STAT 2005 – Programming Languages for Statistics

Tutorial 2 Random Numbers

2020

LIU Ran

Department of Statistics, The Chinese University of Hong Kong

1 Generate Random Numbers

1.1 Random Seed

- The pseudo random number generator in R requires a starting value, which is called 'random seed' or 'seed'.
- Same seed, Same command $\Rightarrow$ Same random numbers

Remark 1.1. Set seed in the first line of your model simulation study in order to get reproducible result.

Remark 1.2. Usually, we set an ordinary number to be the seed.

1.2 Random Sample

sample(x, size, replace = FALSE, prob = NULL)

```
> set.seed(2005)
> sample(10)
 [1]  4  1  5  7 10  2  9  3  8  6
> sample(10)
 [1]  7  5  1  9  2  6  3  8  4 10

> set.seed(2005)
> sample(10)
 [1]  4  1  5  7 10  2  9  3  8  6

> set.seed(2005)
> a = 1 # non-random process will not affect the seed
> sample(10)
 [1]  4  1  5  7 10  2  9  3  8  6
```

For different parameters:

```
> set.seed(2005)
> sample(10)
 [1]  4  1  5  7 10  2  9  3  8  6
> sample(5)
[1] 5 1 4 2 3

> set.seed(2005)
> sample(5)
[1] 4 1 5 3 2
> sample(10)
 [1]  4  2  5  7  3  8  1 10  9  6
```

For different types:

```
> set.seed(2005)
> runif(3)
[1] 0.83248656 0.08690543 0.89776514
> sample(10)
[1]  1  5  7  8  4  2 10  3  9  6

> set.seed(2005)
```

```
> sample(10)
[1]  4  1  5  7 10  2  9  3  8  6
> runif(3)
[1] 0.006695024 0.599185211 0.826918751
```

An exception? (optional and no need to be remembered)

```
> set.seed(2005)
> sample(5)
[1] 4 1 5 3 2

> set.seed(2005)
> runif(1)
[1] 0.8324866
> sample(5)
[1] 4 1 5 3 2

> set.seed(2005)
> runif(1)
[1] 0.8324866
> runif(1)
[1] 0.08690543
> sample(5)
[1] 4 1 5 3 2
```

1.3 Random Sample From Commonly Used Distribution

Distribution	R name	Additional arguments = default value
Uniform	`unif`	`min=0, max=1`
Binomial	`binom`	`size, prob`
Exponential	`exp`	`rate=1`
Poisson	`pois`	`lambda`
Normal	`norm`	`mean=0, sd=1`

Remark 1.3. Take STAT2001 or google for them if you are not familiar with these distributions.

1.3.1 `r`+R Name: Random Numbers

```
> set.seed(2005)
> x = rnorm(1000)
> c(M=mean(x), V=var(x))
           M            V
0.005746068 0.901851609
```

1.4 `d`+R Name: Density (pdf/pmf)

```
> dunif(0.3, 0, 2)
[1] 0.5
> dbinom(1, size = 5, prob = 0.7)
[1] 0.02835
# choose(n, k) Number of combinations
> choose(5, 1)*(0.3)^4*(0.7)^1
[1] 0.02835
```

1.5 `p`+R name: cdf

```
> pnorm(0) # P(X<=0)
[1] 0.5
> pbinom(3, size = 5, prob = 0.7)   # P(X<=3)
[1] 0.47178
```

```
> index = c(0,1,2,3)
> sum(choose(5, index)*(0.3)^(5-index)*(0.7)^index)
[1] 0.47178
```

1.6 q+R name: Quantiles

```
> qnorm(0.5) # P(X<=0) = 0.5
[1] 0
> qbinom(0.17, size = 5, prob = 0.7) # P(X<=3) >= 0.17
[1] 3
> qbinom(0.16, size = 5, prob = 0.7) # P(X<=2) >= 0.16
[1] 2
> pbinom(seq(0,5), size = 5, prob = 0.7) # P(X<=seq(0,5))
[1] 0.00243 0.03078 0.16308 0.47178 0.83193 1.00000
```

Remark 1.4. It returns $\min_{q \in I}\{q : P(X \leq q) \geq \alpha\}$, where α is the probability you input, I is the distribution support, q is the quantile you want to search.

2 apply

- **apply**: Use one function to all elements in one(some) dimension(s) of a matrix(array).

```
> (y = matrix(rep(1:3,rep(5,3)), nrow=5))
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    1    2    3
[3,]    1    2    3
[4,]    1    2    3
[5,]    1    2    3
> apply(y, 1, mean) # operate on each row
[1] 2 2 2 2 2
> apply(y, 2, mean) # operate on each column
[1] 1 2 3
```

```
> (z = array(rep(seq(20, by=-2, length=12), 2), dim=c(3, 4, 2)))
, , 1

     [,1] [,2] [,3] [,4]
[1,]   20   14    8    2
[2,]   18   12    6    0
[3,]   16   10    4   -2

, , 2

     [,1] [,2] [,3] [,4]
[1,]   20   14    8    2
[2,]   18   12    6    0
[3,]   16   10    4   -2

> apply(z, c(1, 3), mean)  # one dimension left
     [,1] [,2]
[1,]   11   11
[2,]    9    9
[3,]    7    7

> z[1,,1] # each element is the result that 'mean' operates on a vector
[1] 20 14  8  2

> c(mean(z[1,,1]),mean(z[2,,1]),mean(z[3,,1]))
[1] 11  9  7

> c(mean(z[1,,2]),mean(z[2,,2]),mean(z[3,,2]))
[1] 11  9  7
```

```
> apply(z, 1, mean) # two dimensions left
[1] 11  9  7

> z[1,,] # each element is the result that 'mean' operates on a matrix
     [,1] [,2]
[1,]   20   20
[2,]   14   14
[3,]    8    8
[4,]    2    2

> c(mean(z[1,,]),mean(z[2,,]),mean(z[3,,]))
[1] 11  9  7
```

Remark 2.1. `MARGIN` argument: Dimensions you want to keep.

3 Monte Carlo Method

3.1 Main theory

The main theoretical basis for Monte Carlo Method is the Law Of Large Numbers(i.i.d):

$$\frac{1}{n}\sum_{i=1}^{n} X_i \xrightarrow{p} E(X) = \int xp(x)dx.$$

And the central limit theorem could give the approximate confidence intervals for $\mu = E(X)$:

$$\sqrt{n}\frac{(\bar{X}_n - \mu)}{\sigma} \xrightarrow{d} N(0,1)$$

Usually, we use the Standard Deviation(SD) of samples to replace σ. ($\sigma = sd(X)$)

Remark 3.1. Standard Deviation is different from Standard Error:

$$\text{SD: } \sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2} \qquad \text{SE: } \sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}}$$

Remark 3.2. More rigorous version for LLN and CLT, please go to the wiki.

3.2 Monte Carlo for integral

Suppose we can draw samples from the distribution $p(x)$ defined on [a,b], and want to calculate the following integral:

$$\int_a^b f(x)dx = \int_a^b \frac{f(x)}{p(x)}p(x)dx = E\left(\frac{f(X)}{p(X)}\right).$$

So, the Monte Carlo estimation for the integral is

$$E\left(\frac{f(X)}{p(X)}\right) \approx \frac{1}{n}\sum_{i=1}^{n}\left(\frac{f(x_i)}{p(x_i)}\right),$$

where x_i is i.i.d sample from $p(x)$.

Exercise 3.1. Calculate the integral $\int_0^2 cos(x)dx$:

```
> set.seed(2005)
> x = runif(1000000, min = 0, max = 2)
> mean(cos(x)*2) # p(x) = 1/2
[1] 0.9093535
> sin(2) - sin(0)
[1] 0.9092974
```

4 Exercises

1. Toss a biased coin for 10 times, with $\mathsf{P}(\text{Head}) = 0.7$. Simulate the result.
2. Generate a 3×4 matrix of uniform random values between 5 and 9 and find the average of each column.
3. Suppose $Z \sim \mathrm{N}(0, 1)$. Find constant c s.t.
   (a) $\mathsf{P}(Z \leq c) = 0.1151$;
   (b) $\mathsf{P}(1 < Z \leq c) = 0.1525$;
   (c) $\mathsf{P}(-c < Z \leq c) = 0.8164$.