



Caminhada aleatória: o bêbado e o abismo - Roteiro em R



Imagine um bêbado andando sempre para frente em uma enorme planície, mas que tem um abismo em um dos lados. A cada passo para frente, ele cambaleia um certo número de passos para a direção do abismo ou da planície, com igual probabilidade.

Este é um dos processos Markovianos mais simples, chamado caminhada aleatória ([random walk](#)) em uma dimensão ¹³⁹. Se o bêbado cai no abismo a caminhada acaba (e o bêbado também), uma condição que chamamos de fronteira de absorção (*absorbing boundary*).

Bebuns virtuais

O que podemos prever deste processo? Vamos soltar alguns bêbados neste mundo virtual. Para isto usaremos uma função em R, cujo código segue abaixo. Copie e cole este código na linha de comando do R.

```
bebado <- function(n=1,step=1,ciclo=1e5,cont=1e3,x1=NULL){
  if(is.null(x1)){
    x1 <- sample(1:200,n,replace=TRUE)
  }
  results <- matrix(NA,nrow=1+ciclo/cont,ncol=n)
  results[1,] <- x1
  X <- x1
  for(i in 2:(1+ciclo/cont)){
    for(j in 1:cont){
      X[X<=0] <- NA
      X <- X +sample(c(step,-1*step),n,replace=T)
    }
    results[i,] <- X
  }
}
```

```
}  
results[is.na(results)] <- 0  
time <- seq(0,ciclo,by=cont)  
matplot(time,results,type="l", col=rainbow(n),lwd=2, xlab="Passos",  
ylab="Distância do Abismo")  
abline(h=0,lwd=4)  
}
```

Esta função tem quatro argumentos:

- `n`: Número de bêbados
- `step`: número de passos para o lado que cada bêbado dá a cada instante de tempo
- `ciclo`: tempo total da simulação (no caso é o mesmo que o número de passos para frente).
- `cont`: intervalo de registro dos dados. Este argumento serve para poupar memória, já que guardar todos os passos em simulações longas pode deixar a simulação muito lenta. Mantendo o valor *default* a posição de cada bêbado é registrada a cada 1000 intervalos de tempo.

A função sorteia a posição inicial dos bêbados entre 1 e 200 passos de distância do abismo. Vamos soltar dez bêbados, que cambaleiam 10 passos a cada intervalo, por dez mil intervalos de tempo:

```
bebado(n=10,step=10,ciclo=1e4,cont=1e3)
```

Como em todo processo estocástico, os resultados variam a cada realização. Por isso repita a simulação para se assegurar que entendeu os resultados. Você pode fazer isso repetindo muitas vezes com dez bêbados, ou simplesmente aumentando o número de bêbados, já que que são independentes.

Efeito do passo

O que acontece se deixamos os bêbados um pouco menos cambaleantes? Experimente reduzir para dois os passos laterais:

```
set.seed(42) # semente de números aleatórios  
bebado(n=10,step=2,ciclo=1e4,cont=1e3)
```

Efeito do tempo

Estes bêbados que balançam menos estão menos sujeitos a terminar no abismo? Certifique-se disto aumentando o número de intervalos de tempo:

```
set.seed(42)  
bebado(n=10,step=2,ciclo=5e4,cont=1e3)
```

Perguntas

1. O bêbado tem igual probabilidade de cair para a direita e para esquerda, portanto ele anda em

linha reta, na média. Esta caminhada aleatória equiprovável com fronteira de absorção tem um único desfecho, dado tempo suficiente. Qual é?

2. O mesmo modelo pode ser aplicado à dinâmica de populações sob [estocasticidade demográfica](#). O que representaria cada um dos parâmetros da simulação neste caso?

139)

como o bêbado dá sempre um passo adiante, apenas o deslocamento lateral é aleatório, e é o que nos interessa aqui. Usamos os passos para frente como medida de tempo

From:

<http://ecovirtual.ib.usp.br/> -

Permanent link:

http://ecovirtual.ib.usp.br/doku.php?id=ecovirt:roteiro:math:bebador_old



Last update: **2016/05/10 07:19**