ランダム数の作成方法

目次

1 乱数って、ランダム？

本来の「乱数」（でたらめの数）とは、人間の意図が加わらず、作為的でなく、規則的でもなく、再現性もおこらず、ということが備わってなければならない。でたらめかどうかの判断は、統計的な検定をおこなって棄却できない（周期性、規則性がない）から、といって保証されることも一般には難しい。そこである程度の理論的裏付けがある場合には、擬似乱数の作成法といい、この乱数を暗号の理論などにもちいる。「メルセンヌ・ツイスタ」とよばれる手法は、ある手法に基づいた乱数列生成式（あるいは生成法）の族を指し、内部状態の大きさや周期は設定可能である。1996 年に国際会議で発表されたもので（1998 年 1 月に論文掲載）松本眞と西村拓士による。(i) $2^{19937}-1$ という長い周期が証明されている。(ii) 高次元（623 次元）に均等分布する。(iii) 出力の中のすべてのビットが統計的に十分ランダムである。(Wikipedia) 過去には、Microsoft 社の表計算ソフト「エクセル」アドインに `http://www.ntrand.com/jp`、このアルゴリズムを組み込む方法も公開されていた。現在も Python, Ruby, R, MATLAB, C++ などに標準ライブラリーに取り入れられていた。また一様乱数ではなく、特定の分布、単変量、多変量の分布を直接生成する。`http://www.ntrand.com/jp/normal-distribution-single/` では正規分布の解説も分かり易く述べられている。

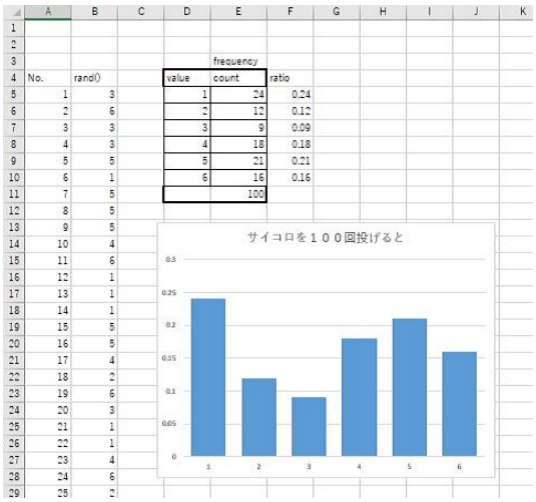

図 1 エクセルでサイコロ投げを集計

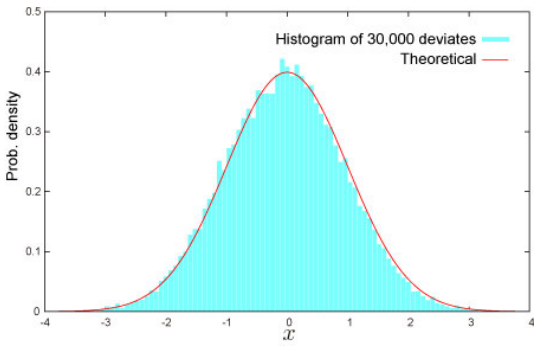

図 2 正規分布のシュミレーション結果

1.1 擬似（ぎじ）乱数

もっとも基本的な考え方で線形合同法とよばれる。種数（日付とか時刻）から一次式の繰り返し式で、範囲を限定するために合同 x_0; seed , repeat $x_n = ax_{n-1}$ modulo m（乗数合同法）x_0; seed , repeat $x_n = (ax_{n-1} + c)$ modulo m（混合合同法）

2 表計算ソフトや R による乱数

厳密な意味での”でたらめ”、ランダムという定義はかなり難しい。ここではある程度の曖昧さをもつとして、パッケージを利用して概要を眺める。

(i) 統計データ分析ソフトウェア「R」による正規乱数の生成 R: rnorm(100) (”r”は randomized normal を 100 個のこと) 生成命令と描画（ヒストグラム）: 図 2 正規分布のシュミレーション結果 では正規乱数の集計と密度関数のグラフを重ね合わせて比較している。

```
> hist(rnorm(n=300, mean=50, sd=10)
# ヒストグラム 個数=300, 平均=50, 標準偏差=10
> xdata <- seq(-3,3,0.05)          #データの範囲設定;-3〜3、幅 0.05 刻み
> plot(xdata, dnorm(xdata, mean=0, sd=1.0))
#確率密度関数 dist. of norm(正規分布) の描画
> curve(dnorm, -4, 4, type="I") # 曲線として密度関数を描く
```

(ii) エクセルの命令 =RAND() 関数によるもの: よく用いられる表計算ソフトでは、組み込み関数、乱数生成 (=rand(), 引数なしでカッコをつける) と 集計関数 (=frequency(データ範囲、区間範囲), 数式入力 CTRL+SHFT+ENTER) によって簡単にできる。上図（図 1 エクセルでサイコロ投げを集計）では 1 から 6 までの数を生成する命令=RANDBETWEEN(1,6) をセルに入れ、これを個数分だけドローする。=RAND() から変換するには, 整数値にする関数=int(6*RAND() +1) を組合せればよい。またたとえば、１０８人の中から３人を選出するには、=RAND(1,108) を 3 回ドローする。同じものが出たら、追加すればよいから簡単である。

2.1 生成乱数により、積分計算の近似をする

擬似乱数； x_0 :*seed*; repeat : $x_n = ax_{n-1}$ modulo m（乗数合同法）x_0 :*seed*; repeat : $x_n = (ax_{n-1} + c)$ modulo m（混合合同法）

乱数発生; パソコンや電卓などでは簡単に（正確かどうかは問わなければ）単位区間 $(0, 1)$ のあいだの数を生成できる。この U を一様乱数とよぶ。これをもとにさまざまな乱数が新たに作り出せる。
“RANDOMIZE; U:=RND.”

積分の近似 関数 $y = g(x)$ の原始関数が求められなくても計算できる。なぜならば，期待値は $E[g(U)] = \int_0^1 g(x)dx$ であるから、大数の法則を用いて、この数値に収束 (確率収束) することが知られている。

(1) 関数 $y = g(x), 0 \le x \le 1$ の積分 $\int_0^1 g(x)dx$ の近似値； パソコン当初の BASIC プログラム

```
10 RANDOMIZE;
20 INPUT K;
30 S=0;
40 FOR I=1 TO K;
50 U=RND; S=S+g(U);
60 NEXT;
70 PRINT S/K.
```

(2) 関数 $y = g(x), a \le x \le b$ の積分 $\int_a^b g(x)dx$ の近似値；$h(y) = (b-a)g(a+[b-a]y)$ とおけば、$\int_a^b g(x)dx = \int_0^1 h(x)dx$ となるから、区間 $[a,b]$ でも単位区間の積分に帰着される。

(3) 関数 $y = g(x), 0 \le x \le \infty$ の積分；有限ではない無限区間の積分であっても $\int_0^\infty g(x)dx$ の近似値を $y = 1/(x+1)$ とおけば、

$$\int_0^\infty g(x)dx = \int_0^1 h(x)dx, \quad h(y) = \frac{1}{y^2}\, g\left(\frac{1}{y} - 1\right)$$

として変換される。

2.2 逆変換法による乱数生成

分布関数 $F_X(a) = P(X \le a)$ であるから、この $x = F_X(a)$ に対する逆関数, $a = F_X^{-1}(x)$ は確率が $[0,1]$ の範囲であるから、x を $[0,1]$ のでたらめ数つまり、一様乱数を入れることで、この分布関数をもつ確率変数 X の乱数を生成できる。正規分布の逆関数は関数式では表せないが、平均や分散をもたないコーシー分布は $F_X(a) = \int_{-\infty}^{a} \frac{1}{\pi(1+x^2)}\, dx = \arctan(a)$ であるから、一様乱数 U から $\arctan(U)$ でコーシー分布の乱数をつくれる。

2.3 簡単な正規乱数の作成法

正規乱数の生成 平均 0，分散 1 の標準正規分布 $\mathrm{N}(0,1)$ にしたがう乱数をつくる。これには，12 個の一様乱数を足し合わせて，これから 6 を引けば，1 個の正規乱数が作れる。至極かんたんな方法である。なぜなら、中心極限定理の近似をもちいたもの。

中心極限定理 平均 μ, 分散 σ^2 をもつ独立同一分布の確率変数は，n が大きいとき，標本平均 $\overline{X_n}$ を基準化して $\dfrac{\overline{X_n} - \mu}{\sqrt{\sigma^2/n}} \sim N(0,1)$ に近づく。

乱数の生成にはこの左辺の式が $n = 12$ で $\mu = 1/2$, 分散 $\sigma^2 = 1/12$ より $X_1 + \cdots + X_{12} - 6$ に等しいから。$n = 12$ が小さいと思われるかも知れないが、理論的に確率変数の和を計算すると、13 次の多項式であり、正規分布の密度関数式に極めて近似がよい。

逆変換法による乱数の生成 $P[X = x_j] = p_j, j = 0, 1, 2 \cdots, \sum_j p_j = 1$

$$U \sim (0,1) \text{ 上の一様分布} \Longleftrightarrow X = x_j \quad \text{if } \sum_{i=1}^{j-1} p_i \le U \le \sum_{i=1}^{j} p_i$$

幾何分布に従う乱数の例： $P[X=i] = pq^{i-1}, i \ge 1, q := 1-p$ で $X = j \Longleftrightarrow 1 - q^{j-1} \le U < 1 - q^j \Longleftrightarrow q^j < 1 - U \le q^{j-1}$. パラメータ n, p の 2 項乱数の例： Step 1: U の一様乱数の生成. Step 2: $c := p/(1-p), i := 0, pr := (1-p)^n, F := pr$. Step 3: If $U < F$, then $X := i$ and Stop. Step 4: $pr := [c(n-i)/(i+1)]pr, F := F + pr, i := i+1$. Step 5: Goto Step 3. なぜなら $P[X=i] = \binom{n}{i} p^i (1-p)^{n-i}, \quad i = 0, 1, 2, \cdots, n$ について $P[X = i+1] = \dfrac{n-1}{i+1}\dfrac{p}{1-p} P[X=i]$ が成り立つから。

棄却採択法による乱数の生成 ある確率密度 $\{q_0, q_1, q_2, \cdots\}$ が与えられていたとき、これをもちいて $P[X = j] = p_j, j = 0, 1, 2, \cdots$ となる確率変数 X をつぎのアルゴリズムで生成できる。

Step 1: Y を確率密度 q_j で生成する。

Step 2: 一様乱数 U を生成。

Step 3: If $U < \dfrac{p_Y}{c q_Y}$, then $X = Y$ and stop, otherwise return to Step 1.

なぜなら，条件付確率の定義から

$$P[\{Y=j\} \cap \{accepted\}] = P[Y=j] P[accepted \,|\, Y=j] = q_j \frac{p_j}{c\, q_j} = \frac{p_j}{c}$$

これを $j = 0, 1, 2, \cdots$ にわたって加え合わせると，$P[accepted] = \sum_j \dfrac{p_j}{c} = \dfrac{1}{c}$。

さらに

$$\begin{aligned} P[X=j] &= \textstyle\sum_n P[\text{値 } j \text{ が } n \text{ 回目で } accept] \\ &= \textstyle\sum_n P[\text{値 } j \text{ が } n-1 \text{ 回目まで } not\, accept, n \text{ 回目で } accept] \\ &= \textstyle\sum_n \left(1 - \dfrac{1}{c}\right)^{n-1} \dfrac{p_j}{c} \\ &= p_j \end{aligned}$$

例題

確率 q_j	.1	.1	.1	.1	.1	.1	.1	.1	.1	.1
Y の値 j	1	2	3	4	5	6	7	8	9	10
X の値 j	1	2	3	4	5	6	7	8	9	10
確率 p_j	.11	.12	.09	.08	.12	.10	.09	.09	.10	.10

$c = \max_j \dfrac{p_j}{q_j} = 1.2$ と選び，Step 1: 一様乱数 U_1 から，$Y = Int[10U_1] + 1$ とおく（Int は整数を返す関数）。Step 2: 2 番目の一様乱数 U_2 をつくる。Step 3: If $U_2 \le \dfrac{p_Y}{0.12}$, then put X:=Y and stop. Otherwise return to Step 1.