

Генерисање случајних бројева и тестирање случајности низова

1.час

1. март 2016.

Случајни бројеви су низови бројева који се не померају никаквом обрасцу. Најчешће се јављају у природним појавама и ситуацијама које је немогуће предвидети.

Статистичка дефиниција случајних бројева: Случајна променљива Z са униформном расподелом на интервалу $(0,1)$ назива се **случајан број**.

Данас постоје три начина генерисања случајних бројева:

- ❶ Таблице – мана таблица је гламазност и ограниченост. Некада су се куповале таблице са случајним бројевима. Овај начин се данас најмање користи.
- ❷ Случајни процеси у природи (нпр. шум у електронској справи)
- ❸ Компјутерски генерисани (псеудо)случајни бројеви

Генератори (псеудо)случајних бројева су алгоритми који на основу почетних услова дају излазни низ псеудослучајних бројева.

Пожељна својстава која се очекују од генерисаног низа су:

- 1 Могућност појаве једног броја треба бити иста за све бројеве у простору у коме се генеришу низови.
- 2 Корелација између генерисаних бројева подниза не сме да постоји, односно ниједан подниз не сме бити зависан од неког другог подниза.
- 3 Генератор је добро конструисан уколико је број стања кроз која он пролази прилично велики. Уколико се генератор у једном тренутку нађе у стању у којем је већ био, долази до понављања неког већ генерисаног подниза. Период доброг генератора мора бити што је могуће већи.

Линеарни конгруентни генератор

Метода линеарног конгруентног генератора (LCG – linear congruential generator) за генерисање случајних бројева заснована је на рекурентној формули:

$$X_{n+1} = aX_n + c \pmod{m},$$

где су a и c константе, X_0 почетни елемент овако насталог низа бројева и m горња граница генерисаних бројева. За број m се користе степени броја 2 или прости бројеви.

Ови параметри морају да задовоље следеће захтеве:

- m и c морају бити узајамно прости
- ако је број m дељив са 4 онда је и број $a - 1$ дељив са 4
- m мора бити облика bp , за бинарне генераторе $b = 2$, а за децималне $b = 10$
- сваки прост делилац броја p мора бити делилац и броја $a - 1$

```
lcg <- function(a,c,m,x1,n) {  
  x <- vector()  
  length(x) <- n  
  x[1]<-x1  
  for(i in 1:n) {  
    x[i+1] <- (x[i]*a+c)%m  
  }  
  return (x)  
}
```

```
> lcg(9,39,1000,742,30)
```

```
[1] 742 717 492 467 242 217 992 967 742 717 492 467 242 217 992 967 742 717 492 467 242  
217 992 967 742 717 492 467 242 217 992
```

```
> lcg (113,23,10000000,721,30)
```

```
[1] 721 81496 9209071 625046 630221 1214996 7294571 4286546 4379721 4908496 4660071  
6588046 4449221 2761996 2105571 7929546 6038721 2375496 8431071 2711046 6348221 7348996  
436571 9332546 4577721 7282496 2922071 194046 1927221 7775996 8687571
```

Лехмеров генератор

Лехмеров генератор псеудослучајних бројева представља посебну верзију линеарних конгруентних генератора. Општа формула је:

$$X_{n+1} = aX_n, \pmod{m},$$

где је a цео број из интервала $[2, m - 1]$ и m прост број.

Лехмеров алгоритам је сличан ЛКГ генератору. Својства која су карактеристична само за Лехмеров алгоритам су:

- У Лехмеровом алгоритму почетни члан X_0 и модул m морају бити узајамно прости бројеви, што није случај за ЛКГ
- Највећи период Лехмеровог алгоритма је $m - 1$
- m мора бити прост број

```
Lg <- function(a,m,Xo,n){  
  x <- vector()  
  x[1]<-Xo  
  for(i in 2:n){  
    x[i]<-(a*x[i-1])%m  
  }  
  return (x)  
}
```

```
> Lg(5,7,6,30)
```

```
[1] 6 2 3 1 5 4 6 2 3 1 5 4 6 2 3 1 5 4 6 2 3 1 5 4
```

Оваквим избором параметара добијамо низ бројева који се понавља, те низ више није случајан него постаје периодичан.

За вредности параметара $a = 7^5 = 16807$, $c = 0$ и $m = 2^{31} - 1 = 2147483647$ добијамо случани низ бројева који пролази многе стандардне статистичке тестове и успешно се користи у многим апликацијама.

```
> Lg(16807,2147483647,243954,30)
```

```
[1] 243954 1952651231 364145963 2020289838 1147364549 1500308630 2081644983 1551136004  
1638828295 161897643 151905152 1859317028 1506742099 687292469 9989270 385936424  
1032864228 1238761295 7127400 1678611215 926019866 801898053 2040691846 446529485  
1513191777 1712848265 822501820 435853001 314667890 1518488316
```

Фибоначијев генератор

Фибоначијев генератор (LFG – Lagged Fibonacci generator) је добио име због сличности формуле са рекурзивном формулом којом се добијају бројеви Фибоначијевог низа. Општи облик формуле за добијање бројева на овакав начин гласи:

$$X_n = X_{n-l} + X_{n-k} \pmod{m},$$

где су l и k цели бројеви, такви да важи $l > k > 0$. За овај генератор потребно је l почетних бројева, а тестови показују да вредност броја l треба да буде већа од 16.

Овакви генератори су у неким својствима значајно бољи од линеарних конгруетних генератора. Период низа бројева добијених овако није ограничен бројем m . Проблем понављања нижих битова се може појавити код малих вредности бројева l и k , али ово се може лако избећи. И овај поступак захтева мало рачунања, те узима мало процесорског времена и захтева памћење задњих l вредности. Битно је нагласити да барем један од тих l бројева на самом почетку мора бити непаран.


```

lfg <- function(x, k, m, n) {
  y <- vector()
  length(y) <- n
  for(i in 1:17) {
    y[i] <- x[i]
  }
  for(i in 17:n) {
    y[i] <- (y[i-16] + y[i-k]) % m
  }
  return (y)
}

```

```
> lfg(a,10,1000,30)
```

```
[1] 73 696 303 766 933 436 963 706 393 576 223 46 453 116 83 786 36 402 696 342 156 482
416 822 476 362 259 448 149 458
```

```
> lfg(c,14,10000000,30)
```

```
[1] 721 81496 9209071 625046 630221 1214996 7294571 4286546 4379721 4908496 4660071
6588046 4449221 2761996 2105571 7929546 9209792 706542 9839292 1840042 7924792 5501542
1674292 9195042 9039792 1496542 9109292 9350042 6554792 691542
```

Комбиноване методе

- Wichman-Hill

$$X_t = 171X_{t-1} \bmod m_1 \quad (m_1 = 30269)$$

$$Y_t = 172Y_{t-1} \bmod m_2 \quad (m_2 = 30307)$$

$$Z_t = 170Z_{t-1} \bmod m_3 \quad (m_3 = 30223)$$

Комбиновани генератор случајних бројева је:

$$U_t = \frac{X_t}{m_1} + \frac{Y_t}{m_2} + \frac{Z_t}{m_3} \pmod{1}.$$

- Combined multiple-recursive

$$X_t = 1403580X_{t-2} - 810728X_{t-3} \bmod m_1 \quad (m_1 = 2^{32} - 209)$$

$$Y_t = 527612Y_{t-1} - 1370589Y_{t-3} \bmod m_2 \quad (m_2 = 2^{32} - 22853)$$

Комбиновани генератор случајних бројева је:

$$U_t = \begin{cases} \frac{X_t - Y_t + m_1}{m_1 + 1}, & \text{ако је } X_t \leq Y_t \\ \frac{X_t - Y_t}{m_1 + 1}, & \text{ако је } X_t > Y_t \end{cases}$$

Тестирање случајних низова

Немогуће је дефинитивно утврдити случајност низова, али их можемо подвргнути неким статистичким тестовима. Ако прођу већину тих тестова, повећава се шанса случајности датих низова, односно знамо да је генератор који их је генерисао бољи. То су **тестови случајности**. Уколико низови не прођу тест случајности (или више њих), онда се неки параметри могу променити или треба да користимо друге случајне податке који ће задовољавати дате услове и проћи тест.

Тестирање случајности низа бројева обухвата:

- постављање хипотезе о некој особини случајне променљиве,
- утврђивање потребног и довољног интервала поверења и
- проверу хипотезе.

Хипотезе дефинишемо на следећи начин:

- H_0 – низ бројева јесте случајан
- H_1 – низ бројева није случајан

У зависности од резултата и унапред задатог интервал поверења хипотеза се прихвата или одбацује.

Тест тачака заокрета

Тест статистика је:

$$Y_n = Y_1 + Y_2 + \dots + Y_{n-1},$$

где је $Y_j = 1$ ако је $(X_{j-1} < X_j \wedge X_j > X_{j+1})$ или $(X_{j-1} > X_j \wedge X_j < X_{j+1})$, иначе $Y_j = 0$.

За велики обим узорка ($n > 50$) тест статистика Z_n^* има приближно стандардну нормалну расподелу, где је

$$Z_n^* = \frac{Z_n - \frac{2(n-2)}{3}}{\sqrt{\frac{16n-29}{90}}}.$$

Тест тачака заокрета

```
testzaokreta<-function(x,alfa){
n=length(x)
s=0
for(i in 1:(n-2)){
a = (x[i + 1]-x[i])*(x[i + 2]-x[i + 1])
if (a < 0) s = s + 1
}
m = (2 * (n-2))/3
dz = (16*n-29)/90
cat("srednja vrednost :",m," uzoracka disperzija: ", dz, "\n")
print("vrednost test statistike je:")
print(s)
print(" standardizovana vrednost test statistike je:")
print((s-m)/sqrt(dz))
c=qnorm(p=(1-(alfa/2)))
cat((1-alfa) * 100," \%-tni interaval poverenja za stand stat je (",-c," ,",c,")" )
}
```

Тест тачака раста

Тест статистика је:

$$R_n = Y_1 + Y_2 + \dots + Y_n,$$

где је $Y_j = 1$ ако је $(X_{j-1} < X_j)$, $Y_j = 0$ ако је $(X_{j-1} \geq X_j)$.

R_n је статистика која представља бројач тачака раста. Погодност статистике R_n је брза конвергенција ка нормалној расподели ($n > 12$).

Тест тачака раста

```
testrasta <- function(x, alfa) {  
  n<-length(x)  
  s<-0  
  for (i in 1:(n-1)) {  
    a<- (x[i+1]-x[i])  
    if (a>0) s<-s+1  
  }  
  m<-(n-1)/2  
  dz<-(n+1)/12  
  cat("srednja vrednost:", m," uzoracka disperzija: ", dz, "\n")  
  print("Vrednost test statistike je:")  
  print(s)  
  print("Stantardizovana vrednost test statistike je:")  
  print((s-m)/sqrt(dz))  
  c<-qnorm(p=(1-(alfa/2)))  
  cat((1-alfa)*100,"%-tni interval poverenja za stand stat je (",-c,"",c,")\n" )  
}
```

Тест разлика рангова

Тест статистика је:

$$Z_n = Y_1 + Y_2 + \dots + Y_{n-1},$$

где је $Y_j = |R_{j+1} - R_j|$, а $(R_1, \dots, R_{n-1})$ је низ рангова низа случајних бројева $(X_1, \dots, X_n)$. Z_n^* има приближно стандардну нормалну расподелу,

где је

$$Z_n^* = \frac{Z_n - \frac{(n-1)(n+1)}{3}}{\sqrt{\frac{(n-2)(n+1)(4n-7)}{90}}}.$$

Тест разлика рангова

```
testrazlikarangova <- function(x, alfa) {  
  n<-length(x)  
  R<-rank(x) #niz rangova  
  s<-0  
  for(i in 1:(n-1))  
  {  
    s<-s+abs(R[i+1]-R[i])  
  }  
  print("Vrednost test statistike je:")  
  print(s)  
  m<-((n-1)*(n+1))/3  
  ds<-((n-2)*(n+1)*(4*n-7))/90  
  cat("srednja vrednost:", m," disperzija: ", ds, "\n")  
  print("Stantardizovana vrednost test statistike je:")  
  print((s-m)/sqrt(ds))  
  c<-qnorm(p=(1-(alfa/2)))  
  cat((1-alfa)*100,"%-tni interval poverenja za stand stat je (",-c,"","c,")\n" )  
}
```

Бартелсов тест

Тест статистика је:

$$B_n = \frac{12Z_n}{n(n-1)},$$

где је $Z_n = Y_1 + Y_2 + \dots + Y_{n-1}$, а сада немамо више збир разлика рангова, већ збир квадрата разлика рангова:

$$Y_j = |R_{j+1} - R_j|^2,$$

где је $(R_1, R_2, \dots, R_{n-1})$ низ рангова случајних бројева $(X_1, X_2, \dots, X_n)$. B_n^* има приближно стандардну нормалну расподелу, где је

$$B_n^* = \frac{B_n - 2}{\sqrt{\frac{4}{n}}}.$$

Бартелсов тест

```
testBartels <- function(x, alfa) {  
  n<-length(x)  
  R<-rank(x)  
  s<-0  
  for(i in 1:(n-1))  
  {  
    s<-s+(R[i+1]-R[i])^2  
  }  
  bn<-(s*12)/(n*(n^2-1))  
  print("Vrednost test statistike je:")  
  print(bn)  
  m<-2  
  dbn<-4/n  
  cat("srednja vrednost:", m," disperzija: ", dbn, "\n")  
  print("Stantardizovana vrednost test statistike je:")  
  print((bn-m)/sqrt(dbn))  
  c<-qnorm(p=(1-(alfa/2)))  
  cat((1-alfa)*100,"%-tni interval poverenja za stand stat je (",-c,"",c,"")\n" )  
}
```

Задаци

1. Претходне кодове за тестирање случајности изменити тако да свака од функција одређује p вредност теста.
2. Генерисати низ од 10000 (псеудо)случајних бројева и применити претходне тестове.
3. Применити тестове на други низ који се добија на следећи начин: ако је $x_1, \dots, x_{1000}$ низ (псеудо)случајних бројева са $U(0, 1)$ расподелом, новодобијени низ ће бити $y_1, y_2, \dots, y_{999}$, где је $y_i = x_i + x_{i+1}$, $i = 1, 2, \dots, 999$. Дискутовати резултате!