

Reporte práctica 1

Movimiento browniano

1. INTRODUCCIÓN

El movimiento browniano se refiere al cambio de posición de una partícula uniformemente al azar.

En la práctica se simplifican a movimiento en dónde los pasos miden lo mismo, teniendo como posición inicial el origen de un plano cartesiano. ^[1]

2. SIMULACIÓN

Se inicia generando un número pseudoaleatorio mediante un *runif* y mediante un *for* se moverá una cantidad de pasos deseados.

La simulación ejemplo cuenta que tan lejos se mueve una partícula con respecto al origen, midiendo la distancia por dos vías: euclidiana y manhattan.

Para la tarea se desea saber cuántas veces la partícula regresa al origen, para esto se declara un vector origen en donde se repite cero la cantidad *dimensión*. Este vector se utilizará para comparar las posiciones de la partícula con el origen.

La comparación se realiza mediante un *if*, en dónde si se cumple la condición, se sumará uno a la variable *contador* antes inicializada en cero.

Se varía las dimensiones de uno a ocho, la cantidad de pasos de 200 a 1000 con saltos de 200 y la duración de la simulación de 200 a 600 variando en 200.

Se utilizó un *for* para hacer cinco repeticiones del código y poder analizar la variación de la cantidad de regresos al origen.

Las modificaciones al código original se muestran a continuación:

```
repetir <- seq(200,1000, 200)
duracion <- seq(200,600,200)

library(parallel)

cluster <- makeCluster(detectCores() - 1)
clusterExport(cluster, "duracion")
clusterExport(cluster, "repetir")
datos <- data.frame()
```

```

for(replica in 1:5){
for (dimension in 1:8) {
  clusterExport(cluster, "dimension")
  resultado <- parSapply(cluster, 1:repetir,
                        function(r) {
                          pos <- rep(0, dimension)
                          origen=rep(0,dimension)

                          contador=0
                          for (t in 1:duracion) {
                            cambiar <- sample(1:dimension, 1)
                            cambio <- 1
                            if (runif(1) < 0.5) {
                              cambio <- -1
                            }
                            pos[cambiar] <- pos[cambiar] + cambio
                          }
                          if (all(pos==origen)) {
                            contador=contador +1
                          }

                          return(contador)
                        })
  res<-cbind(replica, duracion, repetir,dimension, sum(resultado))
  datos <- rbind(datos, res)
  #resultado<-cbind(resultado,res)
}

}

stopCluster(cluster)
colnames(datos)<-c("Replica",
"Pasos", "Repeticiones", "Dimension", "Regresos")
datos$Repeticiones<-as.factor(datos$Repeticiones)
datos$Dimension<-as.factor(datos$Dimension)
datos$Pasos<-as.factor(datos$Pasos)

library(ggplot2)
ggplot(data=datos, aes(x = Dimension, y= Regresos, color=Pasos)) +
  geom_boxplot(position =
position_dodge(1))+facet_grid(Repeticiones~.)+
  theme_bw()+ ylab("Veces en origen")
ggsave("P1_plot.png")

```

3. RESULTADOS

En la figura 1 se muestran el número de veces que la partícula regresa al origen en función de la cantidad de pasos, dimensiones y la duración de la simulación.

Se puede observar que a partir de tres dimensiones la cantidad de veces que regresa al origen es nula, sin importar si varían los otros parámetros.

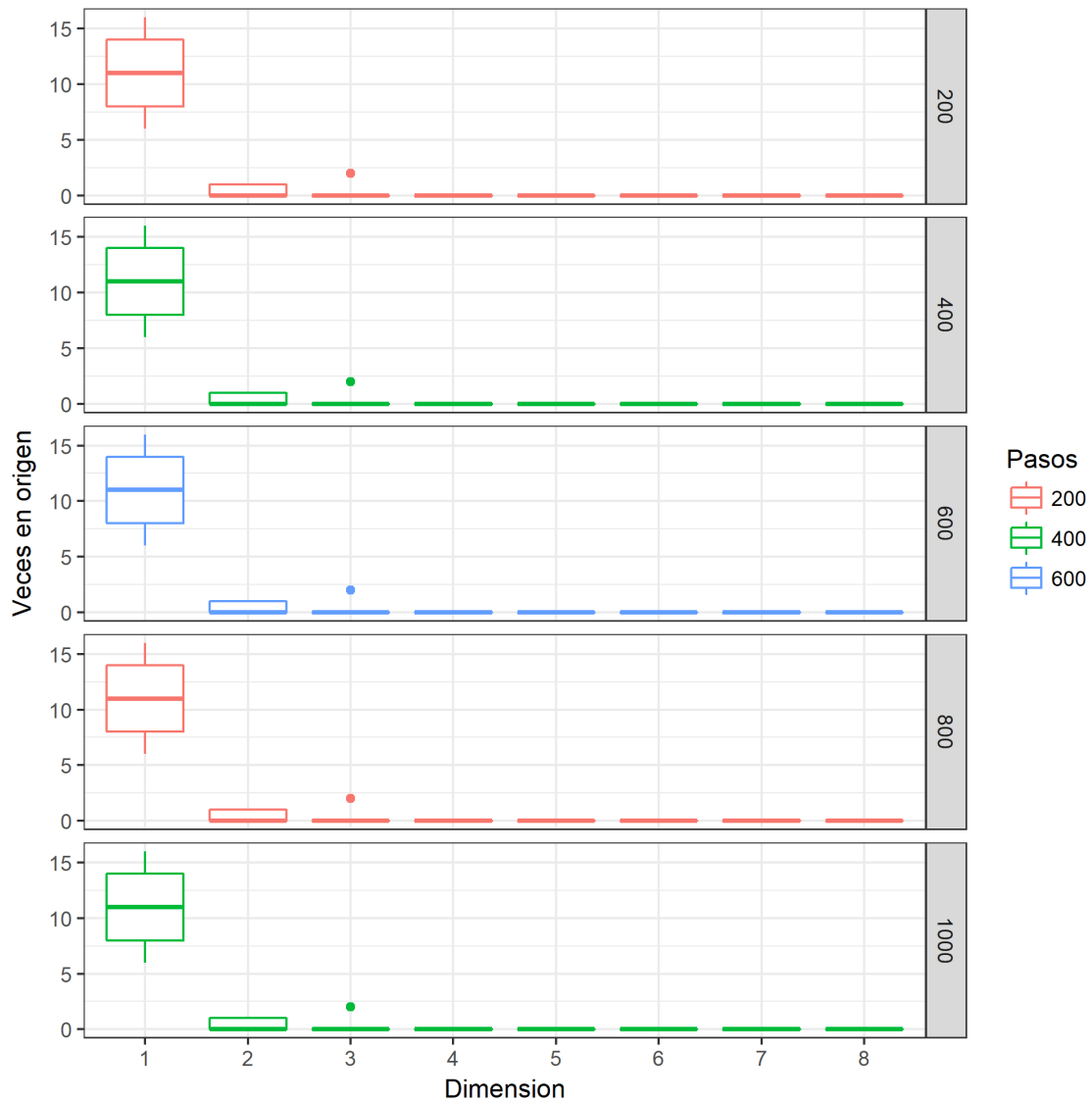


Figura 1. Gráfica caja-bigote de las veces que regresa al origen variando cantidad de pasos (200, 400 y 600), dimensiones (1:8) y duración (200, 400, 600, 800 y 1000).

4. CONCLUSIONES

Mayor a tres dimensiones la probabilidad de que la partícula regrese al origen es nula.

El factor que más influye en que la partícula regrese al origen es la cantidad de dimensiones.

Al variar el número de pasos y la duración de la simulación no existe diferencia en el número de veces en que regresa al origen.

5. REFERENCIAS

[1]Elisa.dyndns-web.com. (2017). P1-R paralelo-Schaeffer. [online]Available at: <http://elisa.dyndns-web.com/teaching/comp/par/p12.html> [Accessed 30 Oct. 2017].