Simulation

Programming in R for Data Science

Anders Stockmarr, Kasper Kristensen, Anders Nielsen

What can random numbers be used for

Some applications:

- Simulation of complex systems
- MCMC (Markov Chain Monte Carlo) methods
- Encryption
- Bootstrap methods
- Design of experiments
- performance testing
- Simulation testing of models and estimation procedures
- Other areas where producing an unpredictable result is desirable

How do we get random numbers?

1. Use a natural phenomena, E.g: A dice, a deck of cards, a Geiger counter, ...
2. Use a table of random numbers
3. Computer algorithm: Easy, but the algorithm should be carefully selected!
   - Fast
   - Simple
   - Reproducible
   - Numbers must be uncorrelated, evenly distributed, and have a long period.

Item 3 is included in **R** . It is easy to simulate from most distributions of interest.

The Mersenne Twister

- Random numbers in R are actually so-called *pseudo-random* numbers.
- Generated from an algorithm;
- for all practical purposes, pseudo-random numbers behave like true random numbers.
- By specifying the input to the algorithm, pseudo-random numbers can be RE-CREATED.

Default Pseudo-Random Number Generator in R: The *Mersenne Twister*.

- Period: $2^{19937} - 1$ (a Mersenne prime). A very long period.
- The most commonly used PRNG;
- also used as default in ao. *Python*, *Maple*, *MATLAB*, *Julia*, *STATA* and *Microsoft Visual C++*.
- invoked by default when you use `rnorm()`, `runif()`, `rgamma()` etc.

Controlling the input: set.seed()

The Mersenne Twister uses a *random seed* as input: Constructed from time and session ID.

- You can replace the random seed with a fixed seed with the `set.seed()` function.
- Two different sets of random numbers:

```
> set.seed(123)
> rnorm(3)
[1] -0.5604756 -0.2301775  1.5587083
> set.seed(456)
> rnorm(3)
[1] -1.3435214  0.6217756  0.8008747
```

- The first set of pseudo-random numbers regenerated:

```
> set.seed(123)
> rnorm(3)
[1] -0.5604756 -0.2301775  1.5587083
```

- Reasons for setting the seed:
  - comparisons;
  - code control;
  - reproducibility.
  - if none af the above applies: DON'T.

A few examples

Simulation from a uniform distribution

```
> runif(5,min=1,max=2)
[1] 1.528105 1.892419 1.551435 1.456615 1.956833
```

Simulation from a normal distribution

```
> rnorm(5,mean=2,sd=1)
[1] 0.4269356 0.8050889 2.9549471 1.2955381 0.4238912
```

Simulation from a gamma distribution

```
> rgamma(5,shape=2,rate=1)
[1] 1.6844459 0.1576027 0.5138852 2.6631387 0.7853806
```

Simulation from a binomial distribution

```
> rbinom(5,size=100,prob=.3)
[1] 31 32 26 38 33
```

Simulation from a multinomial distribution

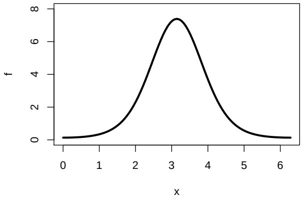
```
> rmultinom(5,size=100,prob=c(.3,.2,.5))
     [,1] [,2] [,3] [,4] [,5]
[1,]   35   33   27   31   31
[2,]   22   17   17   20   14
[3,]   43   50   56   49   55
```

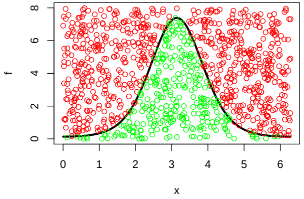
Monte Carlo integration (a miniature example).

- Consider the function $f(x) = e^{2\cos(x-\pi)}$
- Calculate the integral

$$\int_0^{2\pi} f(x)dx$$

- A Monte Carlo method is:
  1. Simulate uniformly distributed points $(x_1, y_1), \ldots, (x_N, y_N)$ in $[0, 2\pi] \times [0, 8]$.
  2. Estimate the probability of points being below the curve f via:
  $$\widehat{P}_u = \frac{\#\text{below}}{N}$$
  3. The area of the entire "box" is 16π, so the estimated integral, the area under the curve, becomes
  $$\int_0^{2\pi} f(x)dx \approx 16\pi \cdot \widehat{P}_u$$





Practical Implementation

```
> x<-runif(1000000,0,2*pi)
> y<-runif(1000000,0,8)
> phat.under<-mean(y<exp(2*cos(x-pi)))
> phat.under*16*pi
[1] 14.29143
```

Verifying via integrate():

```
> f<-function(x){exp(2*cos(x-pi))}
> integrate(f,0,2*pi)$value
[1] 14.32306
```

The standard uncertainty of this Monte Carlo integration can be shown to be ± 0.05.

Simulation study of the binomial estimator - rbinom()

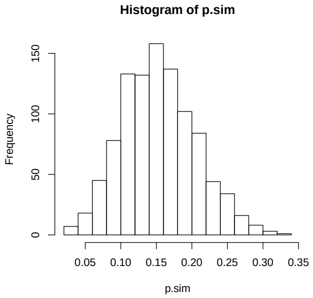
- Now, consider the experiment where data `x` are binomially distributed. Say, the number of 6's in n die rolls: $x \sim Bin(n, p)$.
- We know that our estimate of p is $\hat{p} = x/n$. If the die is fair, $\hat{p}$ should be close to $1/6$. How close?
- We can try to simulate and (re)-estimate $\hat{p}$ a few times:

```
> x <- rbinom(1,50,1/6)
> phat <- x/50
> phat

[1] 0.22

> x <- rbinom(1,50,1/6)
> phat <- x/50
> phat

[1] 0.22

> x <- rbinom(1,50,1/6)
> phat <- x/50
> phat

[1] 0.26
```

Simulation study of the binomial estimator - rbinom()

- To repeat it 1000 times we can do:

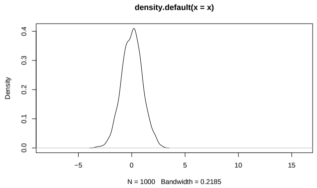
```
> doone <- function(){
+   x <- rbinom(1,50,1/6)
+   p <- x/50
+   p
+ }
> p.sim<-replicate(1000,doone())
> hist(p.sim,breaks=15)
```



- Note that 1/6=`0.1666667`.

Simulated normally distributed data - rnorm()

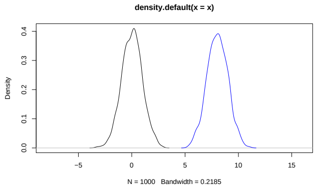
```
> x<-rnorm(1000)
> plot(density(x),xlim=c(-8,16))
```



The numbers distribute themselves around 0.

Simulated normally distributed data - rnorm()

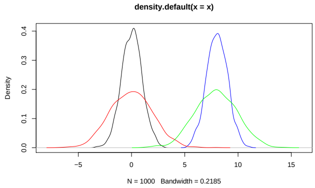
```
y<-rnorm(1000,mean=8)
lines(density(y),col="blue")
```



The numbers distribute themselves around 0 and 8.

Simulated normally distributed data - rnorm()

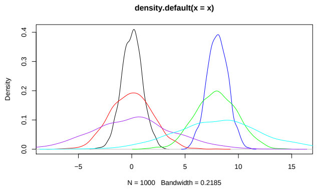
```
lines(density(rnorm(1000,sd=2)),col="red")
lines(density(rnorm(1000,mean=8,sd=2)),col="green")
```



The numbers distribute themselves around 0 and 8, with increased variation.

Simulated normally distributed data - rnorm()

```
lines(density(rnorm(1000,sd=4)),col="purple")
lines(density(rnorm(1000,mean=8,sd=4)),col="cyan")
```



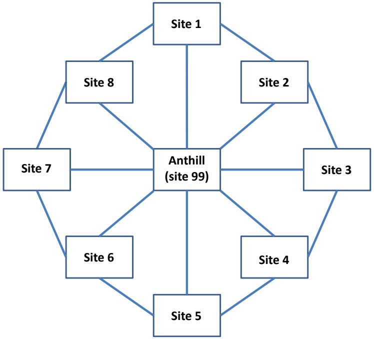
The numbers distribute themselves around 0 and 8, with even more increased variation.

Simulation of complex systems: Example

- David Vose: "Why simulate when you can calculate?"
- sometimes it can be very hard to calculate.

Ants collecting food:

- a number of ants (say, 3) are at the anthill;
- Ants move randomly to sites to collect food (1 time unit);
- Poisson-simulated daily food units at sites (here 8 sites);
- If food is present, ants load it and move to anthill (1 time unit);
- if no food is present, ants move upwards or downwards within sites, with probability 0.5 (1 time unit);
- if all sites have been visited with no food to be found, the ant returns to the anthill and is done (1 time unit).



How many time units does it take to collect the food in a day?

Ants collecting food: Initialization

9 objects to keep track of. We assign initial values, and write them to a text file for later use:

```
> my.text<-"time.ants<-0
+           at.anthill<-rep(TRUE,3)
+           done.all<-0
+           done<-rep(0,3)
+           food<-c(rpois(8,lambda=10),0,0)
+           site<-c(10,10,10)
+           carry<-c(0,0,0)
+           visited.sites<-list(numeric(0),numeric(0),numeric(0))
+           total.visited.sites<-rep(0,3)"
> writeLines(my.text,con="Data/initialize.txt")
```

Ants collecting food: Loop

We write the loop code in a file for later use:

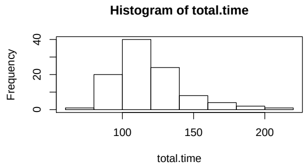
```
> my.text2<-"
+ while(!done.all){
+   time.ants<-time.ants+1
+   at.anthill<-(site==10)
+   done.all<-prod(done)
+   done<-ifelse(done,1,
+          ifelse(total.visited.sites==8,1,0))
+    for(i in 1:3){
+      carry[i]<-ifelse(food[site[i]]>0,1,0)
+      food[site[i]]<-ifelse(carry[i],food[site[i]]-1,0)
+      if(!at.anthill[i] & !carry[i]){
+      visited.sites[[i]]<-unique(c(visited.sites[[i]],site[i]))
+      total.visited.sites[i]<-length(visited.sites[[i]])}
+    }
+   site<-ifelse(done,10,
+          ifelse(carry,10,
+            ifelse(at.anthill,sample(1:8,3,replace=T),
+              site+(-1)^rbinom(3,1,0.5))))
+   site[site==9]<-1
+   site[site==0]<-8}"
> writeLines(my.text2,con="Data/loop.txt")
```

All 9 variables are updated. The loop stops when all ants are done; ie. when `done.all=1`.

Ants collecting food: Simulation

We simulate the the system by sourcing the initialization and and loop code:

```
> total.time<-numeric(100)
> for(k in 1:100){
+ source("Data/initialize.txt")
+ source("Data/loop.txt")
+ total.time[k]<-time.ants
+ }

> hist(total.time); box()
```



Further reading on distributions

Information on basic distributions: T. ex. the normal, uniform, gamma, binomial and multinomial distributions, are included in basic textbooks on statistics. One example could be

Armitage P, Berry G & Matthews JNS:
Statistical Methods in Medical Research
Blackwell Science Ltd (2001)

The mathematical exposition is modest, and while examples are from the medical world, the methods apply universally.

You can also check out your own field of expertise, for a statistics textbook with relevant examples.