

Ejercicios

Mikel Dalmau

15 de Mayo de 2016

Ejercicio 1: Se ejecuta 8 veces un determinado programa y se mide su tiempo de ejecución. Los resultados obtenidos aparecen en la siguiente tabla. Hacer una estimación del tiempo de ejecución del programa.

Tiempo (ms)
23,256
24,128
23,872
24,190
25,876
144,637
22,987
23,386

La media de los datos de la tabla es: 39.0415. Parece que no es acorde a los datos, esto se debe a que la media no es un estadístico robusto y se ve influenciada con facilidad por los valores extremos, no así la moda, que sí lo es y tiene un valor: 24 refleja mucho mejor el tiempo de ejecución del programa.

De todas formas no podemos desechar un valor a la ligera, para ello voy a utilizar un modelo en el que considero como extremos todos los datos que no se encuentren entre los cuartiles Q1 y Q3 de la distribución. Para no hacer un modelo tan restrictivo multiplico el rango intercuartil por 1.5.

```
Q1 <- quantile(x,probs=0.25)
Q3 <- quantile(x,probs=0.75)
step<- 2.5 * (Q3-Q1)
res <- sapply(x,FUN=function(x){
  if( x < Q3-step | x > Q1+step ){
    x <- NA
  }else{ x }})
res
## [1] 23.256 24.128 23.872 24.190 25.876 NA 22.987 23.386
```

En definitiva, el programa tendría un tiempo de ejecución de 24 ms, la media del resto de valores no excluidos.



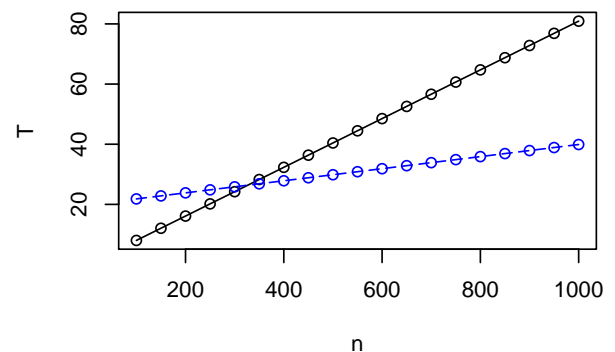
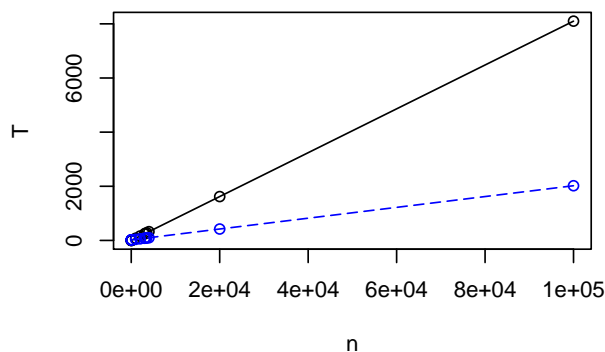
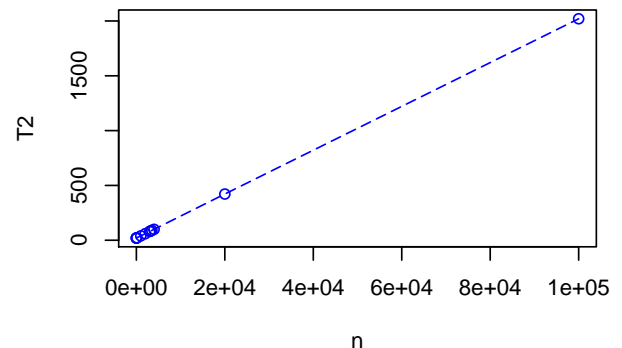
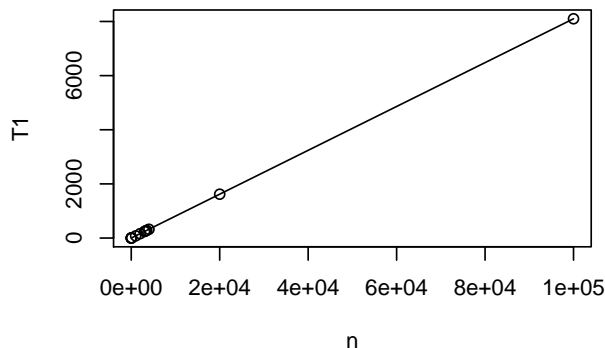
Ejercicio 2: En un determinado experimento se ha medido el tiempo de ejecución de dos alternativas de un programa en función del tamaño de los vectores procesados (y, en consecuencia, en función de la carga de cálculo), y se han obtenido los datos de la tabla. Representar los datos en un gráfico de la manera más adecuada posible y sacar conclusiones.

N(Bytes)	T1(ms)	T2(ms)
1	0,081	20,020
2	0,162	20,053
5	0,405	20,215
100	8,104	22,167
1000	81,062	39,893
2000	162,010	59,756
3000	242,013	80,176
3200	259,197	83,830
3500	283,518	89,875
4000	323,998	99,518
20000	1620,012	421,532
100000	8099,996	2020,225

Los datos de la siguiente tabla abarcan un gran número de valores y N incrementa diferentemente. Sabemos que ambos algoritmos son lineales mirando a los datos, el tiempo crece con $N = 100$ de 8ms para T1 y 22ms para T2 a 8000ms y 20000ms con $N=100*1000$. También sabemos que hay un punto de corte entre 100 y 1000.

Los siguientes gráficos muestran ambas rectas y su punto de corte, que se ha realizando creando puntos entre 100 y 1000 con las funciones ajustadas de T1 y T2.

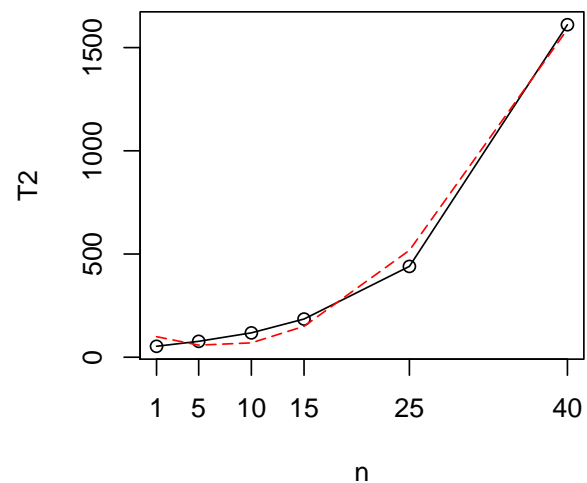
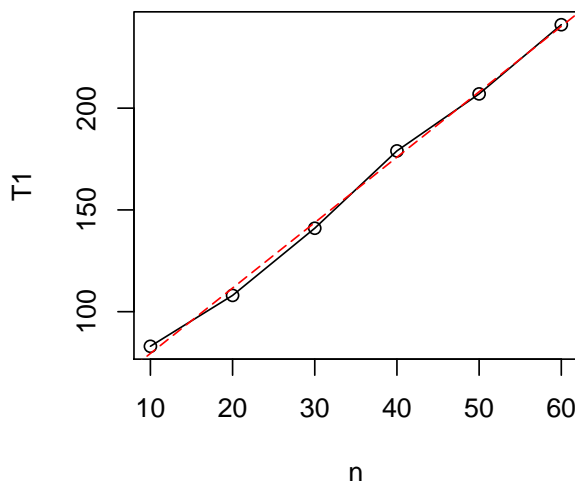
Podemos concluir que el programa 1 es más rápido con vectores pequeños, de 350 Bytes o menos aproximadamente, a partir de dicho tamaño el programa 2 es mejor ya que crece con una pendiente de 2% y el programa 1 del 8%.



Ejercicio 3: Las siguientes tablas muestran los tiempos de ejecución de dos programas diferentes en función del parámetro N. Representa los datos y obtén una expresión matemática que indique cómo varía T en función de N. A partir de ahí calcula el tiempo de ejecución esperado en cada caso para N=100.

N	T1	N	T2
10	83	1	53
20	108	5	77
30	141	10	118
40	179	15	185
50	207	25	440
60	241	40	1611

En las siguientes gráficas pueden verse las representaciones de los datos en negro y sus respectivos ajustes en rojo.



Se ha ajustado las tablas con las siguientes funciones:

Lineal : $y = 47.3333333 + 3.2142857 x$

Cuadrático : $y = 116.7037178 - 18.5565452 x + 1.3840086 x^2$

El tiempo de ejecución esperado a cada caso sería aproximado al los siguientes:

Lineal : 368.7619048 ms.

Cuadrático : 1.2101135×10^4 ms.