

Tarea 2: Juego de la vida

Eduardo Navarro

Septiembre 2021

1. Introducción

En base a lo visto en la practica 2 y con ayuda de una simulación se procederá a calcular la probabilidad del éxito de vida a partir de la probabilidad inicial dada para poder observar la influencia de los estados iniciales.

2. Desarrollo

Se utilizó el lenguaje de programación en R y se trabajó en gran medida en base al código de la Dra. Elisa [3]. Al código se le hicieron modificaciones correspondientes para remover el código no útil para los propósitos de esta practica. Se siguieron las recomendaciones vistas en clase y en el canal de discord de agregar “for” adicionales contenidos uno dentro de otro para poder realizar replicas a diferentes valores de probabilidad. En base a los objetivos propuestos por la tarea se manipularon variables como las repeticiones, la iteración, la probabilidad y el número de dimensiones de la matriz [4]. Se crearon los data.frame correspondientes para la manipulación de los datos obtenidos de los experimentos con ayuda del comando aggregate [1] y la posterior creación de las graficas con ayuda de ggplot2 [5]. Se modificaron los nombres y se editaron las dimensiones de los ejes con guía de un video [2].

Probabilidad inicial	Vida
0.1	0
0.1	0
0.1	0
0.1	1
0.1	0

Tabla 1: Ejemplo de datos obtenidos mediante data frame

En la tabla 1 se observa un ejemplo de los valores obtenidos por el primer data frame. Los resultados de los experimentos se organizan en 5 valores diferentes de probabilidad a 30 repeticiones cada uno, esto nos da un total de

150 valores para esta información en total. El 1 en vida indica que se formo un patron repetible y por lo tanto no desaparece.

Con ayuda de mis compañeros se sugirió el metodo de utilizar un 2do data frame. Natalia Perez sugirió utilizar la función `aggregate` [1] la cual nos ayudo a procesar los datos dentro del segundo data frame obteniendo de esa forma los promedios de cada probabilidad inicial en relación a la vida que se creaba ya que como tenian valor de 1 nos permitieron realizar este procedimiento de esta forma.

Probabilidad inicial	Vida
0.1	0.0333
0.3	0.0333
0.5	0.0666
0.7	0
0.9	0.0333

Tabla 2: Promedios obtenidos para cada valor de probabilidad inicial de dimension 10

En la tabla anterior (2) se ven los datos obtenidos los cuales se procedieron a graficar con la librería `ggplot2` de R [2, 5] obteniendo la siguiente grafica

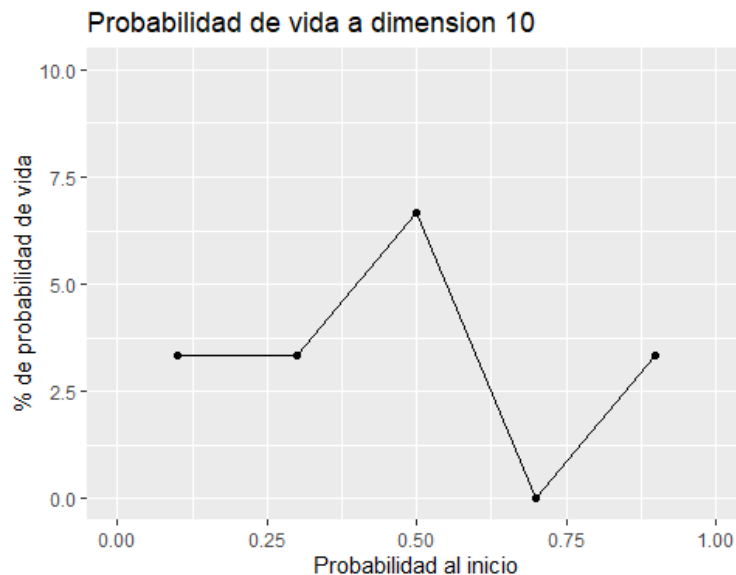


Figura 1: Grafica que se obtiene a partir de la tabla 2

Los valores de la tabla 2 se ven reflejados en la figura 1 y se puede observar

como la región central destaca muy poco por encima de las otras. la baja probabilidad se debe a la pequeña dimension con la que se trabajo. Se realizó el experimento varias veces más para revisar resultados y se obtuvo como máximo un 10 % de probabilidad de vida en diferentes probabilidades de vidas iniciales significando esto un máximo de 3 vidas creadas.

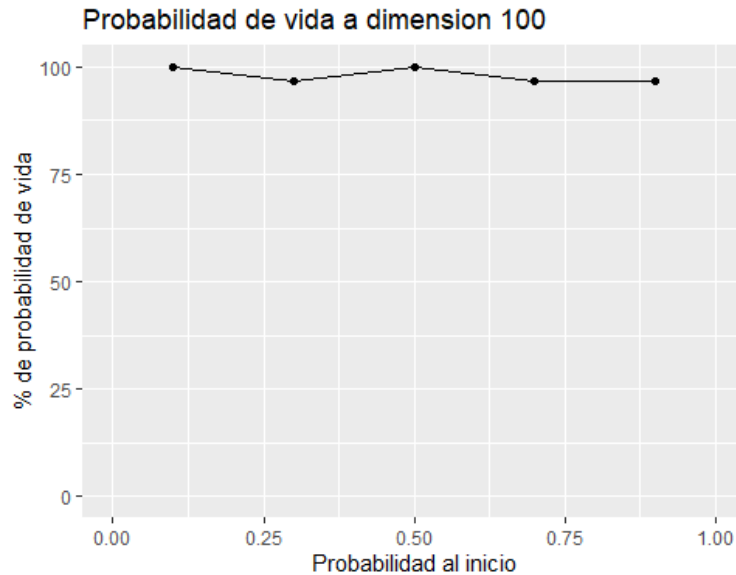


Figura 2: Experimento realizado a dimensiones muy grandes

La figura 2 nos muestra un ejemplo de los resultados obtenidos para dimensiones grandes, en él se puede observar que la probabilidad de vida se obtiene de casi un 100 % en todas las probabilidades iniciales debido a que hay más espacio para que esta se genere. A continuación, mi código.

```
library(parallel)
library(ggplot2)
dim <- 10
num <- dim^2
p <- c(0.1, 0.3, 0.5, 0.7, 0.9)
datos = data.frame()
datos1 = data.frame()
for (inicio in p) {
  for (rep in 1:30) {
    actual <- matrix(round(runif(num) < p), nrow=dim, ncol=dim, byrow=TRUE)
    paso <- function(pos) {
      fila <- floor((pos - 1) / dim) + 1
      columna <- ((pos - 1) %% dim) + 1
      vecindad <- actual[max(fila - 1, 1) : min(fila + 1, dim),
```

```

        max(columna - 1, 1): min(columna + 1, dim)]
    return(1 * ((sum(vecindad) - actual[filas, columna]) == 3))
}

cluster <- makeCluster(detectCores() - 2)
clusterExport(cluster, "dim")
clusterExport(cluster, "paso")

for (iteracion in 1:50) {
  clusterExport(cluster, "actual")
  siguiente <- parSapply(cluster, 1:num, paso)
  vivos = sum(siguiente)
  cat(inicio, rep, iteracion, vivos, '\n')
  actual <- matrix(siguiente, nrow=dim, ncol=dim, byrow=TRUE)
}
if (vivos == 0) {
  vivos = 0 # todos murieron
  print("Ya_no_queda_nadie_vivo.")
} else { vivos = 1
  print("hay_vida")
}

respuesta = c(inicio, vivos)
datos = rbind(datos, respuesta)
stopCluster(cluster)
}

}
names(datos) <- c("probabilidad", "vida")
datos1 = aggregate(vida ~ probabilidad, datos, mean)
ggplot(datos1, aes(x = probabilidad, y = (vida*100))) +
  geom_line() + geom_point() +
  labs(x="Probabilidad_al_inicio",
       y="%de_probabilidad_de_vida",
       title = "Probabilidad_de_vida_a_dimension_10") +
  scale_x_continuous(limits = c(0,1)) +
  coord_cartesian(ylim = c(0,10))

```

3. Conclusiones

En esta práctica aprendimos a agregar en cuenta la importancia de estados pasados para la generación de nuevos estados a partir de ciertas probabilidades dadas. A la vez que las interacciones entre vecinos tienen un efecto sobre los resultados obtenidos. La dimensión es un factor grande que determina la

probabilidad de vida final independientemente de la inicial ya que a mayor dimensión mayor probabilidad de vida hay. La razón de que se obtuvieran graficas relativamente diferentes para cada experimento dentro del mismo rango de dimensión se puede deber a que son probabilidades iniciales y aunque con menor frecuencia, pueden llegar a los mismos valores de poder generar vida dentro del experimento. En lo personal me siento muy contento con el resultado ya que se trabajó exhaustivamente para la realización del presente trabajo.

Referencias

- [1] Becoming a Data Scientist. How to use the aggregate function in r, 2020. URL <https://www.youtube.com/watch?v=RGgkBtaczMY>.
- [2] Statistics Globe. Set axis limits in ggplot2 r plot (3 examples) — how to adjust the range of axes, 2019. URL <https://www.youtube.com/watch?v=u2qtt-UFIWs>.
- [3] Elisa Schaeffer. Repositorio github, simulation, cellularautomata. *GitHub*, 0(0):0, 2020. doi: 0.0/.
- [4] Elisa Schaeffer. Práctica 2: autómatas celulares. <https://elisa.dyndns-web.com/teaching/comp/par/p2.html/>, 2021. [Online; accessed 29-August-2021].
- [5] Hadley wickham. Create elegant data visualisations using the grammar of graphics, 2019. URL <https://ggplot2.tidyverse.org/index.html>.