Counter Probabilities

António Ramos, ajframos@ua.pt, 101193, MEI

***Resumo* – Este trabalho apresenta uma análise comparativa dos contadores exatos, probabilidade fixa de 1/8 e csuros.**

Para o estudo, foi usado livros de grandes dimensões de língua portuguesa, inglesa, francesa e espanhola.

***Abstract* - This work presents a computational comparation of exact counter, fixed probability and csuros.**

For the study, has used large books of Portuguese language, English, French and Spanish.

I. Introduction

When exists the need to count an item of a largest dataset, normally, it's the memory used in computer is largest, so we need to get something that could do the same using less memory. In this work will be compare the number of counting of characters in a file using fixed probability and csuros against exact counter, doing counter 1000 and 10000 times.

II. Algorithm Analysis

In this chapter is it's showed how the algorithm was analysed and what is the approach chosen.

A. Handle information

Due to free choice on books, I choose "*The Bible*" to represent English, don quixote to represent Spanish, "*Histoire de la Nouvelle-France*" to represent French, "*Os Lusíadas*" to represent Portuguese.

Normally in the books exists characters that are not letters, so it's used a verification to ignore them using *isalpha()* in python. Next we replace accents in letters, for example à is count a, etc, to give more counting's, to know what character is, it's used the *ord()* function in python, that converts a string in ascii number, then it's checked the number intended. The book is read it one time, calculation the exact counter and then it's added the line information in a list to do fixed and csuros probabilities.

B. Exact Counter

For exact counting, it's used a simple dictionary. The dictionary saves the letter from A-Z and the number of occurrences.

C. Fixed Probability with 1/8

For fixed probability, it's generated a number between 0 and 1 if the number is larger than 1/8 (0.125), add 1 count to a dictionary. This is executed n times (1000 and 10000).

D. Csuros

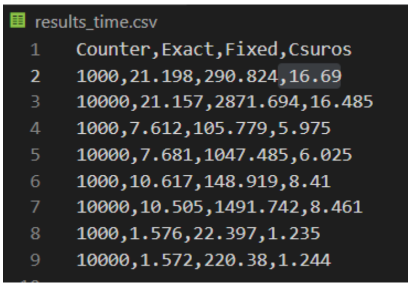
The csuros was implemented according to the slides and the author Miklos Csuros [1]. Its haves two parameters x and d. The parameter M, that is nonnegative integer, is calculated according to the expression $M = 2^d$.

This counter relies on random int number between 0 and 1, if the number generated is 1 then return counter. This is executed n times (1000 and 10000).

III. Tests Sequence Analysis

A. *Execution Time Analysis*

For analyse the time of execution of each counter, and counting, the information it's save in a csv file (figure 1)



```
Counter,Exact,Fixed,Csuros
1000,21.198,290.824,16.69
10000,21.157,2871.694,16.485
1000,7.612,105.779,5.975
10000,7.681,1047.485,6.025
1000,10.617,148.919,8.41
10000,10.505,1491.742,8.461
1000,1.576,22.397,1.235
10000,1.572,220.38,1.244
```

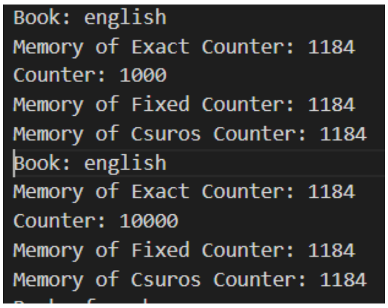
Figure 1 - Time analysis

It's worth pointing out that exact counter is executed one time while other's counters 1000 and 10000 times.

In the figure 1, it shows that fixed probabilities are take more time then other, this because of verification of generation number is bigger than 1/8 takes time. The csuros with parameters 15 and 25 takes less than exact counter even in larger counting. If the parameters change the time of execution changes either.

B. *Memory Analysis*

For analysing the memory that is used in each counter, it's used the function of the sys, getsizeof() in a dictionary of each counter. The information is written in the file "results_memory.txt" (figure 2).



```
Book: english
Memory of Exact Counter: 1184
Counter: 1000
Memory of Fixed Counter: 1184
Memory of Csuros Counter: 1184
Book: english
Memory of Exact Counter: 1184
Counter: 10000
Memory of Fixed Counter: 1184
Memory of Csuros Counter: 1184
```

Figure 2 - English memory

In figure 2 it's shown that the memory doesn't variate from each counter. But these values can't show very good the performance of each counter, so we need to add the time analysis.

C. *Counting Analysis*

For analysing the counting of each character, was made two reports, one with image of counting's of each character of each counter of each book with the following layout "X_count of y generated in 1000/10000 times.png", the second its haves two files, one that saves full dictionary of each counter, the highest, lowest, mean, max deviation, mad and standard deviation of each counter.

In the exact counts description exists two with 1000 and 10000, but it's the same. This error is because of logic for two others counter, so it's written the three like that. (Figure 3). In figure 4, it's shown the results of the file results_counter.txt, it's another away to show the same result.

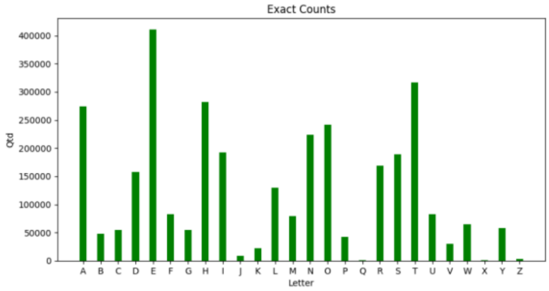
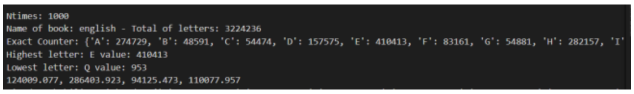


Figure 3 - Exact Counts English (bible, example)



```
Ntimes: 1000
Name of book: english - Total of letters: 3224236
Exact Counter: {'A': 274729, 'B': 48591, 'C': 54474, 'D': 157575, 'E': 410413, 'F': 83161, 'G': 54881, 'H': 282157, 'I'
Highest letter: E value: 410413
Lowest letter: Q value: 953
124009.077, 286403.923, 94125.473, 110077.957
```

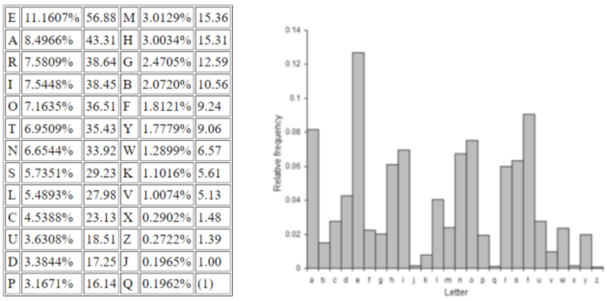
Figure 4 - Results Counter

D. *Language Analysis*

For analyse each language, English, Spanish, French and Portuguese, it was search information in web to compare with the results of algorithms.

According to [2], the most common letter in English is E, and the least Z. The results of the algorithm in exact counts, fixed probability and csuros are shown in figure 6 and 7.

The results show that the most common letter in the book bible is E with 51301059, and the least is Q with 119188, the result of each letter is alike of figure 5.



E	11.1607%	56.88	M	3.0129%	15.36
A	8.4966%	43.31	H	3.0034%	15.31
R	7.5809%	38.64	G	2.4705%	12.59
I	7.5448%	38.45	B	2.0720%	10.56
O	7.1635%	36.51	F	1.8121%	9.24
T	6.9509%	35.43	Y	1.7779%	9.06
N	6.6544%	33.92	W	1.2899%	6.57
S	5.7351%	29.23	K	1.1016%	5.61
L	5.4893%	27.98	V	1.0074%	5.13
C	4.5388%	23.13	X	0.2902%	1.48
U	3.6308%	18.51	Z	0.2722%	1.39
D	3.3844%	17.25	J	0.1965%	1.00
P	3.1671%	16.14	Q	0.1962%	(1)

Figure 5 - English Site

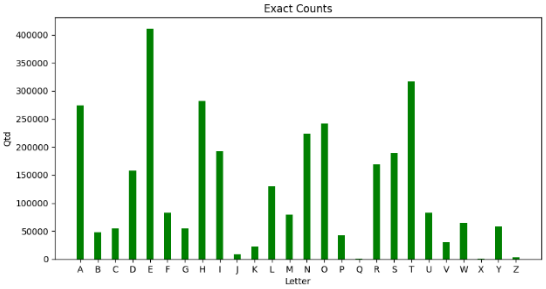
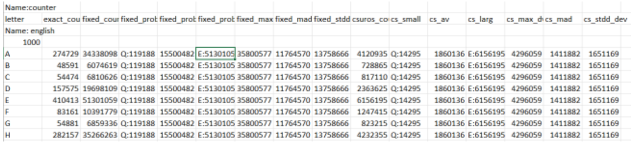


Figure 6 - Result Exact Count English



Name:counter															
letter	exact_cou	fixed_cour	fixed_prot	fixed_prot	fixed_prot	fixed_max	fixed_mad	fixed_stdd	csuros_co	cs_small	cs_av	cs_larg	cs_max_d	cs_mad	cs_stdd_dev
Name: english															
1000															
A	274729	34338098	Q:119188	15500482	E:5130105	35800577	11764570	13758666	4120935	Q:14295	1860136	E:6156195	4296059	1411882	1651169
B	48591	6074619	Q:119188	15500482	E:5130105	35800577	11764570	13758666	728865	Q:14295	1860136	E:6156195	4296059	1411882	1651169
C	54474	6810626	Q:119188	15500482	E:5130105	35800577	11764570	13758666	817110	Q:14295	1860136	E:6156195	4296059	1411882	1651169
D	157575	19698109	Q:119188	15500482	E:5130105	35800577	11764570	13758666	2363625	Q:14295	1860136	E:6156195	4296059	1411882	1651169
E	410413	51301059	Q:119188	15500482	E:5130105	35800577	11764570	13758666	6156195	Q:14295	1860136	E:6156195	4296059	1411882	1651169
F	83161	10391779	Q:119188	15500482	E:5130105	35800577	11764570	13758666	1247415	Q:14295	1860136	E:6156195	4296059	1411882	1651169
G	54881	6859336	Q:119188	15500482	E:5130105	35800577	11764570	13758666	823215	Q:14295	1860136	E:6156195	4296059	1411882	1651169
H	282157	35266263	Q:119188	15500482	E:5130105	35800577	11764570	13758666	4232355	Q:14295	1860136	E:6156195	4296059	1411882	1651169

Figure 7 - Result English csv

As claimed by [3] the most common letter in French is e. The result of the algorithm gets the same result for the most common and the least is W with 7 counts (figure 8 and 9).

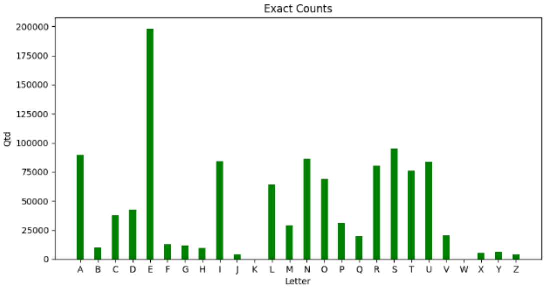
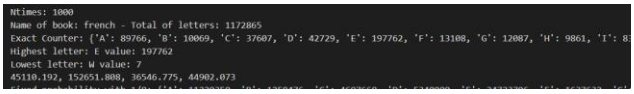


Figure 8 - Exact count of French



```
Ntimes: 1000
Name of book: french - Total of letters: 1172865
Exact Counter: {'A': 89766, 'B': 10069, 'C': 37607, 'D': 42729, 'E': 197762, 'F': 13108, 'G': 12087, 'H': 9861, 'I': 8
Highest letter: E value: 197762
Lowest letter: W value: 7
45110.192, 152651.808, 36546.775, 44902.073
```

Figure 9 - Results French

As stated by [4] the most frequent letter in Spanish is E. The results of the algorithm (figure 9 and 10) shown that the most letter is E with 28822764 occurrences, and the least is W with 273.

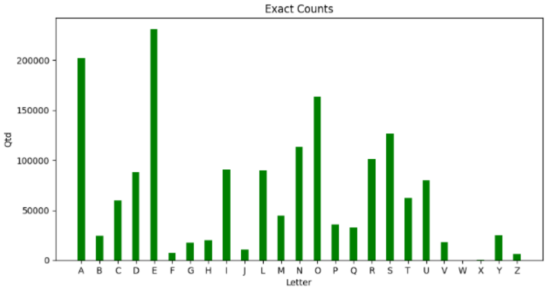
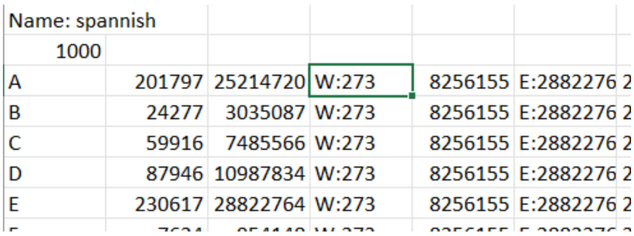


Figure 10 - Results Spanish



Name: spannish						
1000						
A	201797	25214720	W:273	8256155	E:2882276	2
B	24277	3035087	W:273	8256155	E:2882276	2
C	59916	7485566	W:273	8256155	E:2882276	2
D	87946	10987834	W:273	8256155	E:2882276	2
E	230617	28822764	W:273	8256155	E:2882276	2

Figure 11 - Occurences Spanish

According to [5] the most common letter in Portuguese is E, A. The results of the algorithm (figure 12 and 13), shown that the most common letter is A with 4286229, and the least Y with 1010 occurrences.

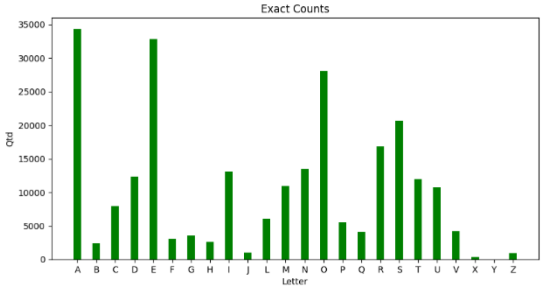
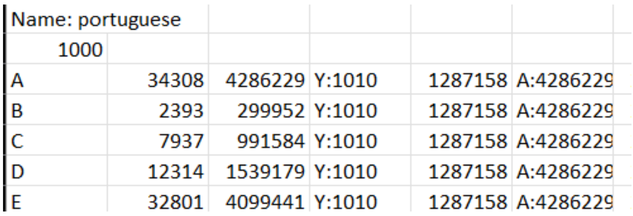


Figure 12 - Results Portuguese



Name: portuguese					
1000					
A	34308	4286229	Y:1010	1287158	A:4286229
B	2393	299952	Y:1010	1287158	A:4286229
C	7937	991584	Y:1010	1287158	A:4286229
D	12314	1539179	Y:1010	1287158	A:4286229
E	32801	4099441	Y:1010	1287158	A:4286229

Figure 13 - Occurrences Portuguese

IV. Conclusion

Using csuros counter, the result of the counting of the letters of the file is faster and the results equal to exact counter, so to analyse the most common letter in fast way, it’s needed to choose this, but if the need of the exact count, we need to choose the exact counter.

The size of the book got impact on execution of the algorithms.

References

[1] M. Csurös, “Approximate counting with a floating-point counter,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6196 LNCS, pp. 358–367, 2010, doi: 10.1007/978-3-642-14031-0_39.

[2] https://www3.nd.edu/~busiforc/handouts/cryptography/letterfrequencies.html

[3] https://www.sttmedia.com/characterfrequency-french

[4] https://www.sttmedia.com/characterfrequency-spanish

[5] https://www.sttmedia.com/characterfrequency-portuguese