

Reporte práctica 6

Sistema Multiagente

11 de septiembre de 2017

1. INTRODUCCIÓN

En esta práctica se implementa un sistema multiagente que simula una evolución epidemiológica. Cada agente puede ser susceptible (S), estar infectado (I) o recuperado (R). Se puede contraer la infección únicamente si se es susceptible; es decir, aquellos que se hayan recuperado de la infección son inmunes a un recontagio. Se considera que la probabilidad de contagio es proporcional a la distancia entre un agente infectado. Cada agente tiene una dirección y una velocidad de movimiento, la región sobre la que pueden moverse es una cuadrícula en donde una vez llegando a una orilla este puede continuar el movimiento desde el lado contrario, espacio geométrico conocido como torus.

2. USO DE PARALELISMO EN LA SIMULACIÓN

Para iniciar ésta simulación, se generan los agentes aleatoriamente en un cuadrado de lado $l = 1.5$. El estado de los agentes se considera susceptible para la mayoría, excepto para algunos infectados con probabilidad p_i . La velocidad y dirección de movimiento se expresan en términos de $(\Delta x, \Delta y)$, las cuáles son generadas uniformemente en el intervalo $\left[-\frac{l}{30}, \frac{l}{30}\right]$. Por facilidad se emplean enteros para representar el estado de los agentes, siendo 1 para infectados, 5 para susceptibles y 4 para recuperados. Así la creación de un agente se hace en R con las siguientes instrucciones:

```
agente=function(i){  
  e <- "S"  
  if (runif(1) < pi) {  
    e <- 1  
  }  
  return(data.frame(x = runif(1, 0, 1), y = runif(1, 0, 1),  
                    dx = runif(1, -v, v), dy = runif(1, -v, v), estado = e))  
}
```

Como la creación de cada agente es independiente, se puede realizar de forma paralela con la instrucción

```
#creacion de agentes  
agentes=parSapply(cluster, 1:n, agente)  
agentes=data.frame(t(agentes))
```

Después, se procede a iterar en el tiempo, en este ejemplo 100 veces, y determinar los nuevos contagios. Se considera que un agente j (en estado susceptible) se contagia con mayor

probabilidad p de un agente infectado i entre más cerca se encuentren. Si la distancia entre ambos agentes es mayor que un umbral $r = 0.1$, no hay posibilidad de contagio. en resumen, la probabilidad de contagio r se representa mediante

$$r = \begin{cases} \frac{r-d(i,j)}{r}, & \text{si } d(i,j) < r \\ 0, & \text{en caso contrario} \end{cases} ;$$

donde $d(i, j)$ representa la distancia euclideana entre i y j .

En vías de paralelizar el calculo de los contagios, se retoma desde una nueva perspectiva. En lugar de revisar desde un agente infectado a quienes se puede contagiar, se revisa desde un agente susceptible si éste se puede infectar. La función base es la siguiente:

```
contagiar=function(i){
  a1 <- agentes[i, ]
  if (a1$estado == 5) {# desde los susceptibles
    for (j in 1:n) {
      a2 <- agentes[j, ]
      if (a2$estado == 1) { # hacia los infectados
        dx <- a1$x - a2$x
        dy <- a1$y - a2$y
        d <- sqrt(dx^2 + dy^2)
        if (d < r) { # umbral
          p <- (r - d) / r
          if (runif(1) < p) {
            return(TRUE)
          }
        }
      }
    }
  }
  return(FALSE)
}
else{
  return(FALSE)
}
```

Lo que resta es mover a los agentes de posición y actualizar su estado de acuerdo al vector contagios. Existe la posibilidad de que aquellos agentes que ya estaban infectados se recuperen con probabilidad $p_r = 0.02$. Una vez que se determina la siguiente posición del agente, se hace el ajuste por si se sale de la frontera regrese desde el lado opuesto simulando el torus. La forma de realizar lo antes dicho es la siguiente:

```
mov_act=function(i) { # movimientos y actualizaciones
  a <- agentes[i, ]
  if (contagios[i]) {
    a$estado <- "I"
  } else if (a$estado == 1) { # ya estaba infectado
    if (runif(1) < pr) {
      a$estado <- 4 # recupera
    }
  }
}
```

```

a$x <- a$x + a$dx
a$y <- a$y + a$dy
if (a$x > 1) {
  a$x <- a$x - 1
}
if (a$y > 1) {
  a$y <- a$y - 1
}
if (a$x < 0) {
  a$x <- a$x + 1
}
if (a$y < 0) {
  a$y <- a$y + 1
}
return( c(x=a$x,y=a$y,dx=a$dx,dy=a$dy,estado=a$estado) )
}

```

De este modo, las actualizaciones correspondientes d cada agente se pueden realizar mediante las siguientes instrucciones:

```

registerDoParallel(cl)
agentes=foreach(i=1:n,.combine=rbind) %dopar% mov_act(i)
stopImplicitCluster()

```

Se realizó un diseño de experimentos para determinar la eficacia del uso de paralelismo en la simulación. Se probaron diez réplicas para $n \in \{20, 50, 100, 200, 300\}$ agentes. La Figura 2.1 muestra los diagramas de bigotes correspondientes a la simulación usando paralelismo y a la implementación secuencial.

3. SIMULACIÓN DE UNA EPIDEMIA

La Figura 3.1 ilustra la evolución de la epidemia durante el tiempo. Los cuadros azules representan a los agentes susceptibles y los círculos rojos a los infectados. Observe como hay un pequeño porcentaje de agentes infectados al inicio (Figura 3.1a). En la Figura 3.1b, aparecen algunos agentes que se ha recuperado, representados por un triángulo de color verde. Debido a la interacción de los agentes, estos se van infectando rápidamente y como se aprecia en la Figura 3.1d, ya hay un gran porcentaje de infectados y solo uno susceptible. Sin embargo, como la recuperación ofrece inmunidad, la población de agentes se va poblando de individuos recuperados y como puede notarse en pasos avanzados (véase Figura 3.1i) ya no hay agentes susceptibles al contagio. A partir de aquí puede notarse como la probabilidad de recuperación se ilustra hasta llegar al final de la simulación en donde algunos se recuperaron (Figura 3.1l). Si continúa la simulación, es de esperar que la infección desaparezca por completo. Esta situación es similar a la presentada en algunas de las epidemias más grandes de la humanidad, como la peste bubónica del siglo XIV en Europa, en donde tras el desarrollo de inmunidad eventualmente desapareció el riesgo de contagio. Para resumir el comportamiento general, observe la Figura 3.2 en donde se aprecia el porcentaje de agentes infectados durante el transcurso del tiempo de simulación. Se observa un rápido crecimiento de la infección al

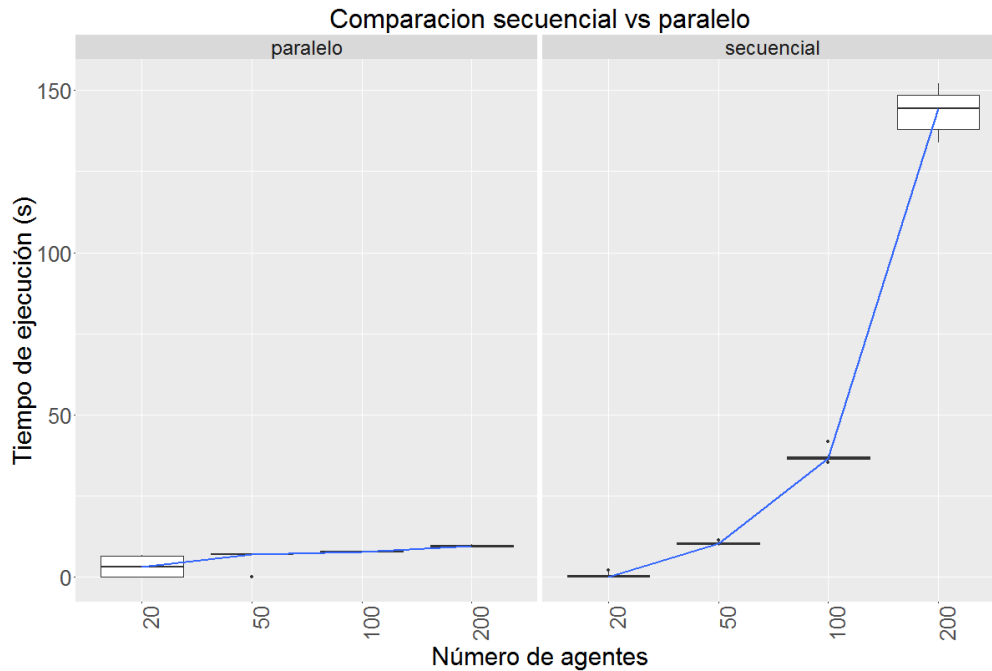
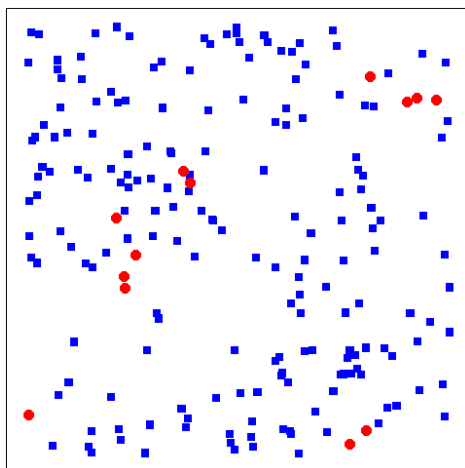


Figura 2.1: Comparación entre la implementación paralela y secuencial de la simulación de una epidemia

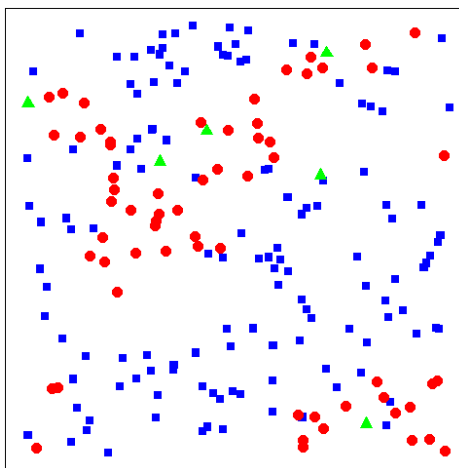
inicio, seguido de un período de estabilidad en donde hay un gran número de individuos infectados, podría decirse que el período más peligroso de la enfermedad. Por último, se aprecia el efecto del desarrollo de inmunidad sobre los agentes y como la epidemia pierde fuerza.

4. RETO 1: VACUNACIÓN PREVENTIVA

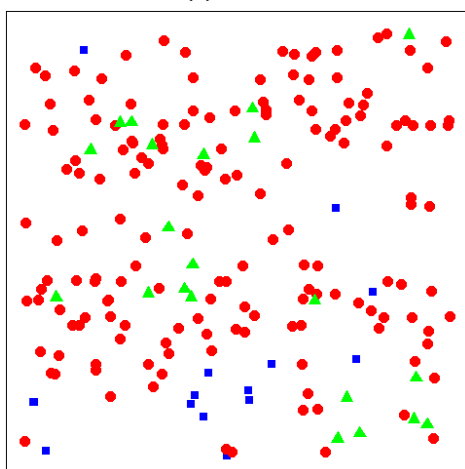
Como es bien sabido, una campaña de vacunación bien aplicada es vital para el control de una epidemia. Con una vacuna, se puede dar inmunidad a un individuo sin tener que infectarse y recuperarse para desarrollarla. Es de esperar, que la evolución de una epidemia sea más controlada si una parte de la población se vacunó preventivamente. Para agregar esta característica a la simulación, incluimos en la generación de los agentes una probabilidad p_v de que sea vacunado. Así, cuando iniciemos la epidemia estos individuos serán inmunes y no podrán propagar la infección, para simplificar supondremos que éstos aparecen como recuperados al inicio. La Figura 4.1 muestra el efecto de la vacunación preventiva de los agentes. Observe como hay una clara disminución de los porcentajes de contagios cuando la probabilidad de que los agentes sean vacunados aumenta. En azul aparece la evolución de la epidemia cuando no hay vacunación. Podemos observar el claro beneficio de la vacunación preventiva para evitar contagios.



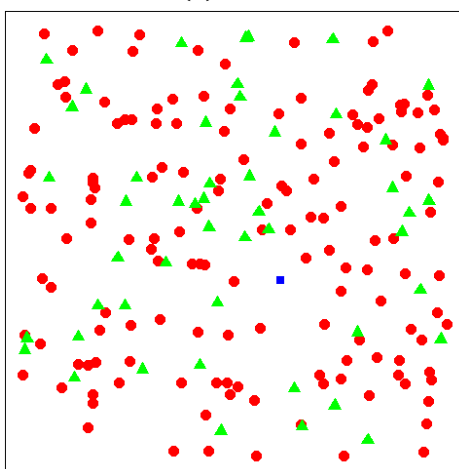
(a) Paso 1



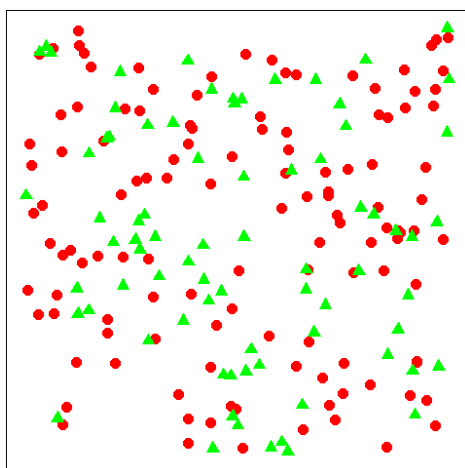
(b) Paso 8



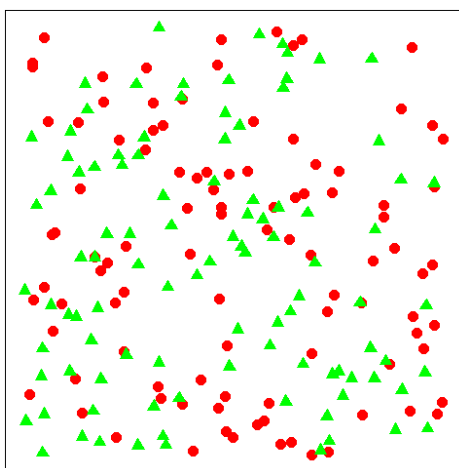
(c) Paso 18



(d) Paso 26



(e) Paso 36



(f) Paso 47

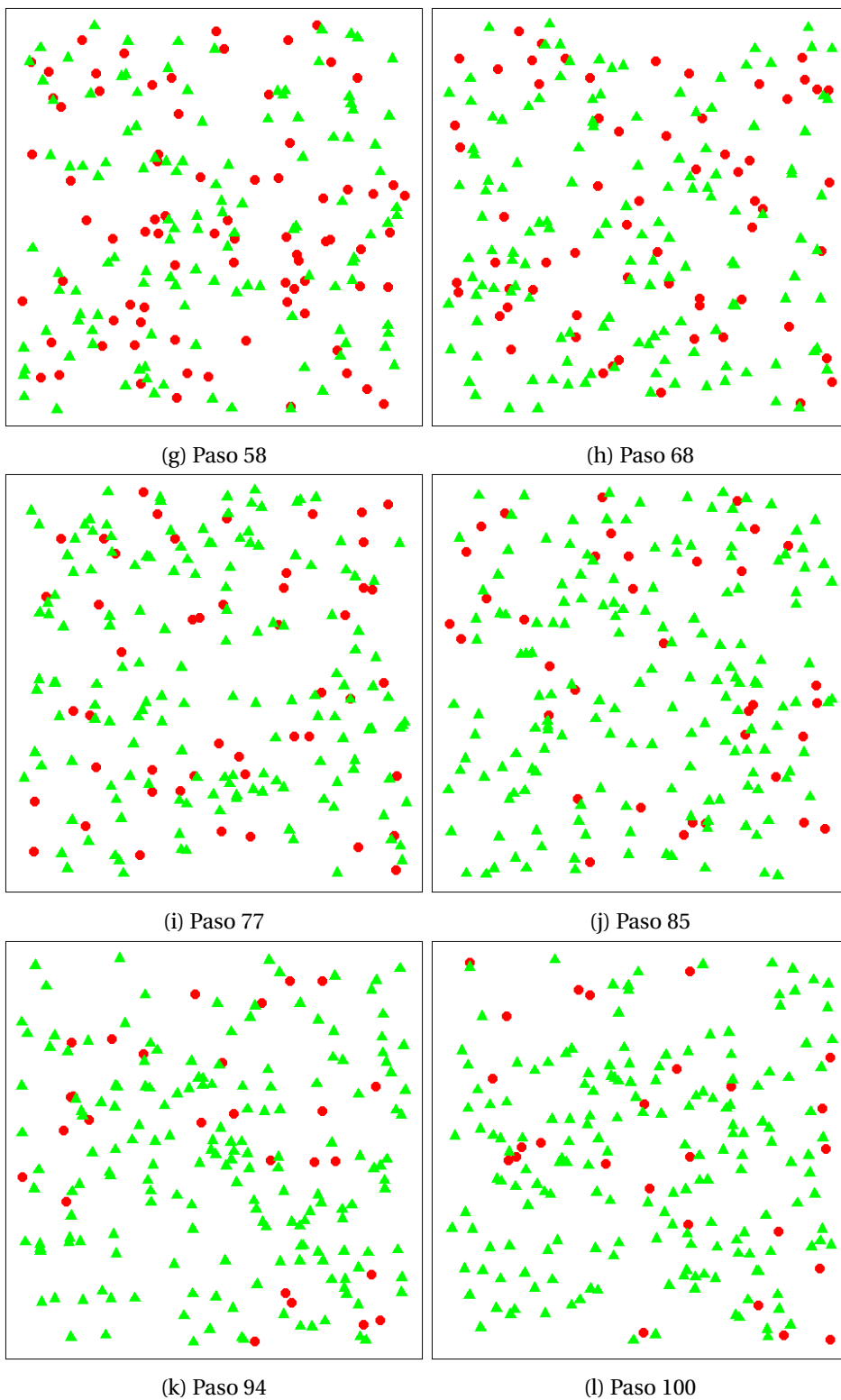


Figura 3.1: Evolución de la epidemia durante el transcurso del tiempo.

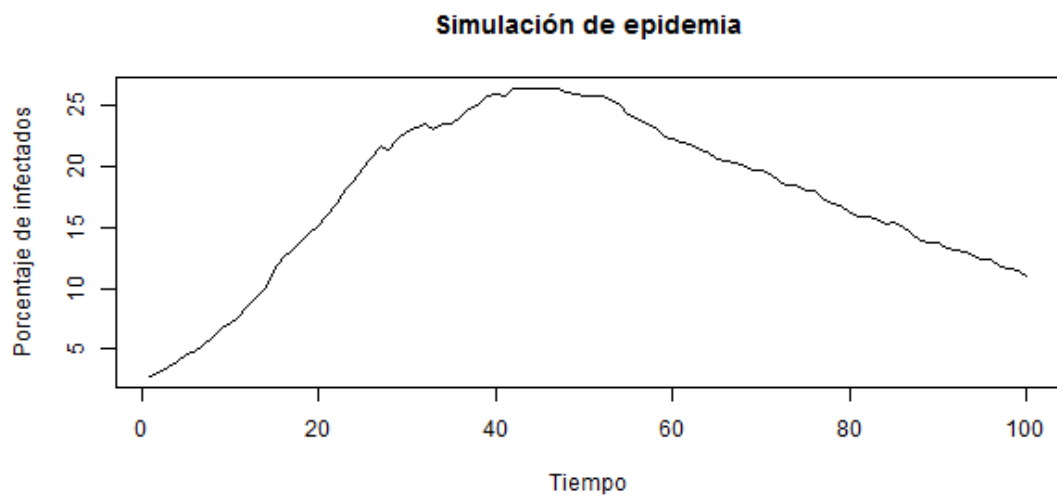


Figura 3.2: Porcentaje de agentes infectados a través del tiempo

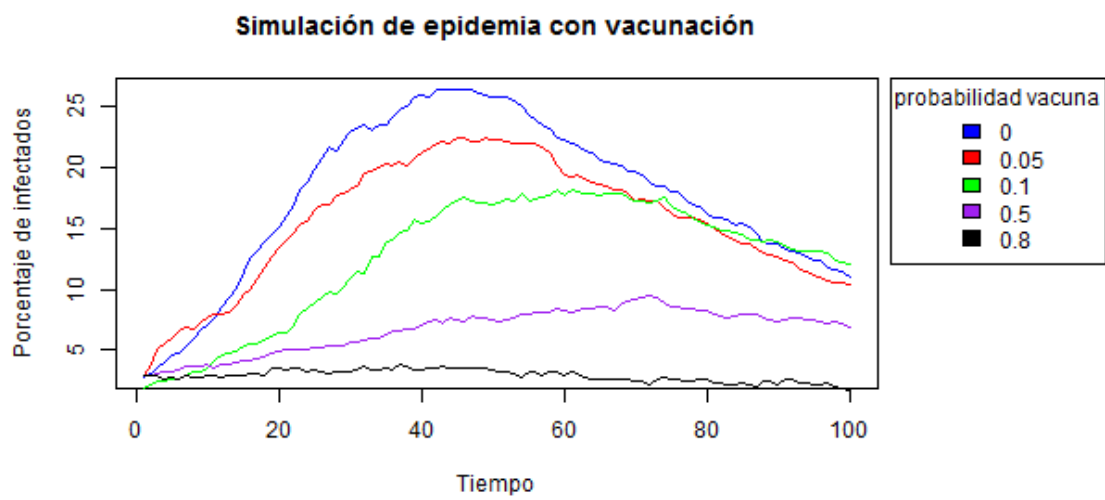


Figura 4.1: Efecto de la vacunación preventiva durante la epidemia

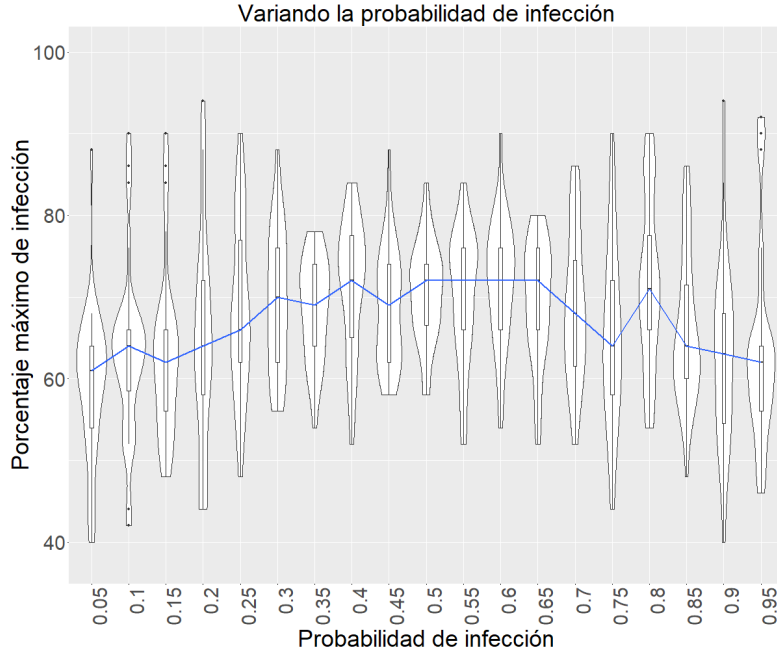


Figura 5.1: Efecto de la probabilidad de infección p_i en la creación de los agentes

5. RETO 2: EFECTO DE LA PROBABILIDAD DE INFECCIÓN p_i

El último reto consiste en estudiar el efecto que tiene la cantidad de agentes que están infectados en un inicio. Cuando comenzamos la simulación, los agentes están infectados con probabilidad p_i y además recuerde que también pueden ser vacunados con probabilidad p_v ; el resto son susceptibles a contagiarse. Para medir el efecto de esta probabilidad se diseña un experimento en donde se considera que p_i puede tomar valores en $[0.05, 0.1, \dots, 0.9, 0.95]$. Para disminuir errores por variabilidad se realizan 30 réplicas de cada tratamiento. La probabilidad de vacunación es del 10% y el resto de los parámetros son iguales a los mencionados en la sección 3. Como medida de comparación, se considera el porcentaje máximo de infección ocurrido durante cada simulación. La Figura 5.1 muestra los diagramas de bigotes y violines correspondientes al experimento. La línea azul une las medianas de cada valor de p_i .

Una prueba de Kruskal y Wallis con un valor p de 1.9×10^{-10} indica el efecto significativo de la probabilidad p_i sobre el porcentaje máximo de infección ocurrido. El comportamiento se asemeja a una campana, era de esperar que si había mucha infección en un comienzo, la cantidad máxima de individuos infectados (antes de recuperarse) fuera alta. El efecto es sin duda a causa de la inmunidad, pues entre mayor número de infectados, también hay mayor probabilidad de recuperarse. Como al recuperarse el agente se hace inmune, el porcentaje máximo de infección debe disminuir. Por tanto se observa un crecimiento mayor de la epidemia en los momentos en que la probabilidad de estar infectado o ser susceptible es casi la misma.