

Math 1180:Lab1

Kyle Gaffney

January 8th, 2014

1 Generating Random Numbers in R

The key to simulating stochastic processes is the generation of (psuedo)random numbers. R has several different commands for generating such numbers, each using a different probability distribution. We'll use three (hopefully) familiar examples. The normal distribution, the uniform distribution and the binomial distribution.

Random numbers from a uniform distribution have an equal probability of being anything between some upper and lower bound. In R, the default min and max values are 0 and 1.

```
>runif(1)
[1] 0.1910009
>runif(2)
[1] 0.2958168 0.4420523
>runif(2,min=3,max=12)
[1] 6.741396 8.428657 10.683639 9.082789 3.261233 5.118001 8.305848
[8] 7.217871 3.710094 4.794811
```

Random numbers from a normal distribution form a 'bell curve'. We can specify a mean and standard deviation, but if we don't the defaults are 0 and 1.

```
> rnorm(10)
[1] -1.5346586 2.7816891 -0.2200338 -0.8633776 -1.3731667 -0.6271000
[7] -0.6548011 -0.1016991 -0.7594421 1.2713166
> rnorm(10,mean=7.5,sd=1.5)
[1] 7.010355 8.656743 9.784120 5.903149 7.364554 8.644147 8.823739
[8] 11.620645 7.072112 11.150432
```

The binomial distribution is discrete. When it generates random numbers, they can only be integers. The binomial distribution represents the number of successes in a series of identical independent trials that all have the same probability of success. For example, a coin flip represents a $1/2$ probability of success. Here, we have R simulate 10, single coin flips. The values can only be 0 or 1.

```
> rbinom(10,size=1,prob=.5)
[1] 1 0 0 0 0 1 1 0 1 1
```

Alternatively, we could have R simulate 10 sets of 2 coin flips.

```
> rbinom(10,size=2,prob=.5)
[1] 1 2 2 0 0 1 0 1 2 0
```

or 2 sets of 10 coin flips

```
> rbinom(2,size=10,prob=.5)
[1] 4 5
```

We could also imagine events that don't happen half the time, such as rolling a six on a regular die. Here, we simulate rolling seven dice ten times and counting the number of sixes that appear each time.

```
> rbinom(10,size=7,prob=1/6)
[1] 2 1 0 2 0 1 1 2 1 0
```

1.1 Example

Consider a Markov Chain of 2 states, state A, and state B. We wish to simulate when we are in state A vs. state B.

Suppose that every year we decide to move from state A to state B with probability 0.2 and we move from state B to state A with probability 0.1. What are the probabilities that we stay in state A or B?

Let us define a variable M . $M=0$ when we are in state A, and $M=1$ when we are in state B.

Then let us make M a vector (list) telling us what state we were in at every time. Since we start in state A, $M[1]=0$.

To update our Markov chain all we need to know at a given time step is what state we were in and our transition probabilities. Then we can generate a random number and have it tell us what state to go into next.

First we need to let R know what state we are in. We will do this using the `ifelse` command. For example:

```
ifelse(M[k]==0,pt<-0.2,pt<-0.1)
```

Here `pt` is a variable called the probability of transfer. If we are in state A ($M=0$) at time step k then `pt=0.2`, otherwise we are in state B and `pt=0.1`. Will this command work if we had more than 2 states?

What type of random number should we generate? Or phrased differently what distribution should we be pulling our random numbers from?

To pull everything together we will be using for loops and the `ifelse` command from last semester.

```

M<-0
n<-50
rn<-runif(n-1)
for (i in 2:n){ifelse(M[i-1]==0,pt<-0.2,pt<-0.1)
ifelse(rn<pt,M[i]<-1-M[i-1],M[i]<-M[i-1])}
plot(M)

```

Here `rn` is a list of random numbers that we will call upon to decide whether we stay or go. In the `for` loop the first `ifelse` statement tells us what our probability of leaving is depending on what state we are in. The second `ifelse` statement lets us know whether or not we transfer and updates our state accordingly.

2 Questions

1. For the following questions, use R's random number generators.
 - (a) Generate 10 random numbers between uniformly distributed -3 and 9.
 - (b) Generate 10 random numbers with mean 3 and standard deviation 2.
 - (c) Simulate 20 roles of 5 dice, record the number of 6's rolled each time.
2. Do problem 6.1.36. Let N_t be the population size at time t .

$$N_{t+1} = \begin{cases} N_t + 1 & \text{with probability } .5 \\ N_t & \text{with probability } .5 \end{cases} \quad (1)$$

Run 3 simulations of this for 50 generations with $N_1 = 0$. Plot the outcome of all 3 simulations on the same graph. Include the code also.

Hint: to plot multiple plots on the same graph start with a `plot` command and then add a `lines` command or a `points` command. To change the color of a plot you can add the following, `col=""` :

```
points(t,f,col='red')
```