

Name: Bharath Rajakumar

Group Members

Manoj Chandran Jaya Chandran
Vivek Venkatesh Ganesan
Radhika Srinivasan

Section 1: Simulating the Experiment

There are 20 computers with 1 computer getting infected with virus. The virus spreads from one infected computer to another computer with probability 0.1. This is equivalent to a binomial distribution with $n=20$ and $p=0.1$. The given experiment can be simulated using Monte Carlo methods. Implementing Monte Carlo method reduces to generation of random variables from the given distributions. Hence, it remains to design algorithms for generating random variables and vectors. First we generate n uniform random numbers and then sum the values which are less than ' $p=0.1$ '.

Example: In our problem if a computer is infected with virus. We can represent it as the following in R

```
> runif(19, 0, 1)
[1] 0.49959646 0.78947178 0.40292003 0.73119738 0.43114891 0.20388417
0.68888006 0.35219826
0.63410817 0.45078476 0.89975063 0.48450874 0.31592915 0.53352271 0.38854984
[16] 0.08051184 0.71848667 0.15141158 0.23734610
```

Here we have $0.0805 < 0.1$. So we assign one among the remaining 19 computers to be affected with virus. This is used to convert a uniform distribution to a Bernoulli distribution. We simulate a Bernoulli trial with a probability of success p where in if the probability is < 0.1 , it is considered to be a success and if probability is > 0.1 , it is considered to be a failure. In the above sample of uniform distribution, we will have four 1's (that signifies that 4 computers have been infected) followed by 0's (computers that are not infected).

Section 2: Answer for each Questions

- a) Expected time to remove virus from the whole network: 1.033239
- b) Probability that all computers are infected at least once: 0.000000
- c) Expected number of computers that get infected: 2.983350

Explanation for R-Code

First assign the number of computers involved in the experiment

Next we are given the probability of a computer getting infected as 0.1

We create an array and fill it with 0's (to denote no computer is infected)

Next we assign one computer to be infected randomly

When the number of computers getting infected is greater than one, we keep continuing with experiment and also the number of days would get incremented while there is at least one computer infected with the virus after it is cleaned.

During each day, we find out the number of computers getting infected and also count the number of days.

If the number of computers that get infected is less than 5, we disinfect them by changing their value to 0 and if the number of computers is greater than 5, we randomly select 5 computers and disinfect them.

This information is used to find out the average time taken to completely remove the virus from the network and the average computers that get infected.

At the end of each experiment, we return the number of days to complete the experiment

The above experiment is performed for a large number of times to get a roughly accurate value for expectation.

The above algorithm returns the expected time (in days) it takes to remove the virus from the network.

[Question a)]

The expected number of computers to be infected is calculated by adding the number of computers getting infected in each simulation and dividing it by the total number of simulations.

[Question c)]

Similarly we can count the number of times each computer in the network is getting affected at least once and divide it by the total number of simulations

[Question b)]

We do this simulation 16577 times to obtain a margin of error ± 0.01 with probability 0.99

Section 3: R Code

Global variables to solve part b) and c)

```
expectedComputersAffected <- 0
allComputersAffectedOnce <- 0
simulation <- function() {
  numberOfComputers = 20
  numberOfDays = 0
```

```
# infectedComputers: A boolean array to see if a particular computer is
affected or not
```

```
# 1 : Represents that the computer is infected
```

```
# 0 : Represents that the computer is not infected
```

```
infectedComputers = numeric(20)
allComputersAffected = numeric(20)
```

```
# One of the 20 computers gets infected with virus initially
```

```
infectedComputers[sample(1:20,1)] = 1
numberOfComputersInfected = 1
p = 0.1
while (numberOfComputersInfected >= 1) {
  numberOfDays = numberOfDays + 1
  # We generate a vector of probabilities with uniform distribution in the range
  (0,1)
```

```
# We select only values which are less than p=0.1 and assign the
corresponding
```

```
computer with 1 (means that computer is infected with virus)
```

```
rand <- runif(20 - numberOfComputersInfected, 0, 1)
```

```
tempRand <- which(rand < p)
```

```
for (i in tempRand) {
```

```
  if(infectedComputers[i] == 1) {
```

```
    infectedComputers[(i+1) %% numberOfComputers] = 1
```

```
  } else {
```

```
    infectedComputers[i] = 1
```

```
  } if(allComputersAffected[i] != 1) {
```

```
    allComputersAffected[i] = 1
```

```
  }
```

```
}
```

```
tempNumberOfComputersInfected = which(infectedComputers == 1)
```

```

expectedComputersAffected <- expectedComputersAffected +
length(tempNumberOfComputersInfected)

# Count the number of computers infected

# if greater than 5, then technician selects 5 random computers and cleans it
# else clean all the computers

if(length(tempNumberOfComputersInfected) > 5) {
  randomFiveComputers = sample(tempNumberOfComputersInfected, 5)
  for (i in randomFiveComputers) {
    infectedComputers[i] = 0
  }
} else {
  for (i in tempNumberOfComputersInfected) {
    infectedComputers[i] = 0
  }
}

# At the end of the day update the number of computers infected
numberOfComputersInfected = length(which(infectedComputers == 1))
}

# This is to check if in this simulation if all the 20 computers are affected
at least once

if (length(which(allComputersAffected==1)) == 20 ) {
  allComputersAffectedOnce = allComputersAffectedOnce + 1
}

# Returns the number of days it took to free the network from virus

return(numberOfDays)
}
display <- function() {
  sum = 0
  for (i in 1:16577) {
    sum = sum + simulation()
  } cat(sprintf(" \n a) Expected time to remove virus from the whole network :
%f", sum /16577))
  cat(sprintf("\n b) Probability that all computers are infected at least once
: %f", allComputersAffectedOnce / 16577))
  cat(sprintf("\n c) Expected number of computers that get infected : %f",
expectedComputersAffected / 16577))
  expectedComputersAffected <- 0
}

```