*Question 3: changing inputs*

*a) Repeat question 2 for annual premiums £5550, £5750, £6000, …., £8000. Plot a graph showing how the probability of bankruptcy varies with annual premium. All else being equal, at what level should the premium be set if the risk of bankruptcy should be no more than 2%?*

For N=10000

|  |  |
| --- | --- |
| **Premium** | **p(bankruptcy)** |
| 5500 | 0.1966 |
| 5750 | 0.1396 |
| 6000 | 0.0963 |
| 6250 | 0.0714 |
| 6500 | 0.0466 |
| 6750 | 0.0307 |
| 7000 | 0.0238 |
| 7250 | 0.0146 |
| 7500 | 0.012 |
| 7750 | 0.0077 |
| 8000 | 0.0066 |

To ensure that the risk of bankruptcy is no more than 2%, the premium should be set at £7250.

### Question 3a)

###Repeat the simulation from premium of £5500 to £8000 in increments of £250

##Need to repeat this function many times and store the result.

assets<-250000

p<-0.1 # probability that a customer makes a claim

n<-1000 # no of customers

z<-5500 # premium

w<-0

#define parento distribution parameters alpha (a) and beta (B)

a<-3

b<-100000 #parento parameters

#success or failure implies that x is a binomial function

### set the number of simulations

N=10000

#set the loop for the variable premium

for(j in 0:10)

{

#set the loop

for(i in 1:N)

{

x=rbinom(1, n, p)

#x is the number of claims made in a year

U=runif(x)

claim<-(((b^a)/(1-U))^(1/a)-b)

claim

d<-sum(claim)

w[i]<-(assets+n\*z-d)

}

#define q as a vector containing the results of the simulation

q<-c(w[1:N])

#count the number of times the insurance company will go bankrupt

count=0

for (i in 1:N)

{

if (q[i]<0) {

count = count + 1

}

}

#number of bankruptcies

# probability of bankruptcy

pbankrupt[j]<-count/N

print(z)

print(count/N)

#reassign next value of the premium z by increasing by £250

z<-z+250

}