Práctica 1

Ricardo Daniel Parga Montemayor

5 de febrero de 2018

1.- Movimiento Browniano

En esta práctica se nos presenta una situación de dos movimientos Euclidiana y el Manhattan que nos sirven como ejemplo para poder realizar la operación que nos pedirán en el apartado dos. El ejemplo que nos proporciona la práctica es referente a una partícula que es simulada para generar un movimiento aleatorio en un cierto sector o más bien dicho una distancia mayor de movimiento, en una serie de desplazamientos que están regidas por el número de dimensiones que se les asignen y para concluir es analizado para estudiar si es más eficiente el tiempo de tardado por los comandos de paralelismo y sin paralelismo.

2 .- Descripción de la tarea

- Examina de manera sistemática los efectos de la dimensión en el la probabilidad de regreso al origen del movimiento Browniano. Recuerda verificar que el número de pasos de la caminata o el número de repeticiones del experimento no estén causando un efecto visible en ilustraciones de esta probabilidad.
- El primer reto es estudiar de forma sistemática y automatizada el tiempo de ejecución de una caminata (en segundos) en términos del largo de la caminata (en pasos) y la dimensión.
- El segundo reto es realizar una comparación entre una implementación paralela y otra versión que no aproveche paralelismo.

3.- Tarea 1

En la tarea lo primero que tenemos que darnos cuenta es hacer que la posición donde se esté repitiendo el número aleatorio sea detectada en una posición origen y que sea compatible en todas las dimensiones que vamos a declarar a su vez ese resultado sea almacenado en una variable que llamaremos contador.

```
# if (sum(pos == origen) == dim) { contador <-  contador + 1  }
```

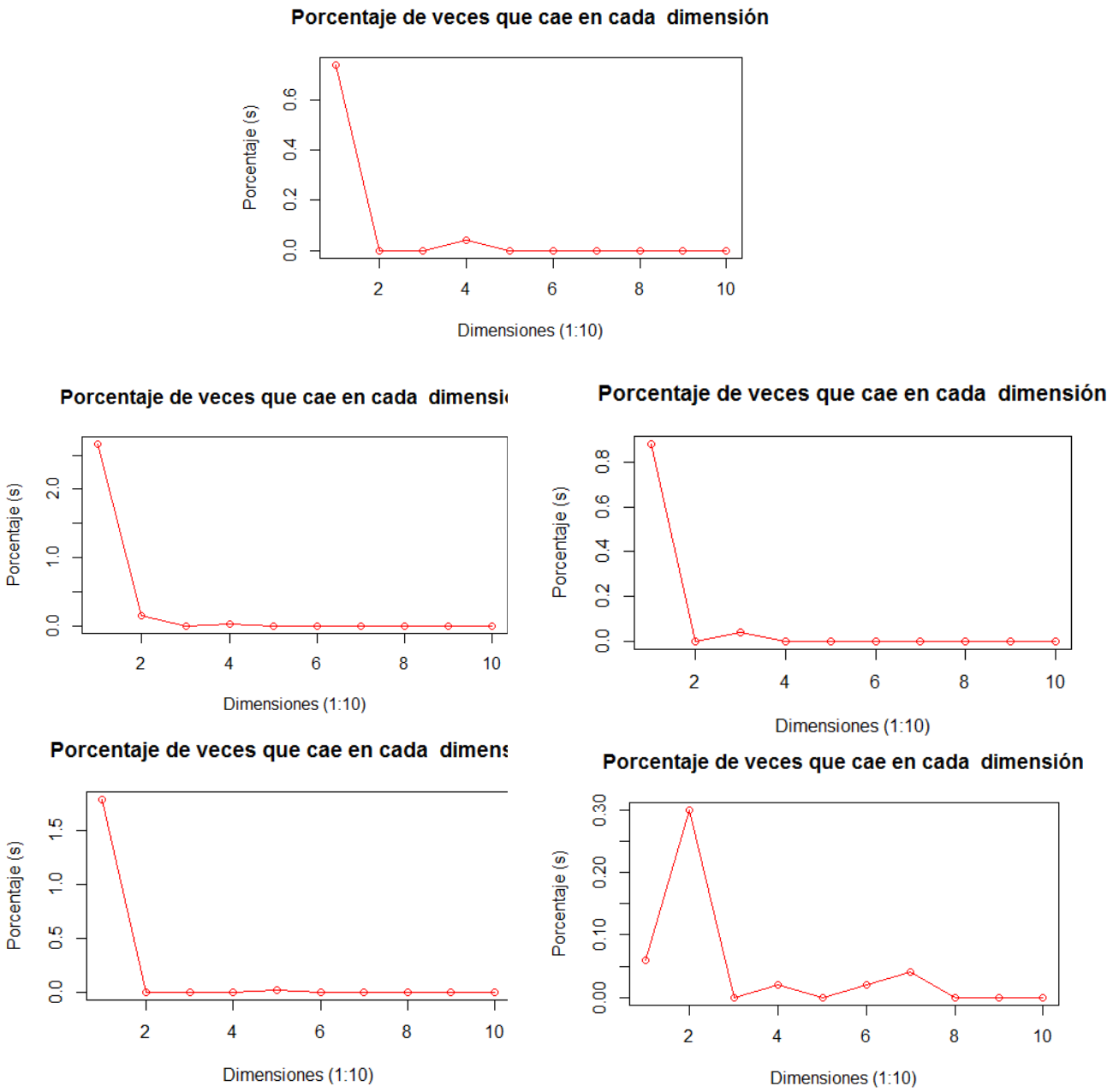
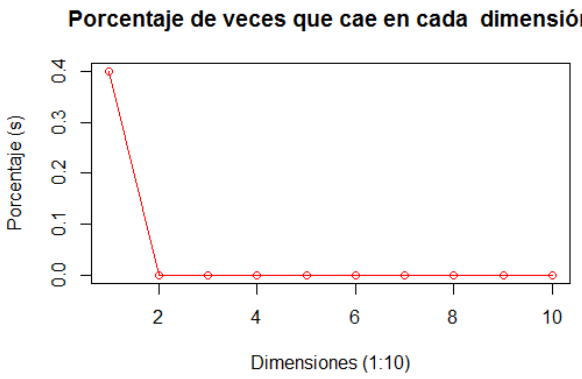
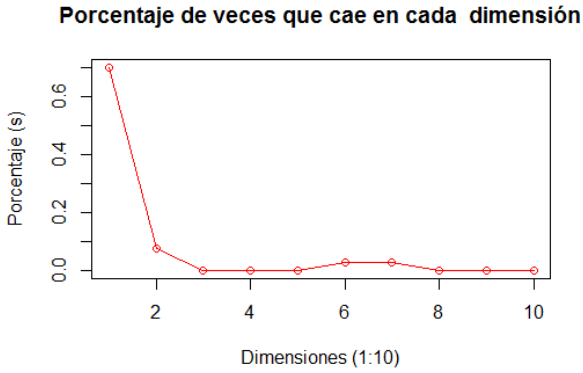


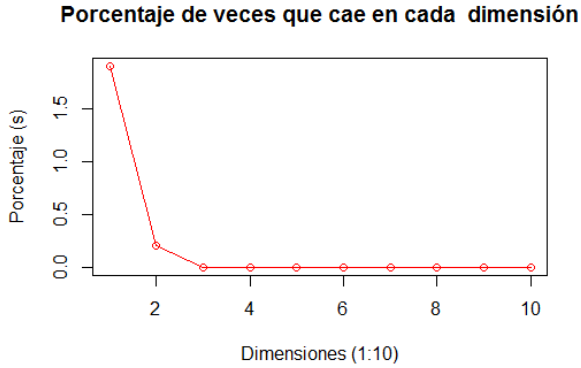
Figura 1. Se muestran en las gráficas el porcentaje de cada vez que pasa por el origen, al mismo tiempo se observa el aumento en cada dimensión hasta llegar a un total de 10, Se repite en una secuencia de 5 eventos cuyas variables son 10 dimensiones a 5000 pasos cada una.



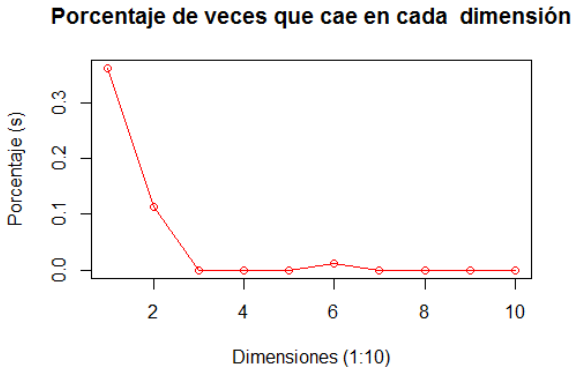
a)500



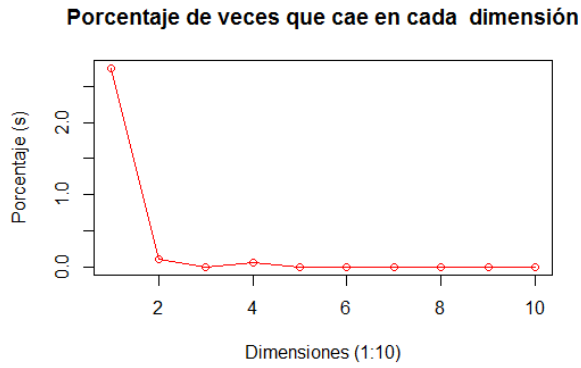
e)4000



b)1000



f)8000



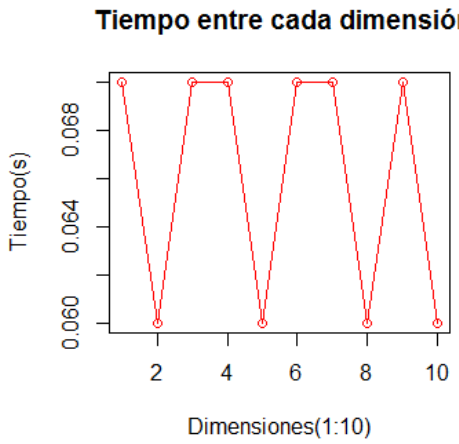
c)2000

Figura 2. Grupo de gráficas que muestran el porcentaje de que pueden caer en el origen todas con una dimensión ascendente de 1 hasta llegar a 10, todas con un valor de pasos de 500, 1000, 2000 ,4000 y 8000 respectivamente a cada gráfica.

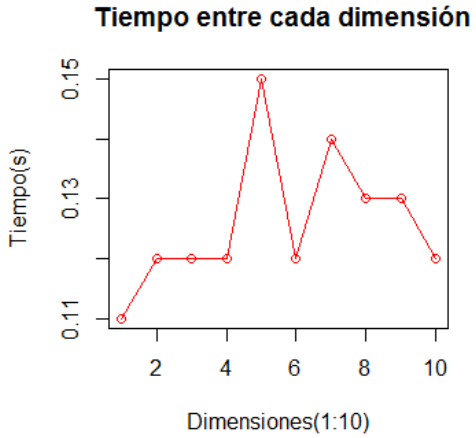
4.- Reto 1.

Para poder realizar el reto 1, primero hay que crear una rutina que repita el programa caminata que tiene cargado las instrucciones de cuantas veces pasara por el origen, a su vez las dimensiones y los números de pasos incrementando en número de veces que se va a repetir todo eso almacenado en una variable que llamaremos elapsedtime que es la que nos mostrará el tiempo que a transcurrido en cada operación y será graficado en un punto de asignado a cada dimensión.

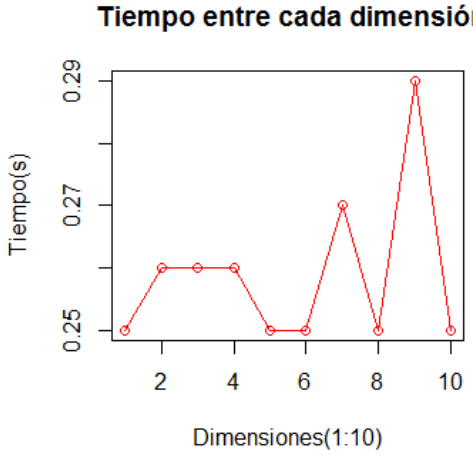
```
# for (i in 1:num) { time=system.time(caminata(dim, 20000,ed.orig))

elapsedtime[dim]=time[3]

dim=dim+1}
```



a)5000



b)10000



c) 20000

Figura 3.- Se muestra en tres casos el tiempo que se tarda en cada dimensión con la diferencia entre cada una de pasos a 5000, 10000, 20000 respectivamente a cada gráfica.

5.- Reto 2.

En este reto se nos pide comprobar la eficiencia del paralelismo para poder realizar en menor tiempo las operaciones que nos tomarían una mayor prolongación de tiempo En la tabla siguiente se puede apreciar que se realizó en varias dimensiones (3, 50, 100) para ver el tiempo que le tomaría realizar la operación. Es más que apreciable que el tiempo se redujo en la mitad al momento de correr el programa con las instrucciones del paralelismo, podemos concluir que gracias a la herramienta que nos enseña en estas prácticas es más que posible eficientar el tiempo de carga que toman los programas de largas repeticiones.

	Tiempo paralelo	*Tiempo no paralelo*
Dimensión 3	6.81166 secs	10.83622 secs
Dimensión 50	1.79563 mins	3.429982 mins
Dimensión 100	5.08228 mins	9.545352 mins

6.- Conclusiones

En la tarea podemos concluir que al momento de ir de una dimensión a una mayor la probabilidad que pasen por el origen es menor ya que tienes un mayor de numero de combinaciones que puede tomar por cada dimensión agregada, por consecuencia el porcentaje disminuye por las casi nulas intersecciones que se observan a partir de la tercera dimensión en adelante como se observa en la figura 1. Otro factor es el número de pasos ya que si a mayor número de pasos tendremos mayor posibilidades de volver al origen pero como en el caso de las dimensiones aunque tengamos mayor número de pasos las dimensiones influyen al momento de regresar.

El tiempo de tardado está relacionado al número de pasos que se generan al realizar la operación y al número de dimensiones como se observa en la gráfica 2, en las ultimas dimensiones el tempo es mucho más considerable que al inicio de las primeras dimensiones.

Como ya se puedo concluir en la sección del reto 2 al momento de paralelizar nuestro programa el tiempo es reducido a la mitad en consecuencia enficionando las jornadas de trabajo que se pueden presentar al momento de realizar operaciones aun mayores.

7.- Especificaciones de equipo

Modelo del sistema Inspiron 5420, fabricada por Dell Inc, procesador Intel(R) Core(TM)i5-3210MCPU @2.50Hz 2.50Hz, memoria instalada (RAM) 8.00 GB(7.86 GB utilizable) , tipo de sistema operativo de 64 bits procesador x64, edición de Windows 10 Pro, tarjeta grafica Intel(R) HD Graphics 4000.