるしくみで構成する

```
import java.io.*;
import java.lang.Math;

class RandTest{
public static void main(String[] args){
int ans;
int r2,r1;
int i,x2;
int x1 = 50249347;
int ix = 0;
int imax= 0;
int data[] = new int [10000];
r1=Integer.parseInt(args[0]);
r2=Integer.parseInt(args[1]);
imax=Integer.parseInt(args[2]);
for(i=0;i<10000;i++){
data[i] = 0;
}
```

下限値・・・第1引数args[0]  
上限値・・・第2引数args[1]  
乱数個数・・・第3引数args[2]



## RandTest.java その2

```
while(true){
if(ix>= imax){
break;
}
else{
x2 = lehmer(x1);
x1 = x2;
ans=(int)((r2-r1+1)*x2/(int)Math.pow(2,15));
ans =ans +r1;
data[ix] = ans;
ix++;
}
}
System.out.println("###乱数データ "+ imax+ "件" + " ###");
for(i=0;i<imax;i++){
System.out.println(data[i]);
}
```

乱数個数imaxだけループして配列dataに生成した乱数を格納していく



## RandTest.java その3

//Lehmerによる乗算合同法

```
static int lehmer(int x){
long n;
int k=23;
int m = (int)Math.pow(2,15);
long w;
w = k * x;
n = w % m;
return (int)n;
}
```

入力パラメータの選定  
k=23  
X0= 50249347(初期値)  
M=2<sup>15</sup>



## 一様乱数の出来具合を調べる

- ・ Rで評価する
- 今回は視覚的な評価だけで厳密には行わない

```
>java RandTest2D 0 9999 2000
###乱数データ 2000件 ###
```



2次元プロット  
RandTest.javaを改造して2000件のデータをx軸、y軸に振り分けてファイル化する  
randdata1.txt randdata2.txt

```
>java RandTestH 0 9999 2000
###乱数データ 2000件 ###
```



ヒストグラム  
RandTest.javaを改造して2000件のデータをファイル化する  
randdata.txt

```
>java RandTest3D 0 9999 3000
###乱数データ 3000件 ###
```



3次元プロット  
RandTest.javaを改造して3000件のデータをx軸、y軸、z軸に振り分けてファイル化する  
rdata1.txt rdata2.txt rdata3.txt



## 2次元プロット用のRコマンド

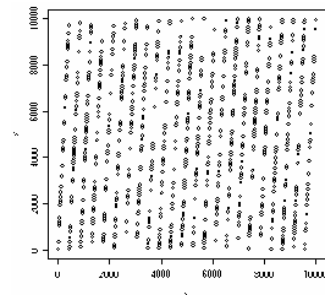
```
> x<- scan("C:¥¥Program Files¥¥R¥¥R-2.6.1¥¥randdata1.txt")
Read 1000 items
> y<- scan("C:¥¥Program Files¥¥R¥¥R-2.6.1¥¥randdata2.txt")
Read 1000 items
> plot(x,y)
>
```

- 関数scan("ファイルパス")  
ファイルからデータを読みベクトルを作成する
- 関数plot()  
ベクトルxやyを座標として散布図などを作図する



## 2次元プロット

- ・ ランダムな表現が目標になる



<今回>

- ①線形性が見られるので乱数の完成度は未だである
- ②ランダム表現の実現を求められる



## ヒストグラム用のRコマンド

```
> r<- scan("C:\\Program Files\\R\\R-2.6.1\\randdata.txt")
Read 2000 items
> hist(r,right=FALSE)
>
```

●関数hist()  
度数分布を棒グラフで表現したベクトルデータのヒストグラムを作成する

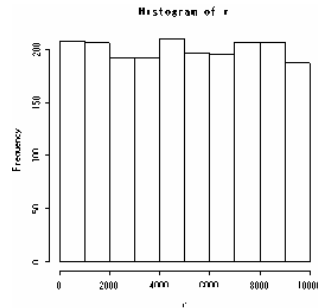
関数hist()のパラメータ

right=FALSEは、階級の区間を下限値以上で上限値未満とする指定で使う



## ヒストグラム

- 棒グラフの高さが一様になっているかを見る



<今回>

- 3000個のデータのため一様な割合が見えない
- 1万個以上のデータで見る必要がある



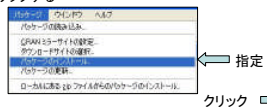
## 3次元プロットをするための準備

- パッケージをインストールする

デフォルトでは下記のパッケージしか実装されていないのでscatterplot3dをダウンロードする

```
> search()
[1] ".GlobalEnv"      "package:stats"    "package:graphics"
[4] "package:grDevices" "package:utils"    "package:datasets"
[7] "package:methods" "Autoloads"        "package:base"
```

- Rのメニューから「パッケージのインストール」をクリックする



指定

クリック



- 「Japan」を指定して「OK」をクリックする

Tsukuba 指定

## 関数 "scatterplot3d" のダウンロード

- 3次元的に散布図を描くことができるパッケージをダウンロードする

downloaded 527 Kb

- 「scatterplot3d」を指定して「OK」をクリックする

クリック



scatterplot3d指定

ダウンロードメッセージ

package 'scatterplot3d' successfully unpacked and MD5 sums checked  
The downloaded packages are in  
C:\Documents and Settings\...Local Settings\Temp\Rtmp64YVh6\downloaded\_packages



## 3次元プロット用のRコマンド

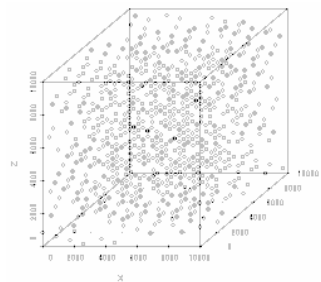
```
> x<- scan("C:\\Program Files\\R\\R-2.6.1\\rdata1.txt")
Read 1000 items
> y<- scan("C:\\Program Files\\R\\R-2.6.1\\rdata2.txt")
Read 1000 items
> z<- scan("C:\\Program Files\\R\\R-2.6.1\\rdata3.txt")
Read 1000 items
> library(scatterplot3d)
> scatterplot3d(x,y,z)
>
```

●関数library(パッケージ名)  
指定のパッケージを読み出す  
●関数scatterplot3d()  
3次元表現の散布図を描く



## 3次元プロット

- 3軸に1000個のデータを振り分けて表現する



<今回>

- 中心にデータの固まりが見られる
- 2次元プロットと合わせて考えると乱数の完成度は弱い

入力パラメータの選定と乱数プログラムの改良が必要と判る

