传染病模型模拟

周世祥

2015 年 11 月 3 日

流行病案例

SIR 模型

SIR 模型分别表示易感者 (Susceptible), 感染者 (Infected)，恢复者 (Removed). 假设任何个体均是上述三种状态之一：如果个体并未感染某种疾病，称之为易感者；如果感染某种疾病，称之为感软者；如果感染过该疾病，且已恢复或者死亡的，称之为恢复者。将时间表示为离散数据形式。在每个时间点，每个感染者可以使易感者感染致病或者本身恢复或者死亡。

令 $S(t), I(t), R(t)$ 分别表示时刻 t 的易感者，感染者和恢复者数量。在任意时刻每个感染者以概率 α 使易感者染病 (或者说每个感染者与所有的易感者有相同的接触机会，这里称之为混合假设)，在有机会感染其他人之后，每个感染者以概率 β 恢复或被移除。

假设初始条件：$S(0) = N; I(0) = 1; R(0) = 0$

注意到总的个体数量是 $N + 1$，即在所有时刻 t, 有

$$S(t) + I(t) + R(t) = N + 1$$

在每个时刻 t，易感者未被感染的概率为 $(1 - \alpha)^{I(t)}$

进一步，假设感染者在变为易感者之后失去传染能力 (失效)，因而，$S(t + 1) \sim b(S(t), (1 - \alpha)^{I(t)})$

在每个感染者以概率 β 恢复正常，即，

$$R(t + 1) \sim R(t) + b(I(t), \beta)$$

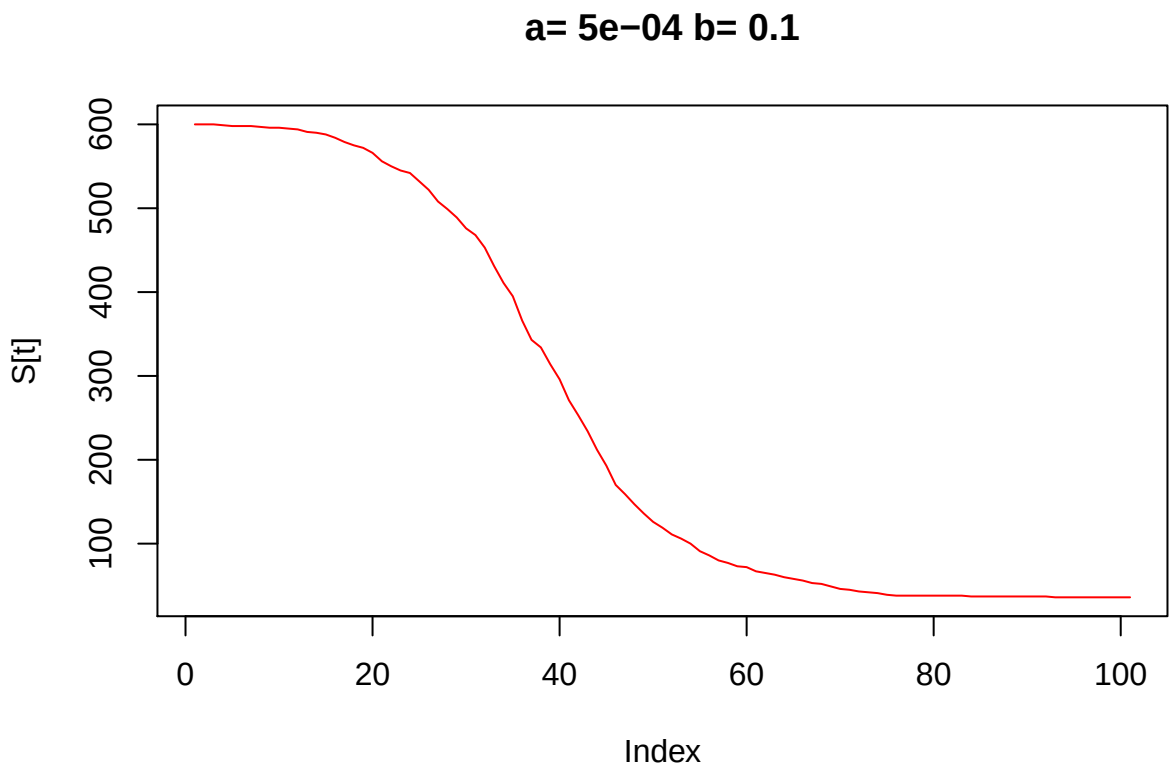
同时，

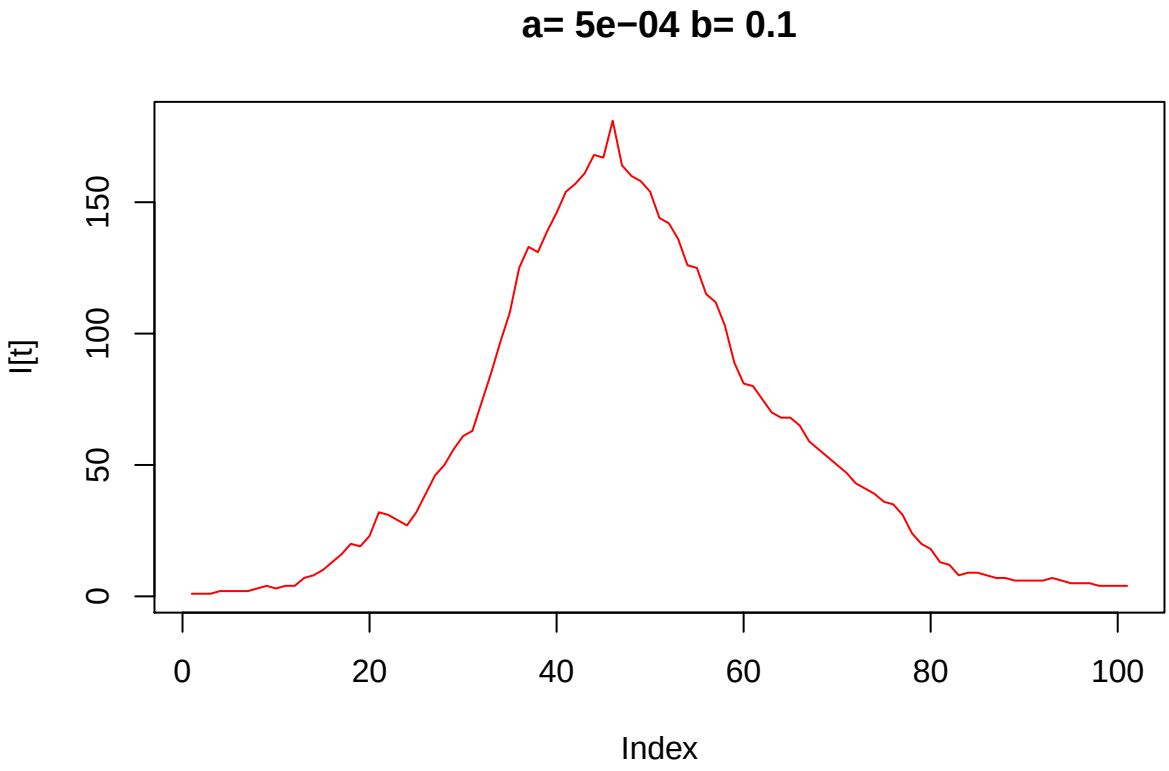
$$I(t + 1) = N + 1 - R(t + 1) - S(t + 1)$$

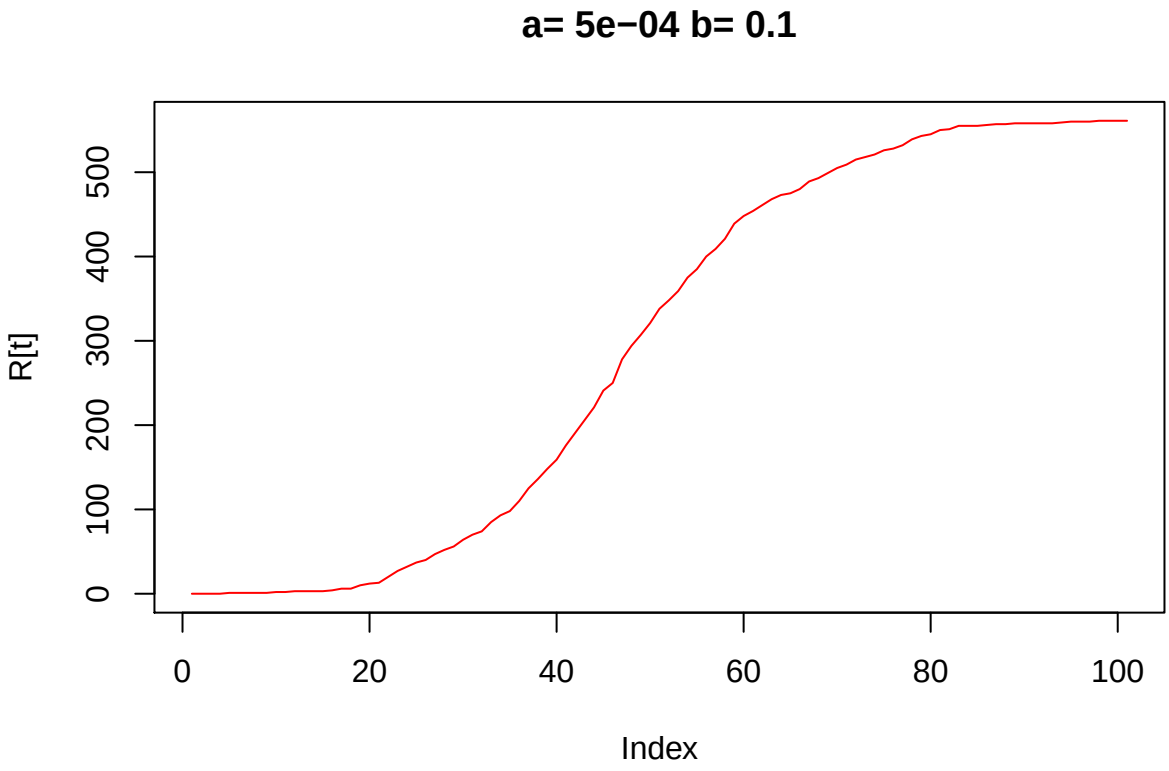
通过上述过程，一个简单的 SIR 模型如下。

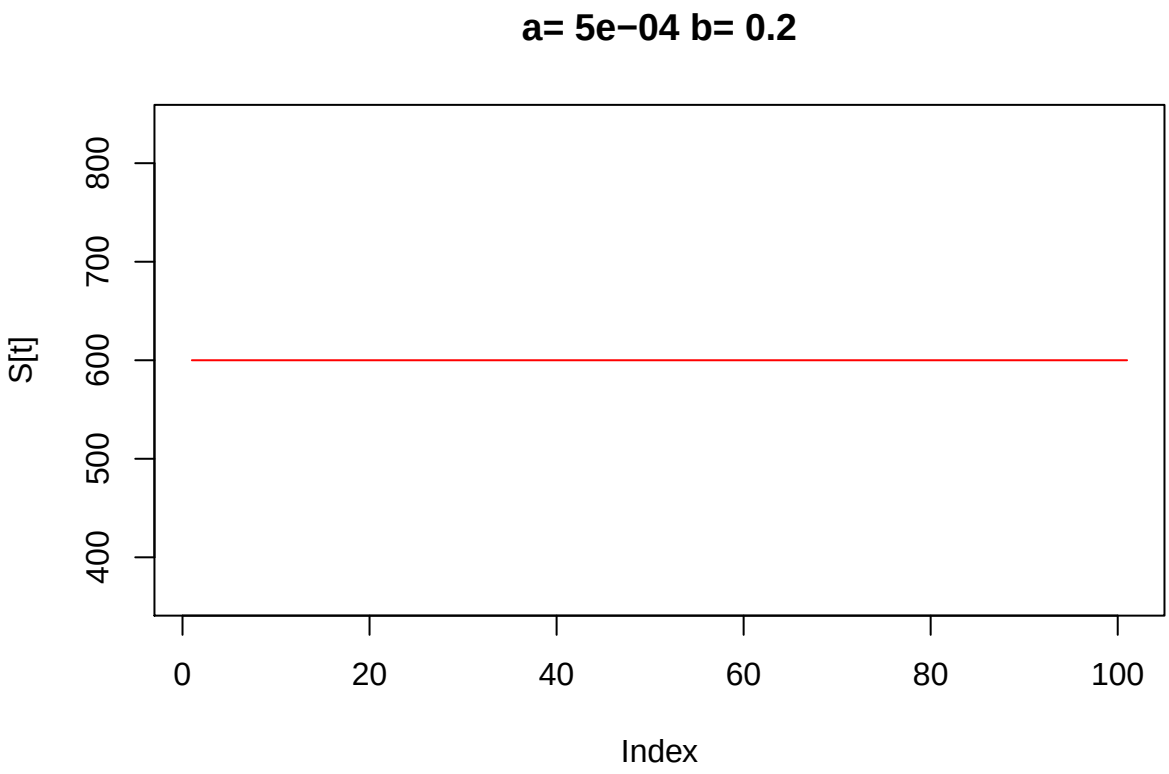
```
# SIRsim.r

# SIRsim <- function(a,b,N,T){
#   S<- rep(0,T+1)
#   I<- rep(0,T+1)
#   R<- rep(0,T+1)
#
#   S[1] <- N
#   I[1] <- 1
#   R[1]<- 0
#   for (i in 1:T){
#     S[i+1] <- rbinom(1,S[i],(1-a)^I[i])
#     R[i+1] <- R[i]+rbinom(1,I[i],b)
#     I[i+1] <- N+1-R[i+1]-S[i+1]
#     }
#  return(matrix(c(S,I,R),ncol=3))
# }
```

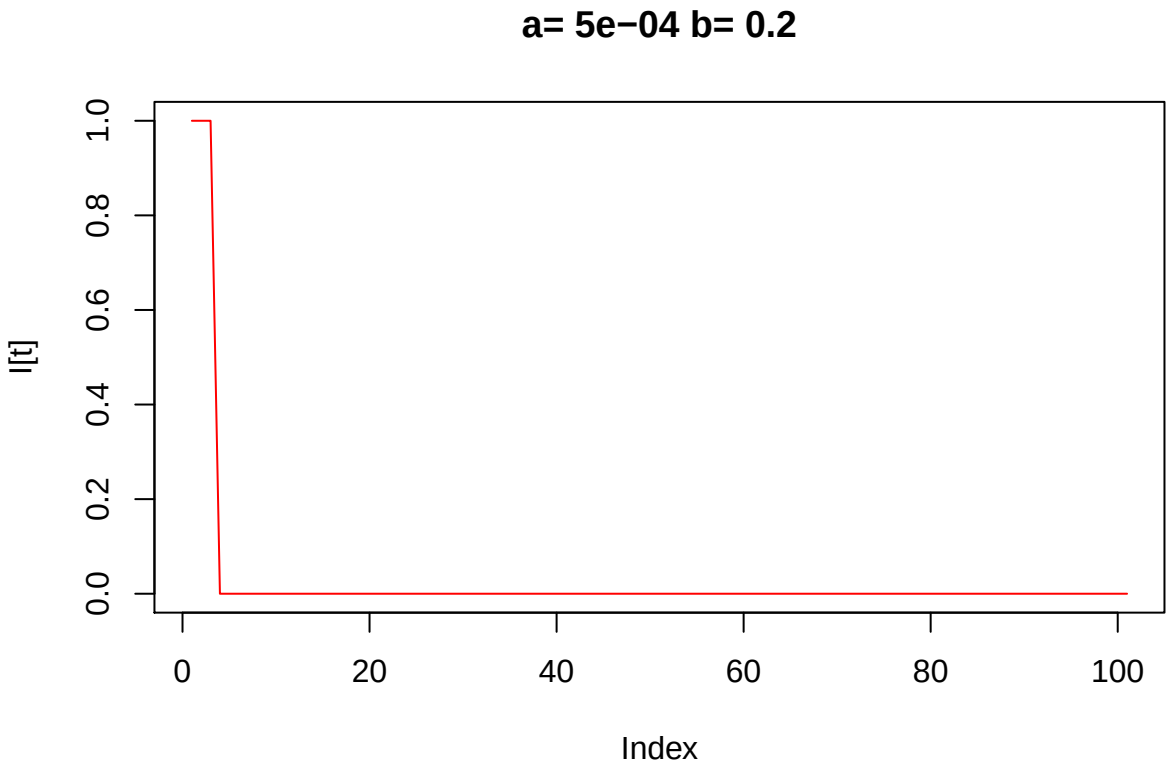
```
source("SIRsim.r")
a <- 0.0005
 b <- 0.1
N <- 600
  T <- 100
# y<- rep(0,100)
 for(b in c(0.1,0.2,0.3,0.4)){
y <- SIRsim(a,b,N,T)
 # show(y)
plot( y[,1],type = "l", col ="red",ylab="S[t]", main=paste("a=",a,"b=",b))
plot(y[,2],type = "l", col = "red",ylab="I[t]",main=paste("a=",a,"b=",b))
plot(y[,3],type = "l", col = "red",ylab="R[t]",main=paste("a=",a,"b=",b))
}
```

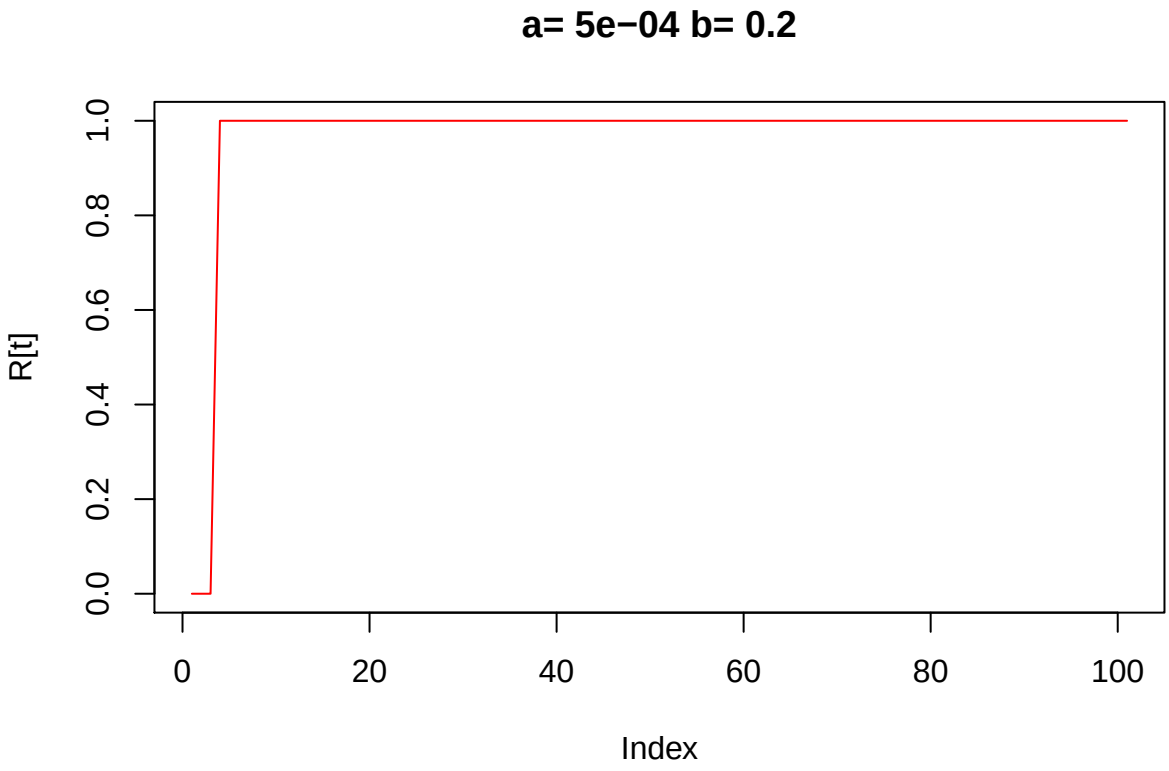


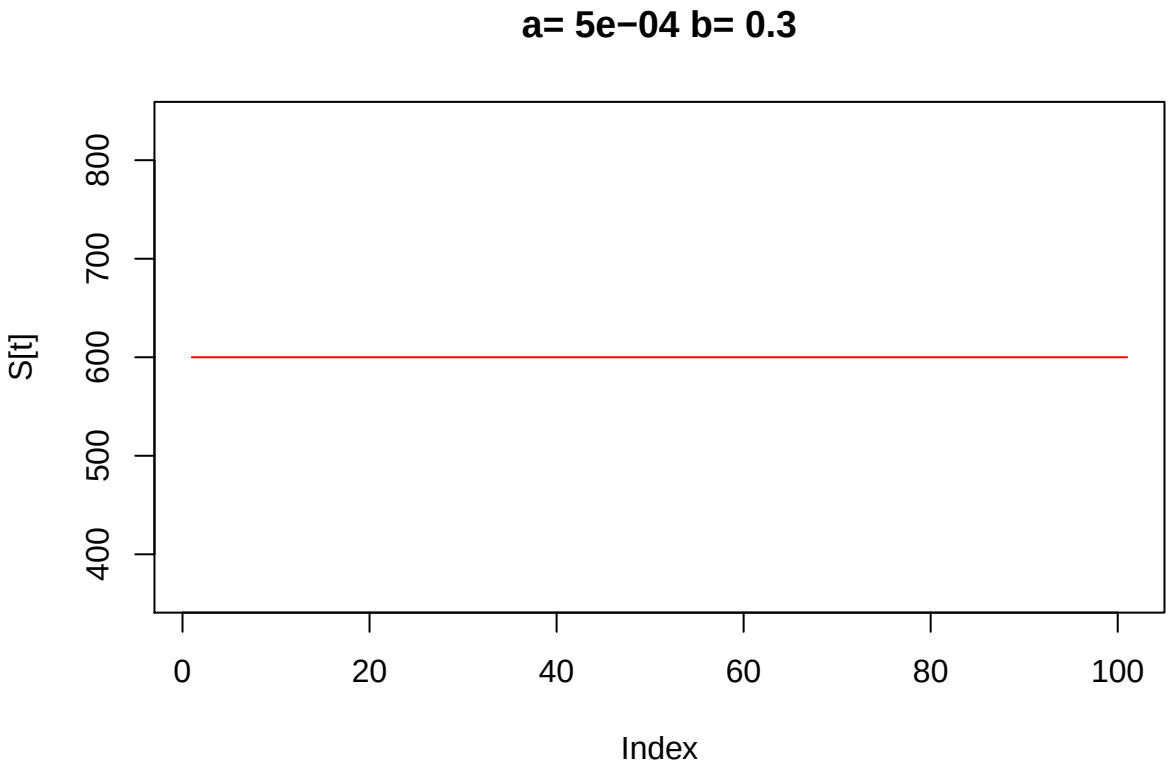

a= 5e−04 b= 0.1
I[t]
0
50
100
150
0
20
40
60
80
100
Index

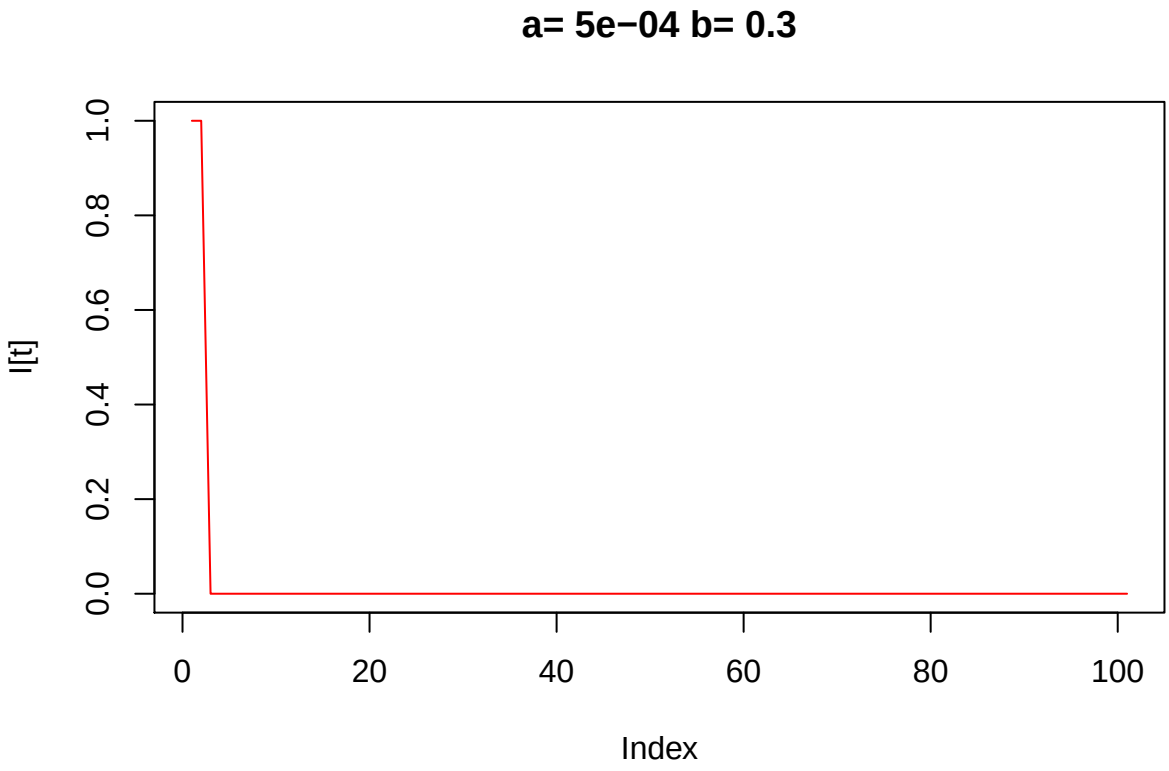

a= 5e−04 b= 0.1
R[t]
0
100
200
300
400
500
0
20
40
60
80
100
Index

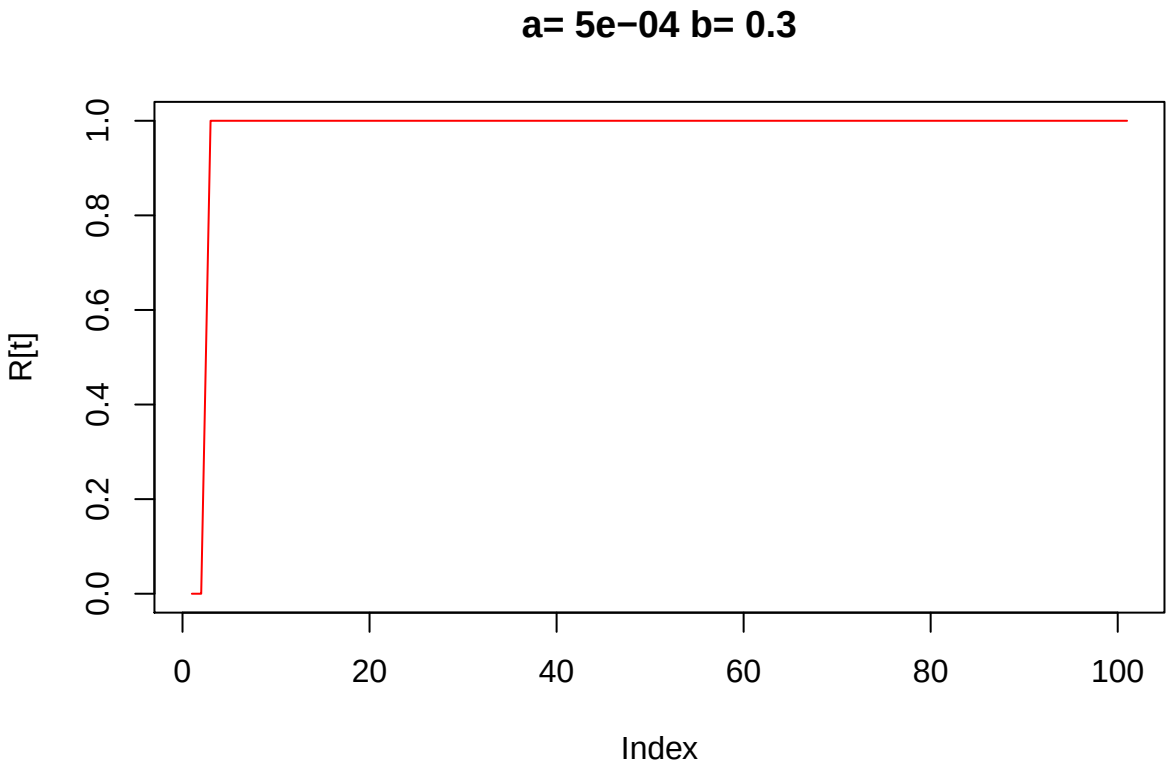

a= 5e−04 b= 0.2
S[t]
400
500
600
700
800
0
20
40
60
80
100
Index

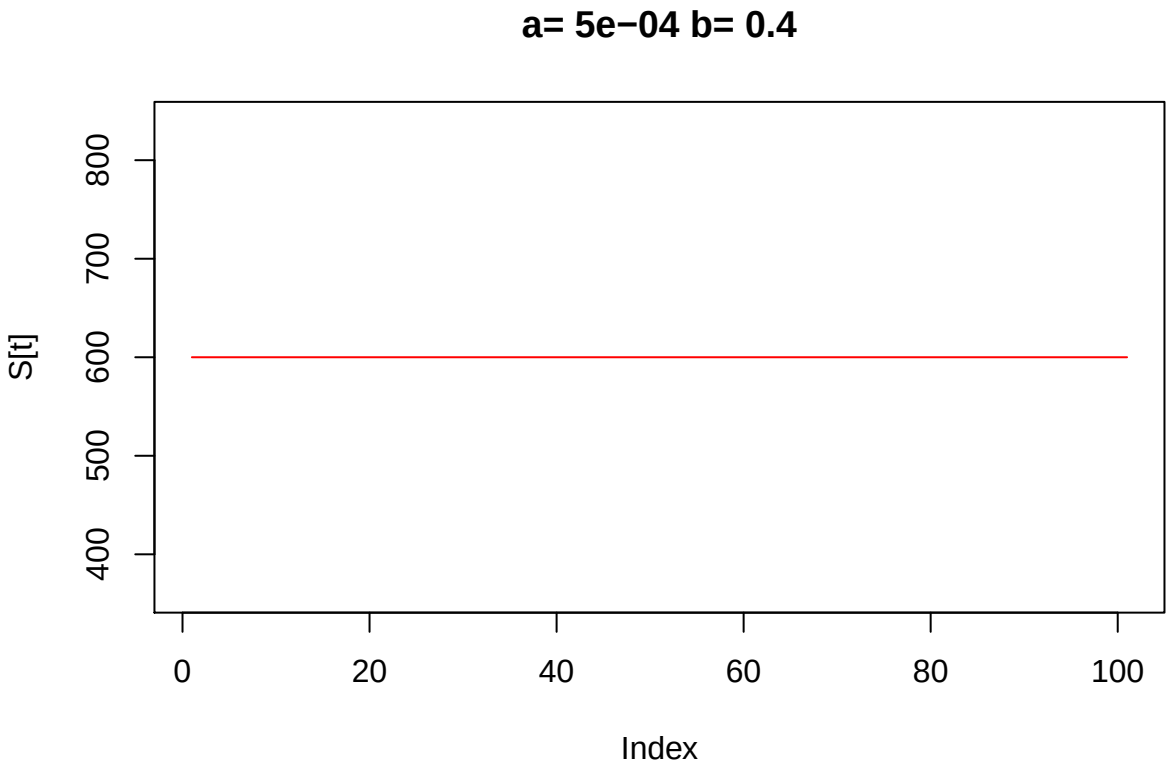

a= 5e−04 b= 0.2
I[t]
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
Index

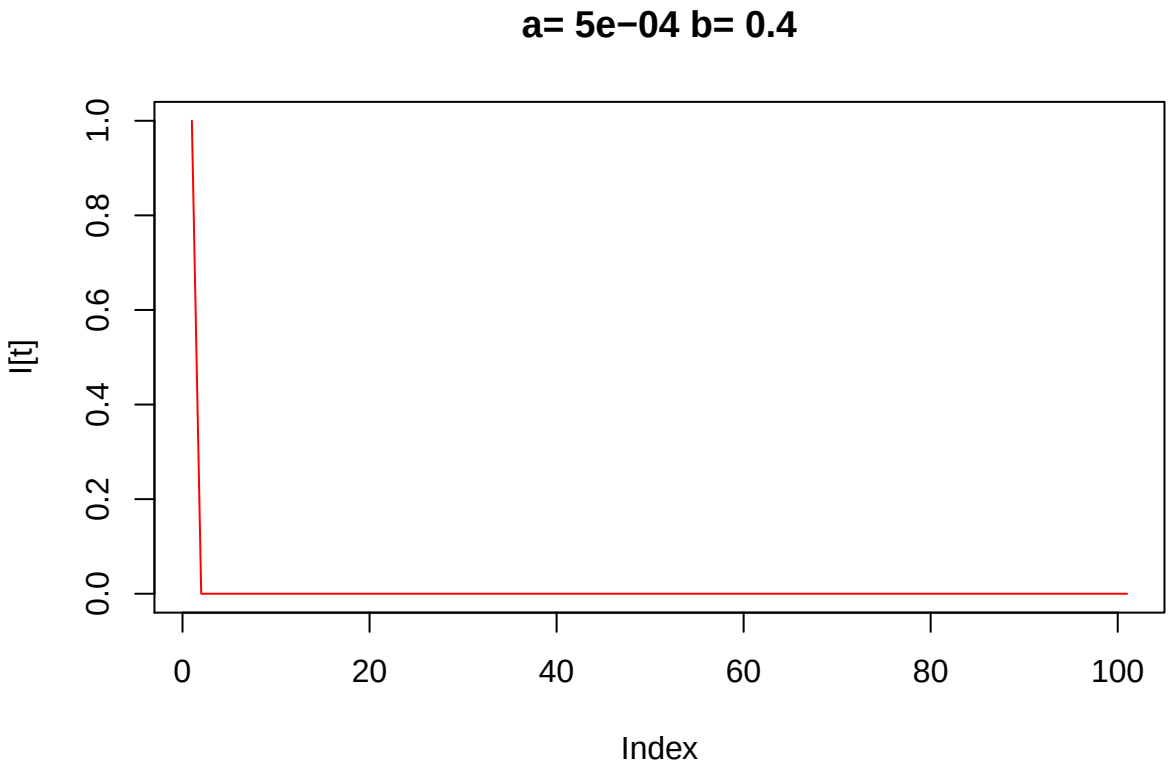

a= 5e−04 b= 0.2
R[t]
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
Index

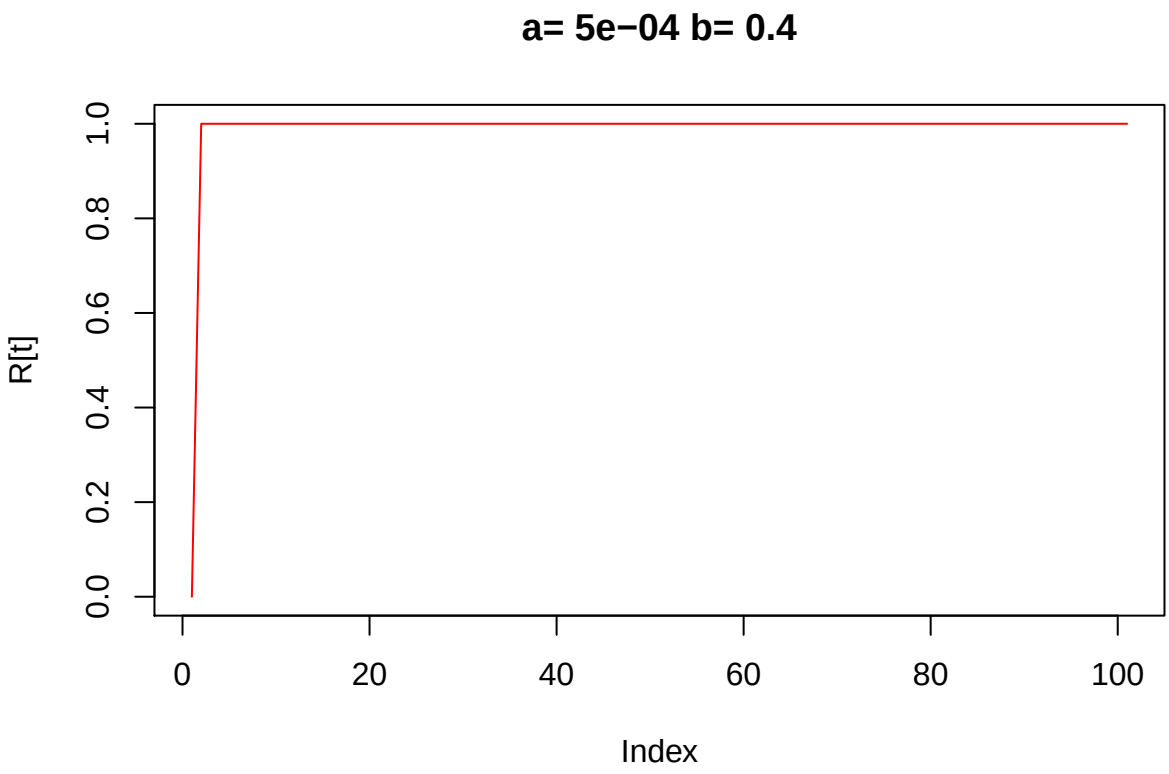

a= 5e−04 b= 0.3
S[t]
400
500
600
700
800
0
20
40
60
80
100
Index


a= 5e−04 b= 0.3
I[t]
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
Index


a= 5e−04 b= 0.3
R[t]
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
Index


a= 5e−04 b= 0.4
S[t]
400
500
600
700
800
0
20
40
60
80
100
Index


a= 5e−04 b= 0.4
I[t]
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
Index



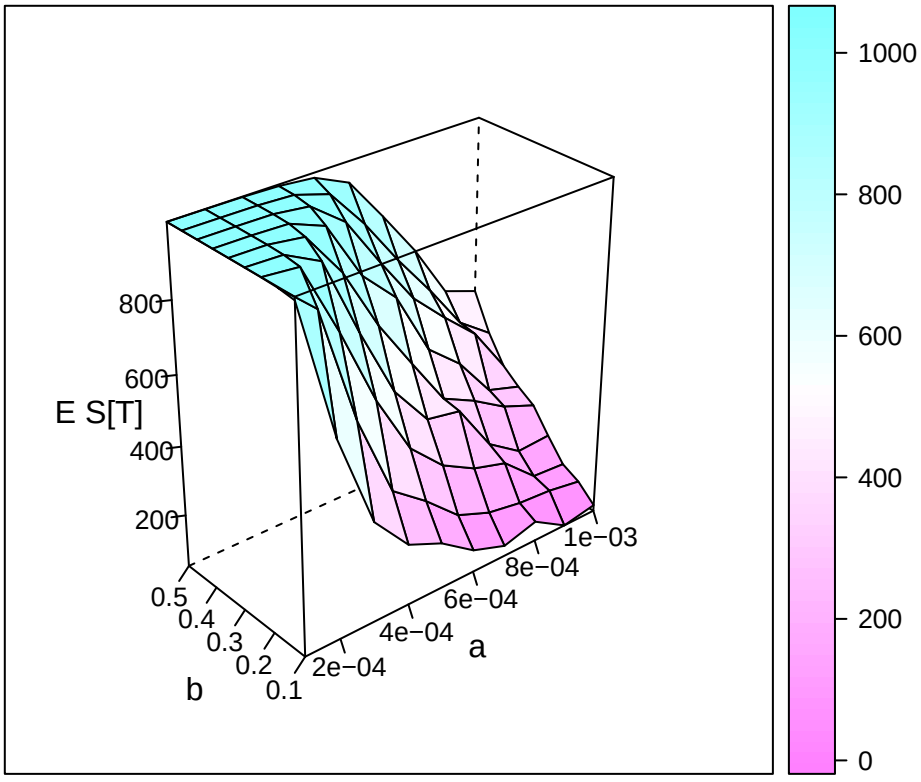
当我们给出 $\alpha = 0.0005, \beta = 0.1, 0.2, 0.3, 0.4$ 时，$S(t), I(t), R(t)$ 的模拟结果。由图形发现随着 β 的增加，传染数量减少。

为了考察当 α, β 取不同值时，模型有关可能行为的变化情况。

```
# SIR_grid.r
#  SIR <- function(a,b,N,T){
#   S<- N
#   I <- 1
#   R <- 0
#
#   for (i in 1:T){
#     S  <- rbinom(1,S ,(1-a)^I )
#     R  <- R +rbinom(1,I ,b)
#     I  <- N+1-R -S
#     }
#   return( c(S,I,R) )
#  }
```

```
source("SIR_grid.R")

# 设置参数值
N <- 1000
TT <- 100
a <- seq(0.0001,0.001,by=0.0001)
b <- seq(0.1,0.5,by=0.05)
```

```
n.reps<-400 # 估计 S[T] 大小的均值
f.name <-"SIR_grid.dat" # 保存模拟结果的文件

# 对不同 a, b 组合估计 E S[T]
write(c("a","b","S_T"),file=f.name,ncolumns = 3)
for (i in 1:length(a)){
 for (j in 1:length(b)){
 S.sum <- 0
 for (k in 1:n.reps){
  S.sum <- S.sum + SIR(a[i],b[j],N,TT)[1]

 }
 write(c(a[i],b[j],S.sum/n.reps),file=f.name,ncolumns=3,append=TRUE)
 }
}
 # 画出估计的 3D 图

 g<- read.table(f.name,header =TRUE)
 library(lattice)
 print(wireframe(S_T ~ a*b,data = g,scales=list(arrows =FALSE),aspect=c(.5,1),drape =TRUE,
```



可以观察到，疾病传播行为在 $N\alpha = \beta$ 时发生变化。$N\alpha$ 表示在时刻 1 时新感染的个体期望数目，β 表示在时刻 1 时感染者恢复正常时的期望数目。当 $N\alpha > \beta$ 时传染病规模变大，但是当 $N\alpha \leq \beta$ 时，转染病的数量急剧降低。

分支过程

在传播初期，如果 E(新感染者)>E(新恢复者)，则传染病有可能大规模爆发，对于一般的传染病疾病，由于个体的交叉影响计算 E(新感染者) 是困难的：- 有限的总体量意味着个体从正常到感染是“互相竞争的”；- 空间因素限制了感染者和易感者之间的接触。

SIR 模型忽略了空间之间的相互作用，仅模拟总体有限时的情形。分支过程忽略了有限总体这一限制，是一个简单而有用的模型。因此分支过程可被视作一种模拟传染病早期阶段的模型。

分支过程常用来描述人口的出生和增长问题，而非疾病的感染。令 Z_n 表示第 n 代时的总体数量，在任一时间，每个个体生产一定数量的后代，用随机变量 X 表示，然后死亡。令 $Z_0 = 1$，则

$$Z_{n+1} = Z_{n,1} + \cdots + Z_{n,Z_n}$$

这里，$X_{n,i}$ 表示第 i 个个体在第 n 代时家族个体数量。显然，$X_{n,i}$ 与 X 相互独立，且服从相同分布。

如果仅仅考虑感染情况，SIR 传染模型的第一步与分支过程第一步相同，即 $X_{0,1} = A + B$，这里 $A \sim binom(N, \alpha)$ 表示新感染者分布，$B \sim binom(1, 1-\beta)$ 等于 1 或者 0 分别表示感染者恢复正常或没恢复的情况。注意到 $EX = N\alpha + 1 - \beta$，传染病增长的条件 $N\alpha > \beta$ 等价于 $EX > 1$。

下面是一个分支过程。

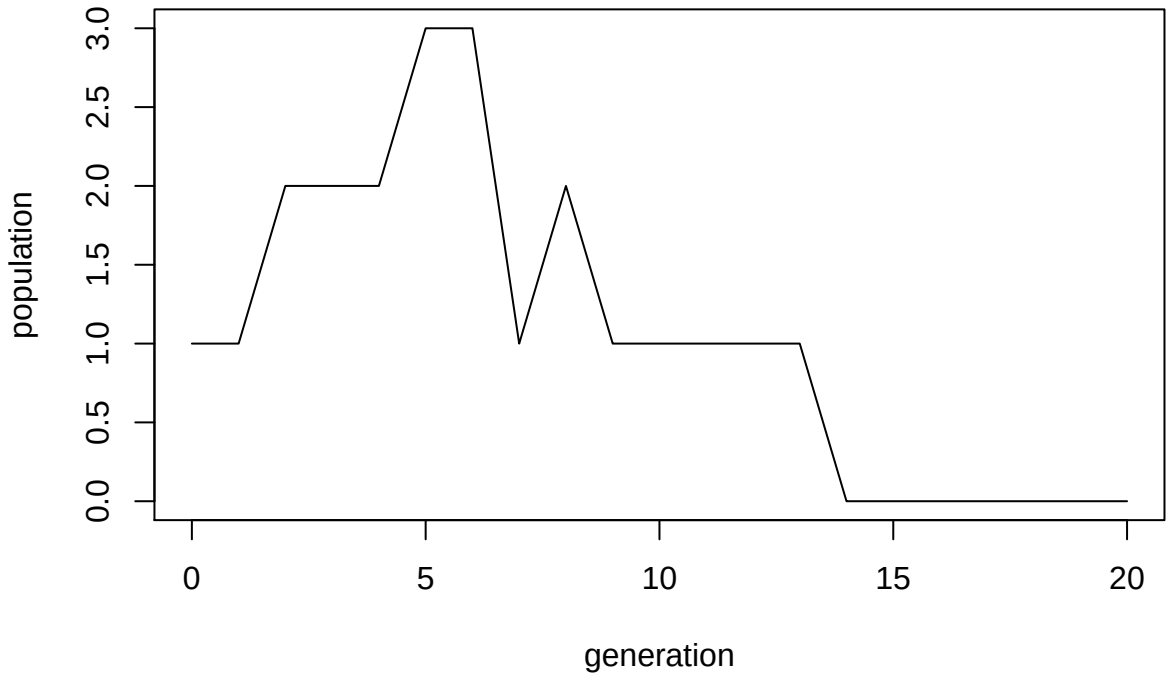
```
 # branching process simulation
 # 从第 0 代到第 gen 代的人口变化情况
# bp.r
#  bp <- function(gen,rv.sim,...){
#    Z <- rep(0,gen+1)
#    Z[1] <- 1
#    for (i in 1:gen){
#      if (Z[i]>0){
#        #Z[i] 是第 i-1 代人口数量
#        Z[i+1] <- sum(rv.sim(Z[i],...))
#      }
#    }
#    return(Z)
#  }

#  bp.plot <- function(gen,rv.sim,...,reps =1,logplot=TRUE){
#    # 模拟分支过程的人口变化曲线
#   # rv.sim() 模拟从后代分布中选 n 随机值
#    Z <- matrix(0,nrow=reps,ncol=gen+1)
#    for(i in 1:reps){
#      Z[i,] <- bp(gen,rv.sim,...)
#    }
#    if (logplot){
#      Z <- log(Z)
#    }
#    plot(c(0,gen),c(0,max(Z)),type="n",xlab="generation",ylab=if(logplot) "log populatio
#    for(i in 1:reps){
#      lines(0:gen,Z[i,])
#    }
#    return(invisible(Z))
#  }
```

当 $X \sim binom(2, 0.6)$ 时，给出 20 次由初始状态传递到第 20 代的模拟。结果显示，其中有一半的总体死亡，同时另一半总体成倍增长。

```
 library(spuRs)
```

```
## Loading required package: MASS
```

```
   source("bp.r")
  source("bp.plot.r")

bp.plot(20,rbinom,2,0.6,logplot=F)
```




```
# bp.plot(20,rbinom,2,0.6,20,logplot=F) 有错误
```

森林火灾

对空间相互作用的模拟。类似于 SIR 模型，假设模型总体由易感者 (未燃烧)、感染者 (燃烧) 和移除个体 (熄灭) 组成，两者的区别是森林火灾模型中个体放置在一个网格中，感染者只能感染与其相邻的易感个体。

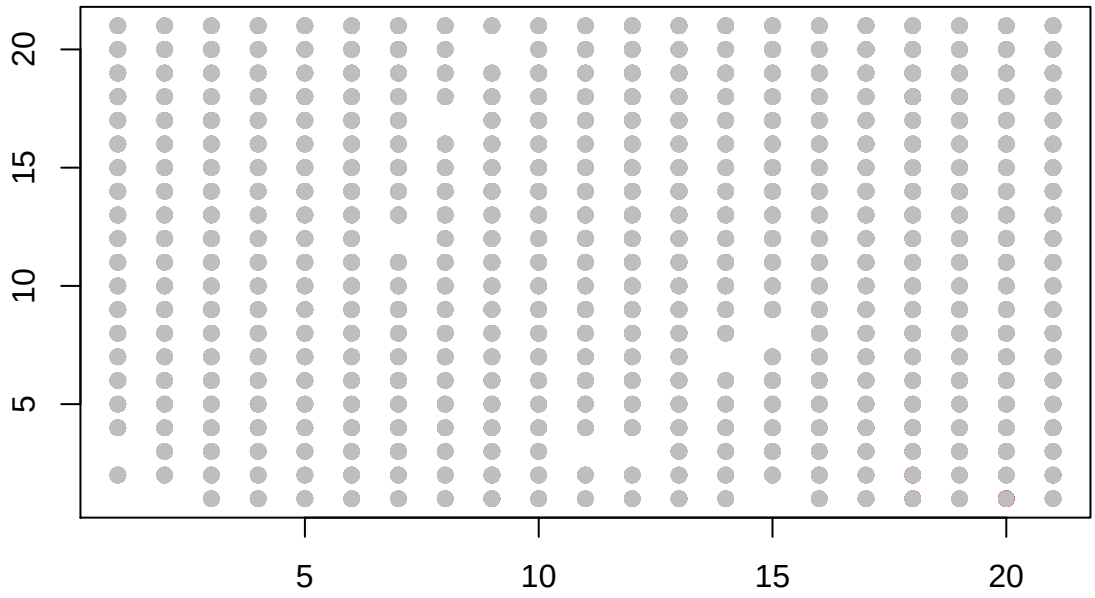
定义点 (x, y) 的 8 个邻点分别为 $(x-1, y-1), (x-1, y), (x-1, y+1), (x, y-1), (x, y+1), (x+1, y-1), (x+1, y), (x+1, y+1)$

用离散化步骤逐步建立模型，每一步中，每个感染个体以概率 α 感染与其相邻的易感个体，令 x 表示与一个易感个体相邻的感染者总数，则易感个体以概率 $(1-\alpha)^x$ 保持未被感染。在感染其他个体之后，每个感染者以概率 β 被移除。

假设森林火灾限制在一个 $N * N$ 的网格中，令 X_t 表示 t 时刻的一个 $N * N$ 矩阵，且令个体在点 (i, j) 为易感者时，$X_t(i, j) = 2$；为感染者时 $X_t(i, j) = 1$; 在该点被移除记 $X_t(i, j) = 0$.

```
# forest_fire.r
# program: spuRs/resources/scripts/forest_fire.r
# forest fire simulation
# rm(list = ls())
#
# neighbours <- function(A, i, j) {
#   # calculate number of neighbours of A[i,j] that are infected
#   # we have to check for the edge of the grid
#   nbrs <- 0
#   # sum across row i - 1
#   if (i > 1) {
#     if (j > 1) nbrs <- nbrs + (A[i-1, j-1] == 1)
#     nbrs <- nbrs + (A[i-1, j] == 1)
#     if (j < ncol(A)) nbrs <- nbrs + (A[i-1, j+1] == 1)
#   }
#   # sum across row i
#   if (j > 1) nbrs <- nbrs + (A[i, j-1] == 1)
#   nbrs <- nbrs + (A[i, j] == 1)
#   if (j < ncol(A)) nbrs <- nbrs + (A[i, j+1] == 1)
#   # sum across row i + 1
#   if (i < nrow(A)) {
#     if (j > 1) nbrs <- nbrs + (A[i+1, j-1] == 1)
#     nbrs <- nbrs + (A[i+1, j] == 1)
#     if (j < ncol(A)) nbrs <- nbrs + (A[i+1, j+1] == 1)
#   }
#   return(nbrs)
# }
#
# forest.fire.plot <- function(X) {
#   # plot infected and removed individuals
#   for (i in 1:nrow(X)) {
#     for (j in 1:ncol(X)) {
#       if (X[i,j] == 1) points(i, j, col = "red", pch = 19)
#       else if (X[i,j] == 0) points(i, j, col = "grey", pch = 19)
#     }
#   }
# }
#
# forest.fire <- function(X, a, b, pausing = FALSE) {
#   # simulate forest fire epidemic model
#   # X[i, j] = 2 for susceptible; 1 for infected; 0 for removed
#
#   # set up plot
#   plot(c(1,nrow(X)), c(1,ncol(X)), type = "n", xlab = "", ylab = "")
#   forest.fire.plot(X)
#
#   # main loop
#   burning <- TRUE
#   while (burning) {
#     burning <- FALSE
#     # check if pausing between updates
#     if (pausing) {
#       input <- readline("hit any key to continue")
```

```
#     }
#
#     # update
#     B <- X
#     for (i in 1:nrow(X)) {
#       for (j in 1:ncol(X)) {
#         if (X[i, j] == 2) {
#           if (runif(1) > (1 - a)^neighbours(X, i, j)) {
#             B[i, j] <- 1
#           }
#         } else if (X[i, j] == 1) {
#           burning <- TRUE
#           if (runif(1) < b) {
#             B[i, j] <- 0
#           }
#         }
#       }
#     }
#     X <- B
#
#     # plot
#     forest.fire.plot(X)
#   }
#
#   return(X)
# }

# spark
source("forest_fire.r")
set.seed(3)
X <- matrix(2, 21, 21)
X[11, 11] <- 1
# big fires
#X <- forest.fire(X, .1, .2, TRUE)
X <- forest.fire(X, .2, .4, TRUE)
```




```
## hit any key to continue
## hit any key to continue
## hit any key to continue
## hit any key to continue
## hit any key to continue
## hit any key to continue
## hit any key to continue
## hit any key to continue
## hit any key to continue
## hit any key to continue
## hit any key to continue
## hit any key to continue
## hit any key to continue
## hit any key to continue
## hit any key to continue
## hit any key to continue
## hit any key to continue
## hit any key to continue
## hit any key to continue
## hit any key to continue
## hit any key to continue
## hit any key to continue
## hit any key to continue
## hit any key to continue
## hit any key to continue
## hit any key to continue
## hit any key to continue
## hit any key to continue
```

```
## hit any key to continue
## hit any key to continue
## hit any key to continue
## hit any key to continue
## hit any key to continue
## hit any key to continue
## hit any key to continue
## hit any key to continue
## hit any key to continue
## hit any key to continue
## hit any key to continue
## hit any key to continue
## hit any key to continue
## hit any key to continue
## hit any key to continue
## hit any key to continue
## hit any key to continue
## hit any key to continue
## hit any key to continue
## hit any key to continue
```

```
# medium fires
#X <- forest.fire(X, .07, .2, TRUE)
#X <- forest.fire(X, .1, .4, TRUE)
# small fires
#X <- forest.fire(X, .05, .2, TRUE)
#X <- forest.fire(X, .07, .4, TRUE)
# X<-matrix(2,21,21)
# x[21,]<-1
```

注：这个程序需要与键盘交互，pdf 文件看不出动态效果来。

在时刻 5,10,15，20 时模拟森林火灾传播。其中，深灰色和浅灰色分别表示感染者移除个体，这里 $\alpha = 0.2, \beta = 0.4$, 且假设在开始时在网格中心存在一个单一着火点，观察图形可以发现，存在一个阈值，在该阈值下森林火灾发生的概率很小，但在该阈值以上，火灾有可能变得很大。同时，该模型存在新感染者出现频率和感染者被移除频率之间的平衡问题。

显然，随着 α 的增加或者 β 的减少，森林发生火灾的概率将大大增加。类似于 SIR 模型和分支过程模型，这里存在一个阈值。在此阈值之上，大火灾的出现几率将大大增加。比如，假设火灾是沿着一个直线方向燃烧，那么每一个未燃烧的树木 (易感者) 都与其他三个燃烧的树木 (感染者) 相邻，则该树木着火的可能性为 $1-(1-\alpha)^3$. 因此，考虑到燃烧树木以概率 β 被移除，可以推测，如果 $1-(1-\alpha)^3 > \beta$，则火势将会变大。

当然，这样的推测低估了火灾发生的可能性。原因在于火灾的传播方向并不是沿直线进行，不规则的火灾传播方向比直线火灾传播速度要快得多。即使火灾传播路线开始沿直线前进也很快会扭曲，这一点可以由模拟初始条件看出。

```
# X<-matrix(2,21,21)
# x[21,]<-1
```

案例参考

http://www.ms.unimelb.edu.au/spuRs/

欧文. 琼斯著，王亮等译，《R 语言的科学编程与仿真》，西安交大出版社，

如有错误请联系 [e-mail]:shixiangbupt@qq.com.