

# Reporte práctica 3

## Teoría de colas

### Introducción

La teoría de colas estudia el comportamiento de las líneas de espera de un sistema. Una herramienta útil para poder medir una línea de espera sería, el tiempo de ejecución del mismo sistema. Para el propio interés, se estudiará el efecto del orden de ejecución de trabajos, así como el número de núcleos dedicados para realizarlos.

### Objetivos

Examinar la diferencia en los tiempos de ejecución en base a los diferentes posibles ordenamientos que pueden presentarse y el número de núcleos designados para la ejecución de cada clúster creado.

Analizar las posibles causas a las diferencias en los tiempos de ejecuciones, razonar como y porque detrás de dichas diferencias.

Aplicar en R pruebas estadísticas para determinar si las diferencias observadas entre los ordenamientos, proporción de primos y magnitud de números son significativas.

### Simulación y Resultados

Para la simulación, se requiere un vector que contenga diversos números, ya sean primos o no primos y de cualquier magnitud que se pueda imaginar, el cual será objeto de análisis y como resultado se debe obtener el tiempo de ejecución. En este caso se utilizará un vector *vf* que contenga doscientos números, el cual esta conformado por cuatro vectores secundarios de cincuenta elementos cada uno: uno que contiene números primos de magnitudes pequeñas, uno que contiene números primos de magnitudes grandes (millones), uno que contiene números primos que varían en magnitud desde un dígito hasta millones y por último uno que contiene los números del uno al cincuenta.

Otra de las características de la simulación será el ordenamiento de los elementos de nuestro vector *vf*, para este caso decidí tener siete diferentes ordenamientos:

- Elementos del vector ordenados de menos a mayor.
- Elementos del vector ordenados de forma inversa, es decir, de mayor a menor.
- Cinco ordenamientos aleatorios de los elementos del vector.

Como la simulación requiere ver los cambios en el uso de distinto número de núcleos, y el sistema de computo que se utilizó para procesar la información tiene cuatro núcleos, se obtuvieron resultados manteniendo cien repeticiones en cada caso por número de núcleos designados.

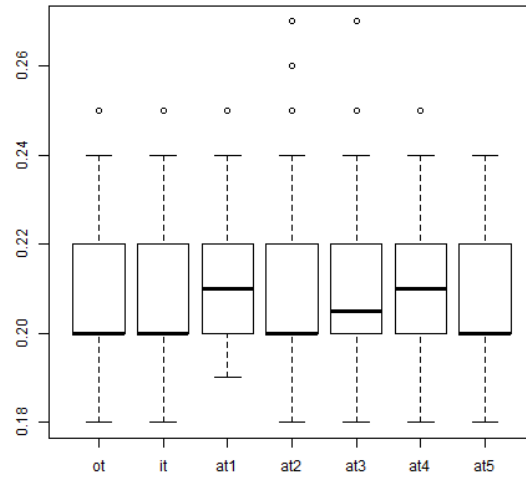


Figura 1. Tiempo de ejecución de los siete ordenamientos utilizando solo un núcleo.

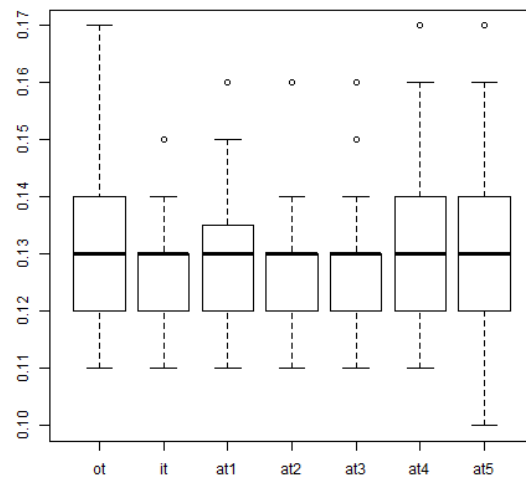


Figura 2. Tiempo de ejecución de los siete ordenamientos utilizando dos núcleos.

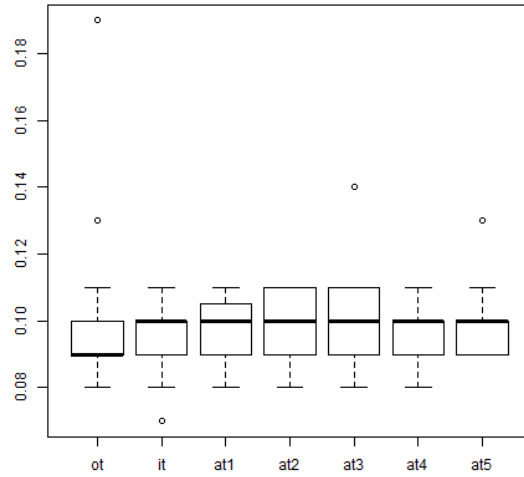


Figura 3. Tiempo de ejecución de los siete ordenamientos utilizando tres núcleos.

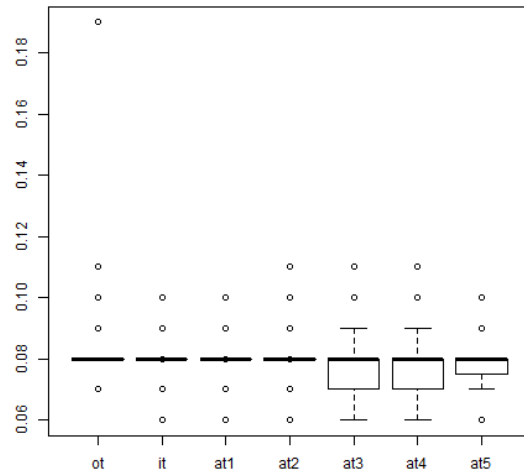


Figura 4. Tiempo de ejecución de los siete ordenamientos utilizando cuatro núcleos.

Como se puede observar a en las figuras 1 – 4, el tiempo de ejecución aumenta notablemente si reducimos el número de núcleos que se utilizan para realizar el trabajo. Además, es fácil de ver que el tiempo máximo de ejecución procede del ordenamiento de menor a mayor en la mayoría de los casos. Mientras que como se predijo en clase, uno de los ordenamientos aleatorios muestra rangos de tiempos de ejecución más cortos, por ende, es más efectivo dicho acomodo. Cabe mencionar que, al utilizar un núcleo, existe un ordenamiento aleatorio el cual tuvo un tiempo de ejecución mucho mas alto que el del ordenamiento de menor a mayor.

## Conclusiones

Para la práctica actual se logró simular en diferentes ocasiones la lectura de un vector de  $n$  cantidad de elementos, los cuales fueron identificados como *primos* o *no primos* y obtener como dato relevante el tiempo que tardó en ejecutarlo nuestro sistema de cómputo. Se observó la dependencia directa del tiempo en base al número de núcleos designados, siendo de cierta manera lógico el resultado. Donde entre más núcleos están disponibles, menor es el tiempo de ejecución.

Otro aspecto de suma importancia es el acomodo u ordenamiento de la información a analizar. Siendo notable que, si ordenamos los elementos de menor a mayor, de este obtendremos al menos un caso donde el tiempo de ejecución es máximo y por ende no muy eficiente.