1.- Práctica 3: Teoría de colas

En esta práctica se utilizara la herramienta matemática de teoría de colas que es utilizada para procesar líneas de tiempo de espera, dándole el ordenamiento de distintos trabajos con diferentes recursos generando distintos tiempos de ejecución.

2.- Descripción de la tarea

- Examina como las diferencias en los tiempos de ejecución de los diferentes ordenamientos cambian cuando se varía el **número de núcleos asignados** al clúster, utilizando como datos de entrada un vector que contiene primos grandes, descargados de https://primes.utm.edu/lists/small/millions/ y no-primos del mismo tamaño. Investiga también el efecto de la proporción de primos y no primos en el vector igual como la magnitud de los números incluidos en el vector.
- El primer reto es argumentar, apoyándose con visualizaciones, las posibles causas a las diferencias en los tiempos de ejecuciones observadas en la tarea base y razonar cómo y por qué detrás de las diferencias observadas.

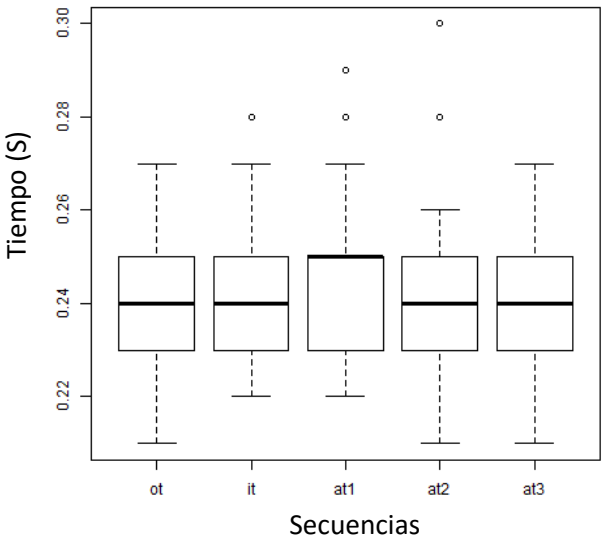
3.- Tarea 1

Lo que se buscó en esta práctica es calcular los diferentes tiempos de ejecución de grandes series de números primos y no primos a diferentes cantidad de núcleos, que realizan secuencias de menor a mayor, mayor a menor y aleatorio.

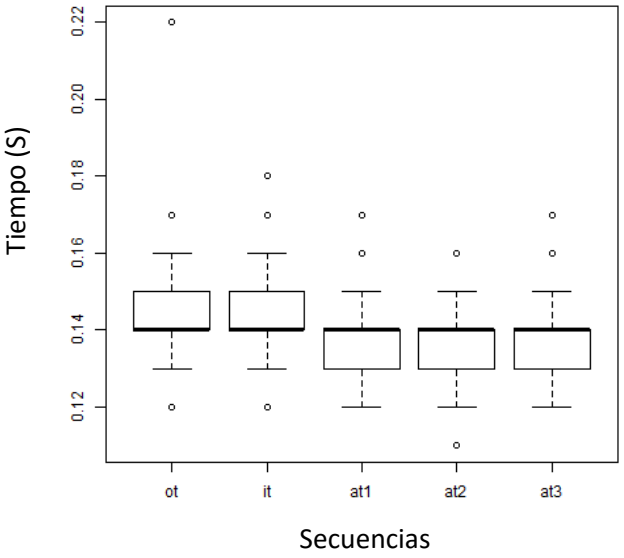
El primer paso que se realizo es mandar a llamar el archivo *prime1*, que contiene una serie grande de números primos, se realizaron modificaciones dando como resultado un vector.

```
Nprimos = read.csv("primes1.txt", sep=" ", header=FALSE) #eliminar encabezado de txt

Nprimos$V1 = NULL #borrar primer columna[nombre]

Nprimos$V10 = NULL #borrar ultima columna[nombre]

m = as.matrix(Nprimos) #crear matriz de ...

v = as.vector(t(m)) #crear vector de una matriz
```

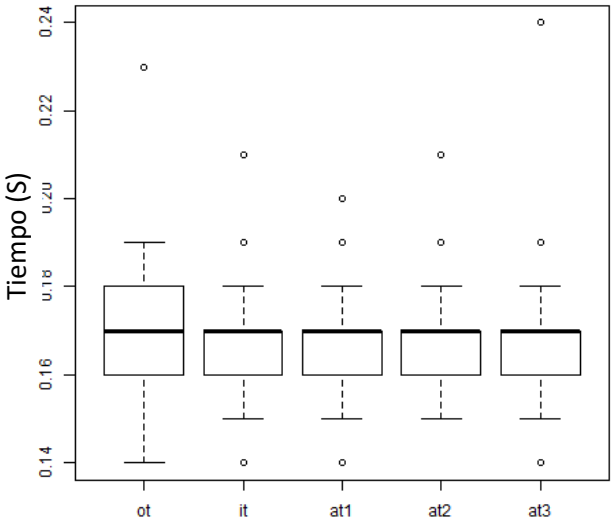
Con dicho vector se tomaron una serie de posiciones guardados en vectores de diferentes longitudes, (corto, mediano y grande) a su vez una vector de números no primos. Se nombró un vector: *vecf* que contiene los vectores de diferentes longitudes, se ejecutaron las secuencias de menor a mayor, mayor a menor y tres secuencias de ordenamiento aleatorio con 50 repeticiones, cada secuencia se realizó de 1 a 4 núcleos como se muestra en la figura 1, dando como resultado los distintos tiempo de ejecución de cada secuencia por núcleo.



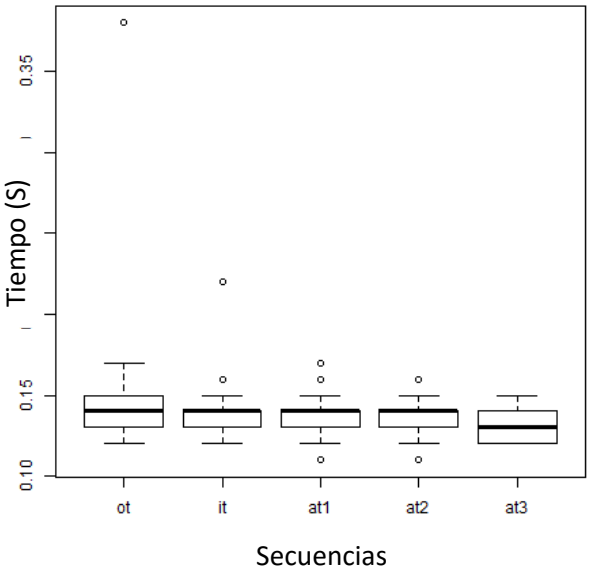
A) Un núcleo



B) Dos núcleos



C) Tres núcleos



D) Cuatro núcleos

Figura 1.- Tiempo de ejecución de cinco secuencias de 1 a 4 núcleos.

4.- Reto 1.

En este reto se nos especifica argumentar los resultados y sus diferencias, teniendo como ilustraciones la figura 1 podemos observar que si gradualmente que vayamos aumentando los núcleos las ejecuciones de tiempo son menores, se puede ver en A) donde todas las secuencias son parecidas se podría de intuir ya que como no tiene más núcleos con quien compartir las tareas el ordenamiento es más complicado y al hacerlo complicado genera mayor tiempo, pero hay casos que no presentan ese comportamiento como podemos observar en C) en las secuencias aleatorias los tiempos son tan parecidos como los de menor a mayor que son la que les toma mayor tiempo de procesado, se lo podemos atribuir a la forma de como ordena la computadora la serie de secuencias ya que lo que observamos es la sumatoria de todos los tiempos, no en todos en los casos son ideales y eso son da como resultado las cajas bigote que todos permanecen en un mismo rango, pero como mencionamos hay puntos que por así decirlo se podían considerarse como no ideales y son los que les toma más tiempo procesarlo por su ordenamiento de los vectores largos y que no sean primos, otro factor que podría influir es los sub-procesos que realiza la computadora o que gaste recursos en segundo plano.

5.- Conclusiones

Se logró simular el tiempo de ejecución de cada secuencia variando la cantidad de núcleos números primos y no primos. Uno de los puntos a notar es la relación de reducción de tiempo al ir aumentando la cantidad de núcleos como muestran en la figura 1, el tiempo de ejecución se reduce a la mitad de un núcleo a dos núcleos, se observa que las secuencias de números aleatorios presentan una mayor eficiencia dado al menor tiempo de ejecución, también otro factor que se presentó en todos las simulaciones que se realizaron es la secuencia de menor a mayor es la que le toma mayor tiempo en realizarla.

6.- Especificaciones de equipo

Modelo del sistema Inspiron 5420, fabricada por Dell Inc, procesador Intel(R) Core(TM) i5-3210MCPU @2.50Hz 2.50 Hz, memoria instalada (RAM) 8 GB (7.86 GB utilizable), tipo de sistema operativo de 64 bits procesador x64, edición de Windows 10 Pro.