

prisoner problem

George Papadopoulos

Wording

You are part of an exploration group of $2n$ people to a new planet. Unfortunately when you reach the new found planet, an alien civilization places you under arrest and imprisons you. They decide to put you on a strange trial and if you win they will set you free. Firstly they assign to each one of you a different number from 1 to $2n$. The prison contains a room in which there are placed $2n$ boxes and on each one of these boxes you can see a painted number from 1 to $2n$. In each of those boxes the alien guards place $2n$ enumerated cards from 1 to $2n$ shuffled. Each one of you enters the room with the boxes and must find the box with the number that is assigned to each prisoner by opening n boxes at maximum. When one enters and finishes the trial, the cards are placed again in each box exactly as they were shuffled before the trial begun, meaning that for each prisoner the placement of the cards is the same. If all of you find your number assigned to you according to the procedure you are all free to go and unfortunately during the trial you cannot communicate with each other. Since you are clever enough and know well the probability theory, before you begin you decide to follow the following strategy:

Each one that enters the room firstly opens a box with the number that he/she is assigned to by the guards, if the number written on the card is the number of the prisoner the subject stops if not she opens the box with the number that he/she reads on the card and so on until n boxes at maximum are opened, if he/she cannot find the number on any card all the prisoners are killed and the trial ends, on the other hand if he/she finds his/her number the cards are shuffled in the same way as for the previous prisoner and are placed in the boxes waiting for the next prisoner to open them.

(a) Write a function *single_prisoner* which will compute experimentally the probability that one of you will fail in the trial.

solution

```

## inputs --> n the number of boxes,
## k --> the number of the prisoner,
## nreps --> the number of repetitions of the experiment

single_prisoner <- function(n = 10,k = 1,nreps = 1000) {
  if (k > 2*n) stop(sprintf("prisoner number exceeds %d",2*n))
  cache <- vector("numeric",nreps)
  for (ii in 1:nreps) {
    boxes <- sample(2*n)
    q <- boxes[k]
    for (kk in 1 : n) {
      if (k == q) {
        cache[ii] <- 1
      } else {
        q <- boxes[q]
      }
    }
  }
  sum(cache)/nreps
}

```

(b) Write a function named *all_prisoners* which will compute experimentally the probability to leave you free, which means that all the prisoners have to succeed in the trial. The function will take as inputs the same arguments as in the previous function except *k* since all prisoners will be put to the trial, remember that if one fails, the guards kill everyone.

solution

```

all_prisoners <- function(n = 10,nreps = 1000) {
  cache <- vector("numeric",nreps)
  for (ii in 1:nreps) {
    cache1 <- vector("numeric",2*n)
    boxes <- sample(2*n)
    for(jj in 1:(2*n)) {
      q <- boxes[jj]
      for (kk in 1:n) {
        if (q == jj) {
          cache1[jj] <- 1
          break
        }
      }
      else {
        q <- boxes[q]
      }
    }
  }
  sum(cache1)/nreps
}

```

```

    }
  }
}
if (sum(cache1) == 2*n) cache[ii] <- 1
}
cache
sum(cache)/nreps
}

```

(c) Suppose that you don't have an extended knowledge of mathematics and you don't follow any strategy. Before each prisoner gets into the room he/she chooses beforehand n numbers from 1 to $2n$ and then opens the corresponding boxes hoping that one of them will contain the card with his/her number. Compute the theoretical probability that he/she will be freed and write a program to confirm your theory. Finally compute the theoretical lower bound of the probability of success for the strategy followed in the first two problems.

solution

The probability is $\frac{n}{2n} = \frac{1}{2}$.

```

single_prisoner_no_strategy <- function(n = 10, k = 1, nreps = 1000) {
  if (k > 2*n) stop(sprintf("prisoner number exceeds %d", 2*n))
  cache <- vector("numeric", nreps)
  for (ii in 1:nreps) {
    boxes <- sample(2*n)
    choices <- sample(2*n, n, replace = FALSE)
    if (length(intersect(boxes[choices], k)) > 0) {
      cache[ii] <- 1
    }
  }
  sum(cache)/length(cache)
}

```

We expect that for the random strategy of pre-choosing n boxes this strategy will lead to sure death of all prisoners since $\frac{1}{2^{2n}} \rightarrow 0$ as $n \rightarrow \infty$.

```

all_prisoners_no_strategy <- function(n = 10, nreps = 1000) {
  cache <- vector("numeric", nreps)
  for (ii in 1:nreps) {
    boxes <- sample(2*n)
    cache1 <- vector("numeric", 2*n)
  }
}

```

```

    for (jj in 1:(2*n)) {
      choices <- sample(2*n,n,replace = FALSE)
      for (kk in 1:n) {
        if (length(intersect(choices,jj)) > 0) {
          cache1[jj] <- 1
        }
      }
    }
    if (sum(cache1) == 2*n) cache[ii] <- 1
  }

  sum(cache)/nreps
}

```

Theoretical Explanations For Questions (a) And (b).

Computation of minimum probability for success if the strategy explained in (a) is followed:

Let us compute the permutations that contain the circles with exactly length ℓ greater than n , these are the circles that lead to failure.

$$\bullet \sum_{k>n, k \leq 2n} \binom{2n}{k} (k-1)!(2n-k)! = \sum_{k>n} \frac{(2n)!}{k}$$

Thus the probability of failure is

$$\bullet \frac{1}{(2n)!} \sum_{k>n} \frac{(2n)!}{k} = \sum_{k>n} \frac{1}{k}$$

and so if the strategy is followed the minimum probability of failure is

$$\bullet \sum_{k>n} \frac{1}{k} \leq \int_n^{2n} \frac{1}{t} dt = \log(2n) - \log(n) = \log(2)$$

and finally the minimum probability of success is $1 - \log(2)$.