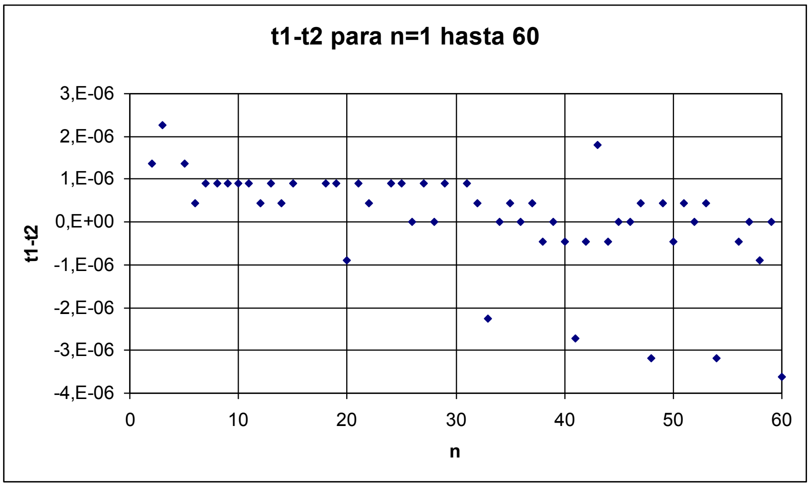
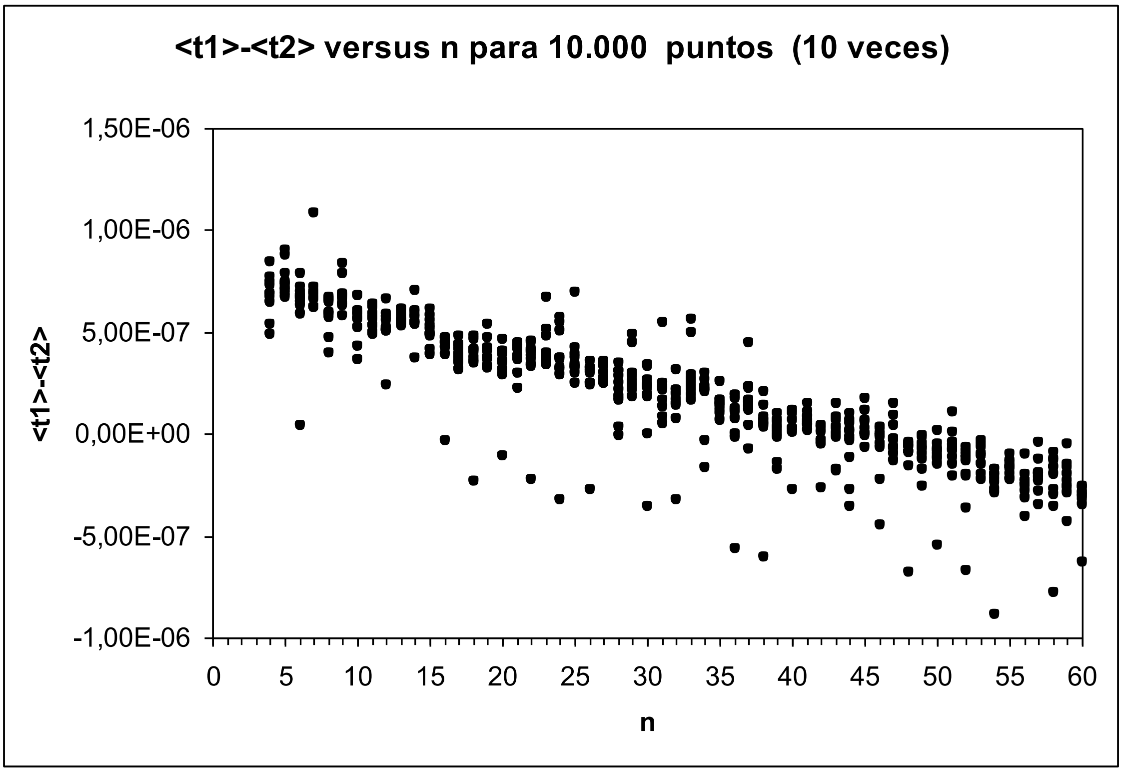
6. - ¿Para qué valores de n es más eficiente generar valores segun $f_{\tilde{x}}(x) = x^n$ usando la relación $x = \max(u_1, \ldots, u_n)$ o usando $x = u^{1/n}$?

Para buscar el valor de n, vamos a generar un algoritmo que nos calcule el tiempo que tarda en cada caso, y después vamos a realizar la diferencia entre ambos tiempos t1, para $x=u^{1/n}$ y t2 para $\max(u_1,\ldots,u_n)$, para ver el valor de n en que dicha diferencia cambia de signo

Como se puede ver en esta gráfica, la dispersión de los datos no permite dar un resultado preciso para n,



Por lo que vamos a repetir el algoritmo una cantidad de veces considerable (10*10000) para obtener un conjunto de 10 valores promediados para t1-t2 para cada n.

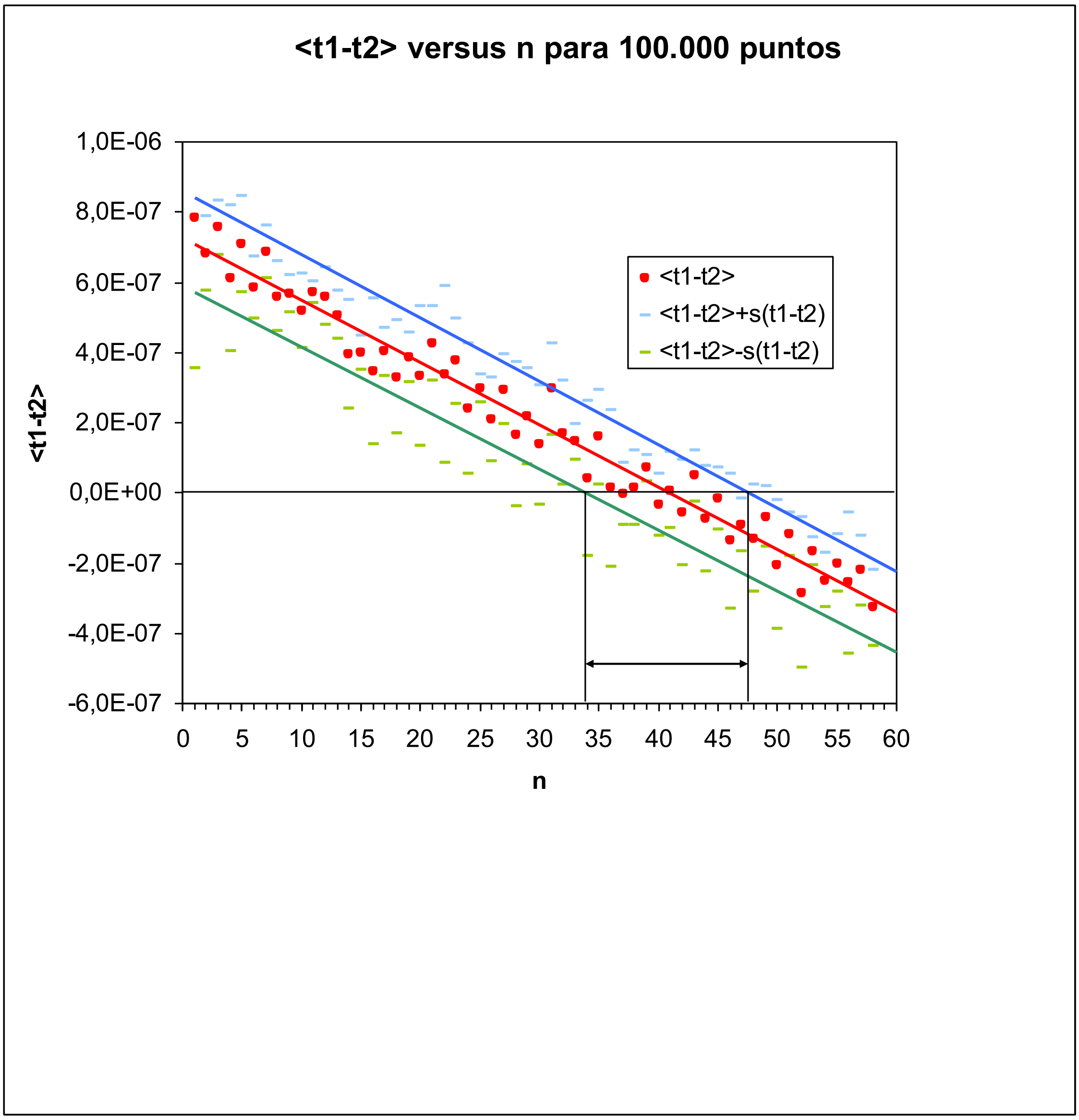


He eliminado deliberadamente los valores para n=1 y 2 , pues generan resultados muy dispares comparados con la tendencia lineal que tiene <t1-t2> respecto a n.
Ahora podemos calcular la media de estos 10 experimentos ya promediados sobre 10000 (lo denotaremos como <<t1-t2>>)y la desviación típica con lo que podremos construir unos límites superior e inferior necesario para el cálculo del valor de n mediante las regresiones lineales de los respectivos límites.

t1-t2=<< t1-t2>>+/- σ(<t1-t2>)

n	<< t1-t2>>	σ(<t1-t2>)	<< t1-t2>>+σ(<t1-t2>)	< <t1-t2>>-σ(<t1-t2>)
1	-1,19E-06	9,5899E-06	8,40E-06	-1,08E-05
2	3,43E-06	8,94694E-06	1,24E-05	-5,52E-06
3	7,85E-07	4,30265E-07	1,22E-06	3,55E-07
4	6,82E-07	1,06079E-07	7,88E-07	5,76E-07
5	7,56E-07	7,80051E-08	8,34E-07	6,78E-07
6	6,13E-07	2,07852E-07	8,20E-07	4,05E-07
7	7,08E-07	1,37869E-07	8,46E-07	5,71E-07
8	5,87E-07	8,82905E-08	6,75E-07	4,98E-07
9	6,87E-07	7,50227E-08	7,62E-07	6,12E-07
10	5,60E-07	9,87924E-08	6,58E-07	4,61E-07
11	5,68E-07	5,32726E-08	6,21E-07	5,14E-07
12	5,18E-07	1,07411E-07	6,25E-07	4,10E-07
13	5,71E-07	3,00832E-08	6,01E-07	5,41E-07
14	5,60E-07	8,1908E-08	6,42E-07	4,78E-07
15	5,06E-07	6,84611E-08	5,75E-07	4,38E-07
16	3,95E-07	1,53194E-07	5,48E-07	2,42E-07
17	3,99E-07	4,95571E-08	4,49E-07	3,50E-07
18	3,45E-07	2,07054E-07	5,52E-07	1,37E-07
19	4,02E-07	6,70852E-08	4,69E-07	3,35E-07
20	3,30E-07	1,61659E-07	4,91E-07	1,68E-07
21	3,86E-07	7,12768E-08	4,57E-07	3,14E-07
22	3,32E-07	1,97694E-07	5,30E-07	1,35E-07
23	4,25E-07	1,04415E-07	5,30E-07	3,21E-07
24	3,35E-07	2,51944E-07	5,87E-07	8,34E-08
25	3,75E-07	1,23414E-07	4,98E-07	2,51E-07
26	2,40E-07	1,84697E-07	4,24E-07	5,51E-08
27	2,97E-07	4,19105E-08	3,39E-07	2,55E-07
28	2,09E-07	1,18675E-07	3,28E-07	9,02E-08
29	2,94E-07	9,81443E-08	3,92E-07	1,96E-07
30	1,66E-07	2,06116E-07	3,72E-07	-4,04E-08
31	2,16E-07	1,3686E-07	3,53E-07	7,93E-08
32	1,36E-07	1,71318E-07	3,07E-07	-3,52E-08
33	2,97E-07	1,30588E-07	4,27E-07	1,66E-07
34	1,70E-07	1,48154E-07	3,19E-07	2,23E-08
35	1,46E-07	5,12565E-08	1,97E-07	9,48E-08
36	4,09E-08	2,21736E-07	2,63E-07	-1,81E-07
37	1,59E-07	1,34052E-07	2,93E-07	2,54E-08
38	1,36E-08	2,22964E-07	2,37E-07	-2,09E-07
39	-3,93E-09	8,76426E-08	8,37E-08	-9,16E-08
40	1,36E-08	1,07422E-07	1,21E-07	-9,38E-08
41	7,12E-08	3,75795E-08	1,09E-07	3,37E-08
42	-3,36E-08	8,70417E-08	5,35E-08	-1,21E-07

43	6,97E-09	1,08873E-07	1,16E-07	-1,02E-07
44	-5,44E-08	1,50053E-07	9,57E-08	-2,04E-07
45	4,80E-08	7,21135E-08	1,20E-07	-2,41E-08
46	-7,47E-08	1,48896E-07	7,42E-08	-2,24E-07
47	-1,54E-08	8,83087E-08	7,29E-08	-1,04E-07
48	-1,38E-07	1,91562E-07	5,35E-08	-3,30E-07
49	-9,31E-08	7,53656E-08	-1,77E-08	-1,68E-07
50	-1,30E-07	1,5406E-07	2,42E-08	-2,84E-07
51	-6,79E-08	8,57905E-08	1,79E-08	-1,54E-07
52	-2,05E-07	1,83613E-07	-2,11E-08	-3,88E-07
53	-1,17E-07	6,25089E-08	-5,46E-08	-1,80E-07
54	-2,84E-07	2,15261E-07	-6,90E-08	-5,00E-07
55	-1,67E-07	4,11097E-08	-1,26E-07	-2,08E-07
56	-2,50E-07	7,72929E-08	-1,73E-07	-3,27E-07
57	-2,01E-07	8,12903E-08	-1,20E-07	-2,82E-07
58	-2,55E-07	2,0114E-07	-5,44E-08	-4,57E-07
59	-2,21E-07	9,89132E-08	-1,23E-07	-3,20E-07
60	-3,27E-07	1,09143E-07	-2,18E-07	-4,37E-07



	Recta regresión	**R^2**	**Corte eje x -> n**
<< t1-t2>>+σ(t1-t2)	y = -1,8055E-08x + 8,6004E-07	(0,94369)	**48**
< <t1-t2>>	y = -1,7710E-08x + 7,2429E-07	0,96763	**41**
<< t1-t2>>- σ(t1-t2)	y = -1,7364E-08x + 5,8854E-07	(0,89607)	**34**

Por tanto el resultado es **n = 41+/-7,**

Para valores inferiores a n=34 podemos decir que es más eficiente utilizar max($u_1,\ldots,u_n$) para generar números aleatorios que sigan la función de distribución de probabilidad $f(x)=x^n$,y para valores superiores a 48 és más eficiente el uso directo de $x=u^{1/n}$.

Algoritmo en matlab:

```
for k=1:10; %repetimos el experimento 10*10000 veces
      for j=1:10000;
      m=60;  %valor máximo para n
            for n=1:m; %para cada valor de n calculamos los tiempos de
      ejecución de ambos métodos

            t0=tic;
            rand()^(inv(n));
            t1=toc(t0);
            F(n,j)=t1;

            t0=tic;
            max(rand(n,1));
            t2=toc(t0);
            G(n,j)=t2;

            end;
      end;

A=F-G;  %diferencia de tiempos, en principio, el valor de n por el
cual un método
%pasa a ser más eficiente que el otro

      for n=1:m;  %calculamos la media entre los   10000 experimentos
      (repetido 10 veces)
      media(n,k)=mean(A(n,:));
      end;
end;
```