

```
library(dplyr)
```

```
##  
## Attaching package: 'dplyr'  
  
## The following objects are masked from 'package:stats':  
##  
## filter, lag  
  
## The following objects are masked from 'package:base':  
##  
## intersect, setdiff, setequal, union
```

```
set.seed(911)
```

```
#Part 1
```

```
#1  
network <- read.csv("network.csv")  
attributes <- read.csv("attributes.csv")  
  
#2  
SI_contagion_model <- function(network, initially_infected, p, time_steps) {  
  for (h in 1:time_steps) {  
    if (h == 1) {  
      time_points <- c()  
      for (i in initially_infected) {  
        time_points[i] <- h  
      }  
    }  
    for (i in initially_infected) {  
      for (j in 1:ncol(network)) {  
        if (j == 1) {  
          in_network <- c()  
        }  
        if (network[i,j] == 1) {  
          in_network <- c(in_network, j)  
        }  
      }  
      if (is.null(in_network) == FALSE) {  
        random_in_network <- sample(in_network, 1)  
        if (sum(random_in_network == initially_infected) == 0 & runif(1) <= p) {  
          initially_infected <- c(initially_infected, random_in_network)  
          time_points[random_in_network] <- h  
        }  
      }  
    }  
  }  
  return(time_points)  
}
```

```
#3
```

```
initially_infected <- subset(attributes, adoption_date==1) %>% row.names() %>% as.numeric()
SI_contagion_model(network, initially_infected, .1, 18)
```

```
## [1] 1 13 NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA 18 NA 15 NA NA NA
## [26] 12 1 NA NA NA NA NA NA NA NA NA NA 17 NA NA NA NA 9 NA 17 NA 13 NA NA NA NA
## [51] 16 NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA 1
## [76] NA NA 1 NA NA NA NA NA NA NA NA NA 3 NA NA 18 NA NA 1 NA NA NA NA NA NA NA
## [101] NA NA NA NA NA NA NA NA NA NA 4
```

```
SI_contagion_model(network, initially_infected, .5, 18)
```

```
## [1] 1 18 5 12 6 10 15 NA NA NA 2 10 9 16 NA 4 NA 10 NA 7 13 2 7 11 4
## [26] NA 1 15 6 NA 8 NA 18 NA 8 6 6 NA 15 11 5 6 NA 9 14 5 NA NA 5 NA
## [51] 4 9 NA 9 3 6 17 NA 16 NA NA NA NA 12 13 3 NA NA 14 NA NA NA NA NA 1
## [76] NA 8 1 NA NA 12 13 NA 17 NA 14 4 NA 13 2 NA 2 1 7 5 18 NA 6 7 7
## [101] 7 12 NA NA NA 13 NA NA 3 8 NA NA 9 9 NA NA 11
```

```
SI_contagion_model(network, initially_infected, .9, 18)
```

```
## [1] 1 9 6 2 11 10 3 NA 3 NA 3 11 7 7 NA NA 15 NA 13 6 6 8 4 NA 9
## [26] 9 1 5 7 NA 6 11 5 8 7 5 5 5 16 1 6 6 4 3 7 2 10 NA 5 NA
## [51] 13 5 NA 6 4 9 8 NA NA NA NA NA NA NA 11 8 NA NA 5 NA NA NA NA NA 1
## [76] NA 1 1 NA NA 3 9 NA 8 NA 7 4 NA 7 3 NA 3 1 6 13 9 NA 4 12 2
## [101] 4 7 10 NA NA 4 NA NA 11 1 12 NA 5 9 NA 11 15
```

#Part 2

#4

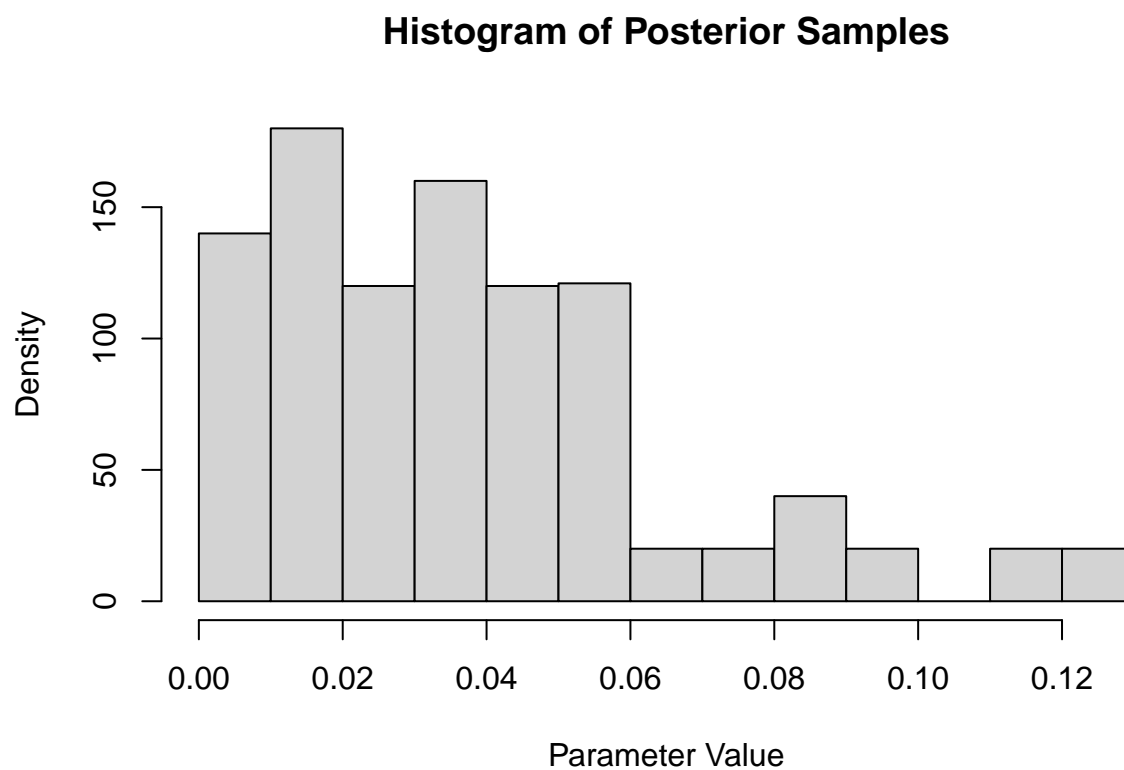
#98 and 99 are codes for missing data, not adoption time

```
for (i in 1:nrow(attributes)) {
  if (attributes[i,"adoption_date"] == 98 | attributes[i,"adoption_date"] == 99) {
    attributes[i,"adoption_date"] <- NA
  }
}
```

```
ABC <- function(N, tolerance_level) {
  theta_value <- c()
  summary_statistic <- c()
  for (i in 1:N) {
    p <- runif(1)
    generated_data <- SI_contagion_model(network, initially_infected, p, 18)
    summary_statistic[i] <- sum(abs(generated_data - attributes$adoption_date), na.rm=TRUE)
    theta_value[i] <- p
  }
  proposals <- data.frame(theta_value, summary_statistic)
  proposals <- proposals %>% arrange(summary_statistic)
  posterior_samples <- proposals$theta_value[1:N*tolerance_level]
  return(posterior_samples)
}
```

```
#5
```

```
posterior_samples <- suppressWarnings(ABC(N=1000, tolerance_level=.05))  
hist(posterior_samples, ylab="Density", xlab="Parameter Value", main="Histogram of Posterior Samples")
```



```
mean(posterior_samples)
```

```
## [1] 0.03687349
```

```
quantile(posterior_samples, .025)
```

```
##          2.5%  
## 0.003170931
```

```
quantile(posterior_samples, .975)
```

```
##          97.5%  
## 0.1148283
```