The Airplane Problem: Monte Carlo Simulation

Jeffrey Cheng

June 19th, 2019

1 Introduction

Problem: one hundred people line up to board an airplane. Each has a boarding pass with assigned seat. However, the first person to board has lost his boarding pass and takes a random seat. After that, each person takes the assigned seat if it is unoccupied, and one of unoccupied seats at random otherwise. What is the probability that the last person to board gets to sit in his assigned seat? Problem found ***here***. Answer is 1/2.

2 Monte Carlo Simulation

```
```{r}
set.seed(100)
sample_size <- 10000
count <- 0

for(i in 1:sample_size) {
  #create array for passengers, list for seats, list for available
      seats
   passengers <- array(1:100, dim = c(1, 100))
   seats <- as.list(rep(0, 100))
   available_seats <- 1:100
```

In this simulation, two parallel arrays are created: an array from 1 to 100 for passengers, and an array from 1 to 100 for available seats. The seat number "1" corresponds to the correct seat for the passenger denoted number "1". A list of 100 0's is also created and labeled "seats" to keep track of which seats are taken.

```
  #set first passenger to sit in a random seat. Domain: a random
      seat from 1-100
   passengers[1] <- floor(runif(1, 1, 100))
```

The first passenger is assigned to a random seat.
```

```
  for(j in 1:100) {
    if(seats[passengers[j]] == 0) {
      seats[passengers[j]] <- 1
      available_seats <- available_seats[available_seats!=
          passengers[j]]
    } else {
      #new seat to sit in is generated randomly. Domain: a random
          seat that is available
      passengers[j] <- sample(available_seats, 1, replace = T)
      seats[passengers[j]] <- 1
      available_seats <- available_seats[available_seats!=
          passengers[j]]
      }
  }
```

This for loop makes the decision for the rest of the passengers. If their seat is empty, they will sit in it. If their seat is full, they will randomly take a seat from the list of available seats that is constantly updated with each passenger.

```
  #deterministic calculation: count = count + 1 if last passenger
      is in his original seat
    if(passengers[100] == 100) {
      count = count + 1
      }
}
#aggregation of results
print(count/sample_size)
```

The number of times the 100^{th} passenger sat in the 100^{th} seat is recorded and printed as a fraction over the total number of trials. If the seed is set to 100 and the sample size is 10000, the result should be a probability of 0.5093.