

C. MARÍA MONTEMAYOR PALOS

---

## 1. OBJETIVO

Examinar y analizar el efecto estadístico del valor de  $p_v$  (de cero a uno en pasos de 0.1) de un sistema multiagente con aplicación epidemiológica, con una vacunación con probabilidad  $p_v$  a los agentes al momento de crearlos de tal forma que desde el inicio se encuentren en el estado R y ya no podrán contagiarse ni propagar la infección, y el porcentaje máximo de infectados durante la simulación y el momento (iteración) en el cual se alcanza ese máximo.[3].

## 2. METODOLOGÍA

Para efectos de la tarea se utiliza el programa R versión 4.0.4 [4] para Windows. Los agentes siguen el modelo SIR, en el cual se pueden encontrar en tres estados diferentes: susceptibles, infectados o recuperados. Se considera como parámetros un cuadro de tamaño  $1 \times 1$ . en donde se posicionan  $n$  agentes uniformemente al azar y con una probabilidad de  $p_i$  en el que el agente puede estar infectado desde el inicio. Suponiendo que la infección produce inmunidad en los recuperados, por lo cual solamente los susceptibles podrán ser infectados. La probabilidad de contagio será proporcional a la distancia euclideana entre dos agentes  $d(i, j)$  de la siguiente manera:

$$(1) \quad p_c = \begin{cases} 0, & \text{si } d(i, j) \geq r, \\ \frac{r - d}{r}, & \text{en otro caso,} \end{cases}$$

donde  $r$  es un umbral.

## 3. CÓDIGO

Se modifica el código de Schaeffer [2]. Se establecen los números de agentes y se vacuna con probabilidad `pv` al momento de crearlos, así no podrán ser infectados en ninguna de las iteraciones posteriores. Se varía dicha probabilidad `pvvar` de cero a uno en pasos de 0.1. Se generan diez réplicas del experimento.

```
1 l <- 1.5
2 n <- 50 #Número de agentes
3 pi <- 0.05 #Probabilidad de infección
4 pr <- 0.02
5 v <- 1 / 30
6
7 pvvar <- seq(0,1,0.1) #Variacion de la probabilidad de 0 a 1
8 infectadosMax <- c()
9 for(pv in pvvar) { #Incremento en la probabilidad de vacunados
10   for(rep in 1:10) { #Replicas del experimento
```

Se generan los gráficos del porcentaje máximo de infectados durante la simulación y el momento (iteración) en el cual se alcanza ese máximo.

```
1 salida <- paste("vacuna=", pv, ".png", sep="")
2   png(salida)
3   plot(1:length(epidemia), 100 * epidemia / n, xlab="Tiempo",
4       ylab="Porcentaje de infectados",
5       main=paste("Vacunados: ",pv * 100, "%"))
6   graphics.off()
7   infectadosMax <- c(infectadosMax, max(epidemia)) #Se guardan los infectados
8   maximos por cada replica
```

Se realiza un `data.frame` para concatenar la probabilidad de vacuna con sus respectivas réplicas y se obtiene el porcentaje de infectados. Se realiza un diagrama violín para visualizar los resultados y se genera un gif de todos los pasos con la librería `gifski`.

```
1 datos <- data.frame(Probabilidad_de_vacuna = c(rep("0",10),rep("0.1",10),
2       rep("0.2",10),rep("0.3",10),
3       rep("0.4",10),rep("0.5",10),
4       rep("0.6",10),rep("0.7",10),
5       rep("0.8",10),rep("0.9",10),
6       rep("1",10)),
7       Infectados_maximos = (infectadosMax/n)*100)
8
9 library(ggplot2)
10 tiff("t6-Violin.png", units="in", width=12, height=6.8,
11     res=300, compression = 'lzw')
12 g <- ggplot(datos, aes(Probabilidad_de_vacuna,Infectados_maximos))
13 g + geom_violin(alpha = 0.5,draw_quantiles = c(0.25, 0.5, 0.75),
14     trim = FALSE,adjust=1.5,aes(fill= Probabilidad_de_vacuna))
15 + geom_jitter() + labs(x= "Probabilidad de vacuna",
16     y = "Porcentaje máximo de infectados",
17     fill = "Probabilidad de vacuna" )
18 graphics.off()
19
20 #Generación del GIF
21 library(gifski)
22 png_files <- list.files( pattern = "p6_t.*png$", full.names = TRUE)
23 gifski(png_files, gif_file = "animation.gif", width = 800, height = 600, delay = 1)
```

#### 4. RESULTADOS Y DISCUSIÓN

En la figura 1 se observa la primera (ver figura 1a) y la última iteración (ver figura 1b) de la simulación en la cual los agentes vacunados se encuentran en el estado R (representados con el triángulo color naranja) y muy pocos agentes en estado S (representados con cuadrado color verde). Debido a la vacunación hay una minoría de agentes infectados (representados con círculo color rojo), por lo tanto la propagación de la infección se mantiene baja. Los gifs generados se encuentran en el repositorio de Montemayor [5].

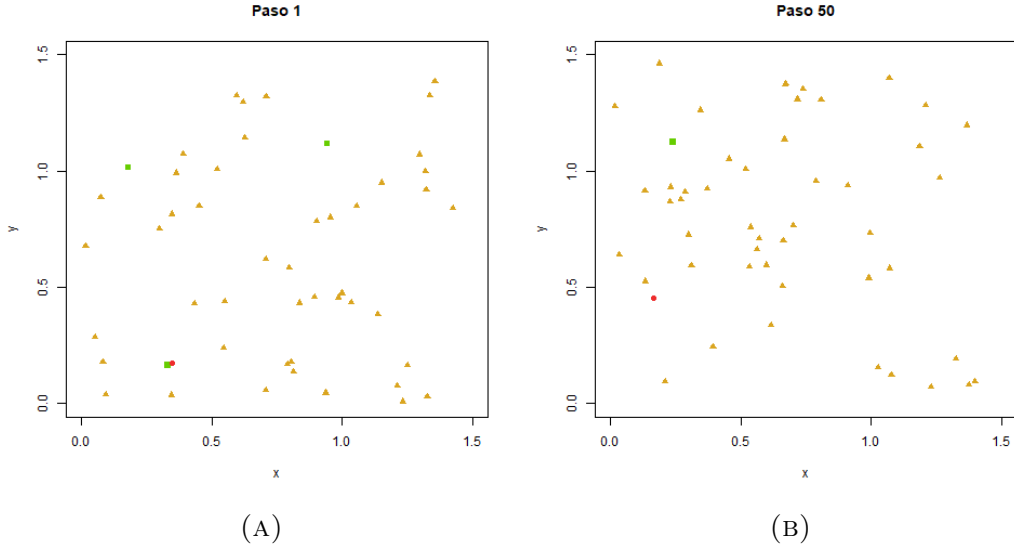


FIGURA 1. Iteración de infectados.

Se comparan los diferentes diagramas generados del porcentaje de máximo de infectados y el momento en el cual se alcanza ese máximo. En la figura 2 se observa que disminuye ligeramente el porcentaje de infectados con la vacunación del 10 % de los agentes como se muestra en la figura 2b a comparación de la figura 2a con el 0 % de agentes vacunados.

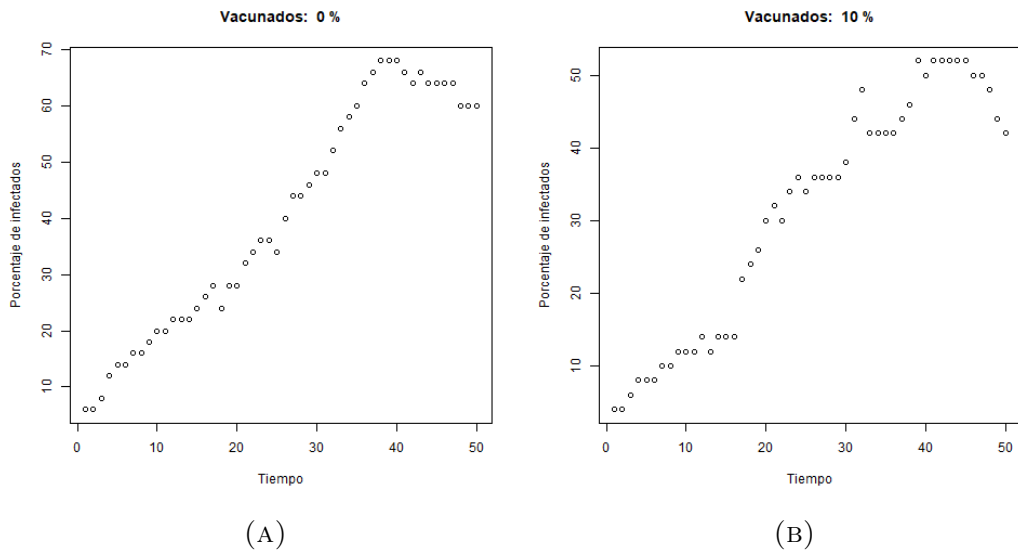


FIGURA 2. Comparación del porcentaje de máximo de infectados y el momento en el cual se alcanza el máximo.

En la figura 3 se compara el porcentaje de infectados, en la cual se logra apreciar un cambio radical entre la vacunación del 50 % de los agentes (ver figura 3a) y cuando los agentes están vacunados al 100 % logrando así un porcentaje de infección nulo (ver figura 3b).

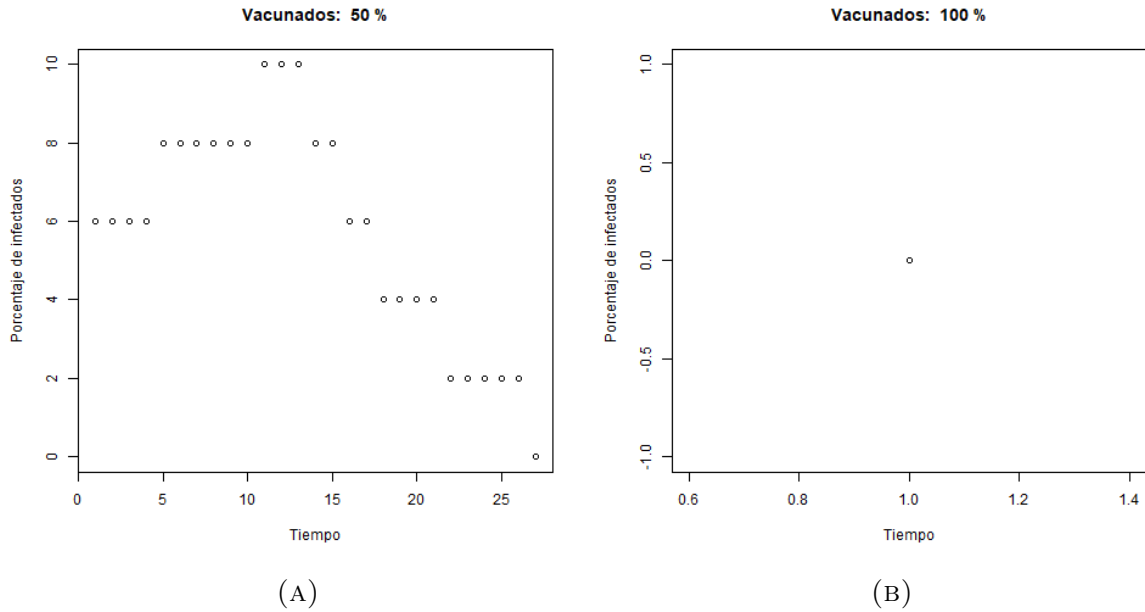


FIGURA 3. Comparación del porcentaje de máximo de infectados y el momento en el cual se alcanza el máximo.

En la figura 4 se muestran los gráficos de violín generados variando la probabilidad  $p_v$ . Como se puede apreciar en dicha figura, cuando se aumenta la probabilidad de vacunación, el porcentaje de infectados disminuye.

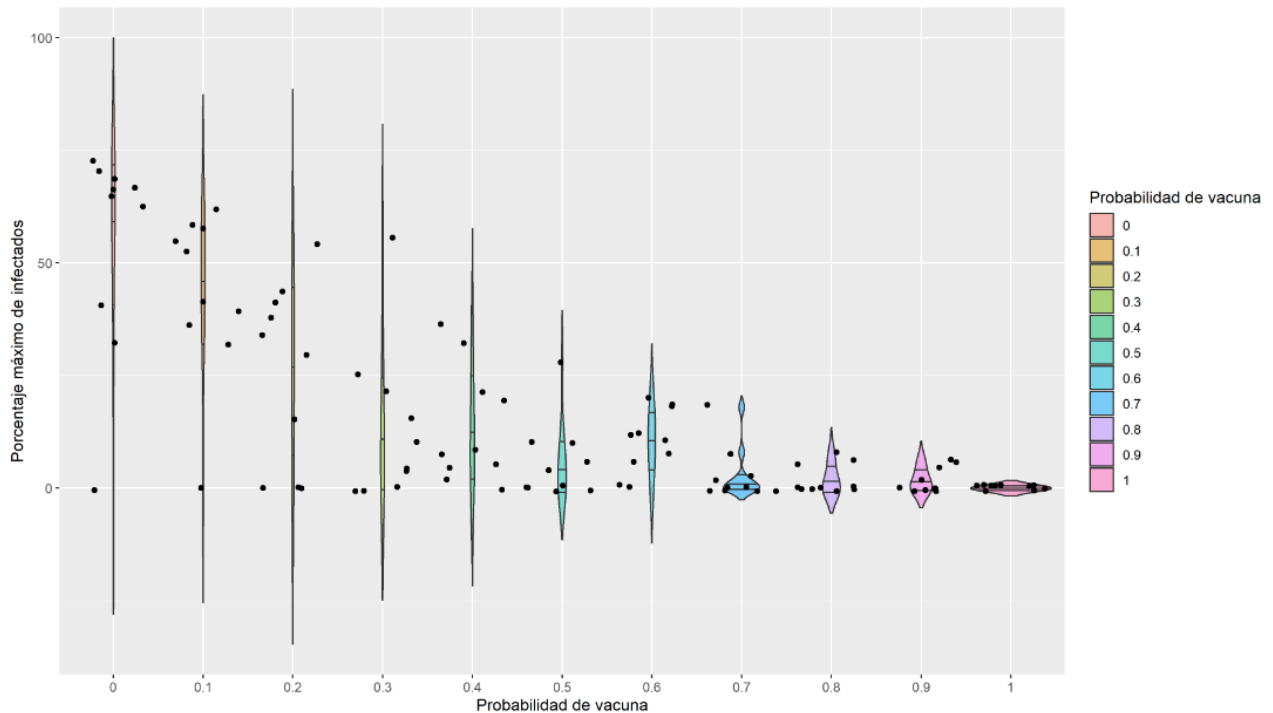


FIGURA 4. Gráficas de violín del porcentaje máximo de infectados variando la probabilidad de vacuna.

## 5. CONCLUSIÓN

Como es de esperarse, a mayor probabilidad de vacunación, menor porcentaje de infectados. Al estar los agentes vacunados desde el inicio, se garantiza que disminuyan de manera más rápida los contagios.

## 6. RETO 1

El objetivo del primer reto consiste en cambiar los patrones de movimiento a que no tengan una trayectoria fija los agentes, a que utilice el modelo de punto intermedio aleatorio (inglés: random waypoint model): cada agente tiene una posición meta  $(x, y)$  hacia al cual se mueve con una velocidad  $v$ ; al alcanzar (o superar) su meta, elige al azar una nueva meta uniformemente al azar. La velocidad de cada agente es una constante, normalmente distribuido sobre la población de agentes. El objetivo es examinar si surgieron cambios en el efecto de  $p_v$  por esta modificación. Se modificó el código de Schaeffer [2] y se apoyó en el repositorio de Llano [1]. Se realiza un `data.frame` en el cual se guardan las posiciones metas.

```
1 pvvar <- seq(0,1,0.1) #Variacion de la probabilidad de 0 a 1  
2 infectadosMax <- c()  
3 for(pv in pvvar) { #Incremento en la probabilidad de vacunados  
4   for(rep in 1:10) { #Replicas del experimento  
5     agentes <- data.frame(x = double(), y = double(),  
6                           pmx = double(), pmy = double(),  
7                           dx = double(), dy = double(),  
8                           estado = character(), rest = integer())
```

Se generan los pasos por recorrer y las velocidades de los agentes hasta sus puntos meta.

[illegible]

Se generan los nuevos puntos meta y sus velocidades.

```
1   } else {  
2       pasos <- sample(5:30,1)  
3       a$rest <- pasos  
4       a$x <- a$pmx  
5       a$y <- a$pmy  
6       a$pmx <- runif(1,0,1)  
7       a$pmy <- runif(1,0,1)  
8       vx = (a$pmx - a$x)/pasos  
9       vy = (a$pmy - a$y)/pasos  
10      a$dx <- vx  
11      a$dy <- vy
```

En conclusión comparando los resultados de los diagramas violín generados, podemos observar en la figura 5 que existe un comportamiento similar al de la figura 4, pero los porcentajes de contagio son ligeramente mayores debido a que este tipo de movimiento de los agentes provoca que tengan mayor contacto con los demás.

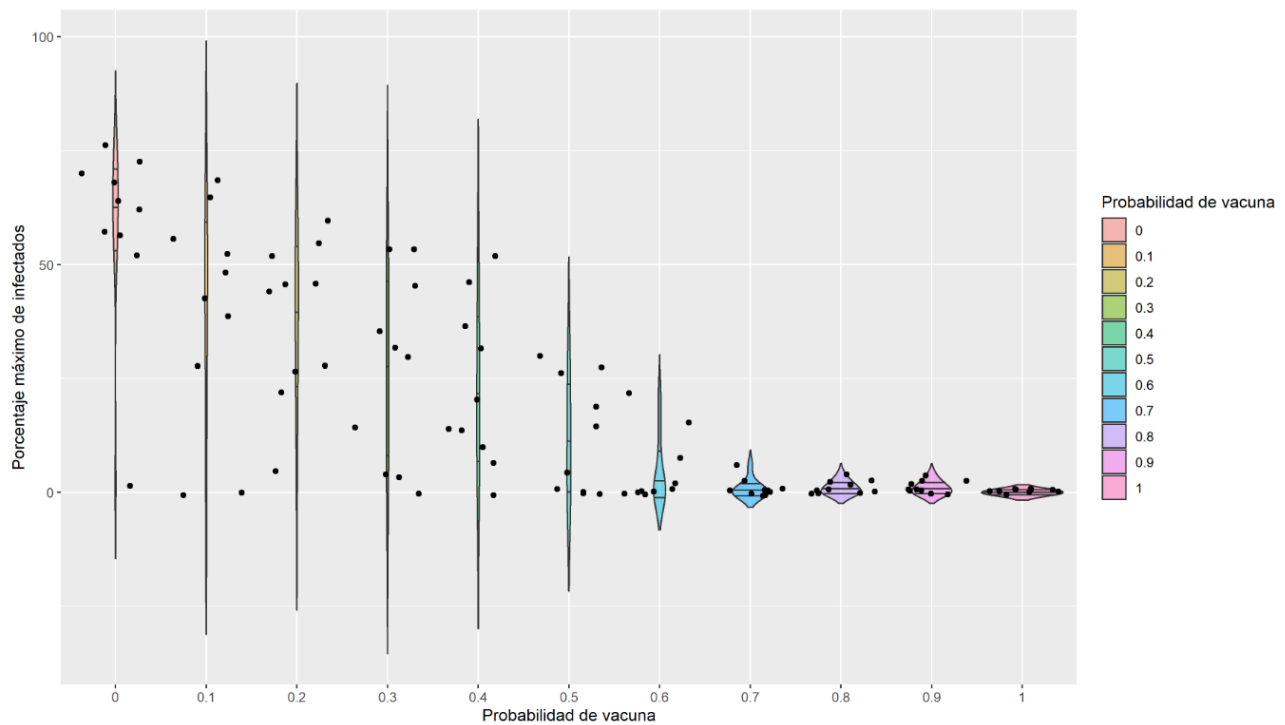


FIGURA 5. Gráficas de violín del porcentaje máximo de infectados variando la probabilidad de vacuna.

## 7. RETO 2

El reto dos consiste en que los agentes tienen amistades: si se encuentran una distancia euclídeana no mayor a  $r_a$  de un amigo suyo, se disminuyen su velocidad a la mitad por  $k_a$  iteraciones (para saludar a su amigo). Cada par de agentes tiene una amistad con una probabilidad  $p_a$ . Examina nuevamente si surgieron cambios en el efecto de  $p_v$  por esta modificación, con valores  $0 < r_a < 1$ ,  $k_a > 1$  y  $0 < p_a \ll 1$ . Se asignan los valores a elección para **ra**, **pa**, y **ka**.

```
1   r <- 0.1
2   ra <- 0.7
3   pa <- 0.4
4   ka <- 8
5   tmax <- 50
6   digitos <- floor(log(tmax, 10)) + 1
7   for (tiempo in 1:tmax) {
```

Se modifica el código para que los agentes disminuyan su velocidad cada que se encuentren a un amigo suyo.

```
1   if (d < ra) { # umbral amistad
2       p <- (r - d) / r
3       if (runif(1) < pa) {
4           for (i in 1:ka) { # iteraciones por la mitad
5               dx <- (a1$x - a2$x)/2
6               dy <- (a1$y - a2$y)/2
```

En conclusión se puede observar en la figura 6 que es similar a las figuras 5 y 4 debido a que al inicio se vacunan los agentes, por lo tanto no hay un índice elevado de infectados, pero existe cierta diferencia en la que disminuye de manera más rápida el porcentaje de infectados a pesar de que los agentes disminuyen sus velocidades cuando se encuentran con un amigo.

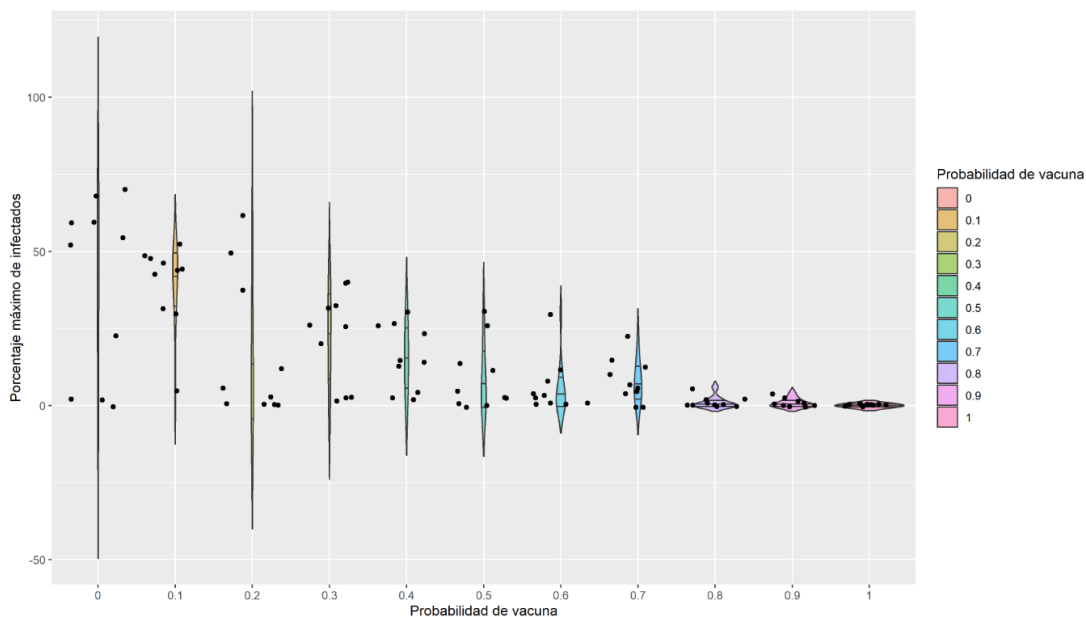


FIGURA 6. Gráficas de violín del porcentaje máximo de infectados variando la probabilidad de vacuna.

## REFERENCIAS

- [1] Llano A. Práctica 6: Sistema Multiagente., 2021. URL <https://github.com/anaeli24/simulacion/tree/master/P6>.
- [2] Schaeffer E. Sistema Multiagente., 2021. URL <https://github.com/satuelisa/Simulation/blob/master/MultiAgent/SIR.R>.
- [3] Schaeffer E. Práctica 6: Sistema Multiagente., 2021. URL <https://elisa.dyndns-web.com/teaching/comp/par/p6.html>.
- [4] The R Foundation. The R Project for Statistical Computing, 2021. URL <https://www.r-project.org/>.
- [5] Montemayor M. Tarea 6: Sistema Multiagente., 2021. URL <https://github.com/MariaMontemayor/Simul/tree/main/tarea6>.