(The text file used: 61 first pages of the book ‘Aion’)

total number of words(tokens) in the text: 37885

total number of distinct words(vocab) in the text: 9089

Zipf’s law:

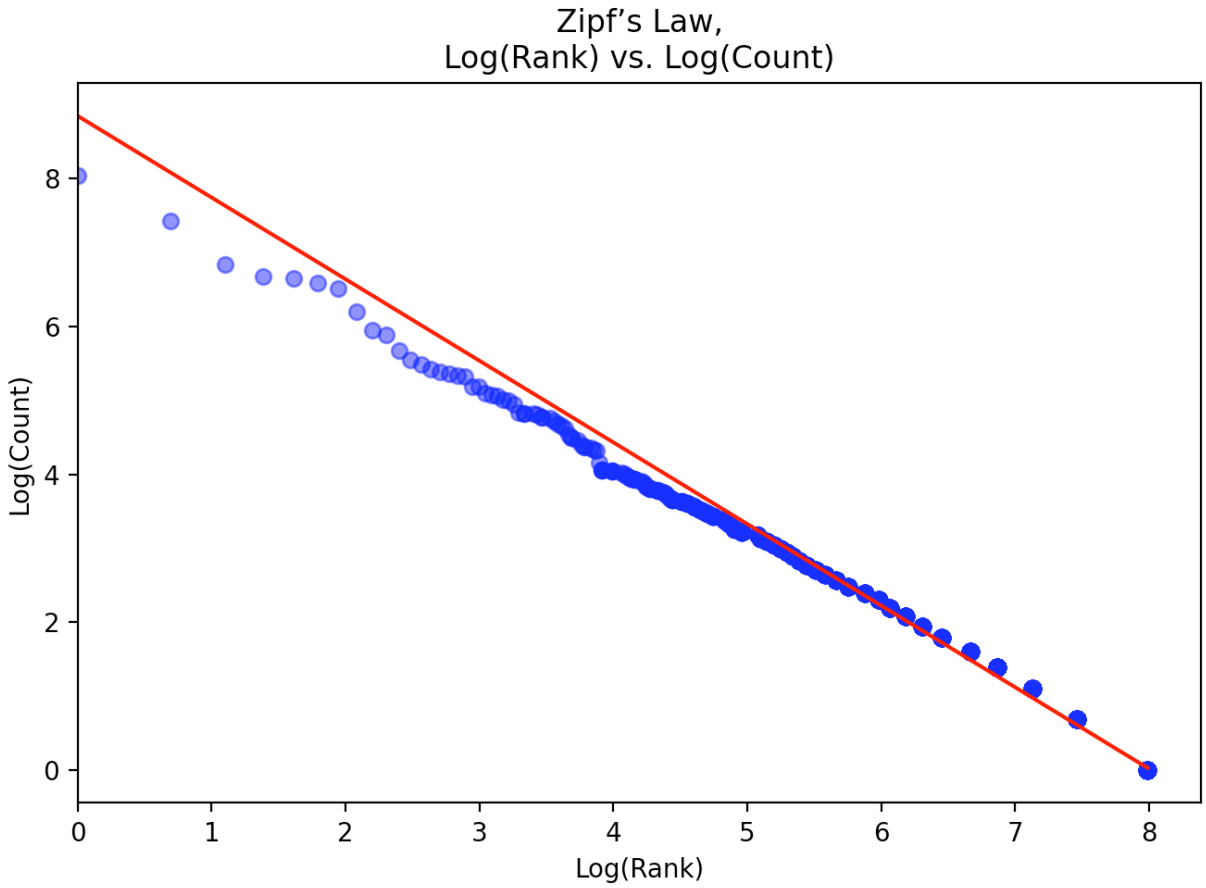
(First 25 vocab words)
word: | frequency:

the	3099
of	1690
and	937
is	793
in	774
to	726
a	677
that	494
as	383
it	362
this	293
be	257
by	242
not	227
with	219
for	213
which	208
are	205
from	180
but	178
on	164
he	160
an	157
his	151
one	148

The line equation is: $y = -1.1x + 8.85$
(calculated in the code)

Log(f) = Log(c) - s Log(r)
Where f is frequency, r is rank, and s and c are parameters that depend on language.

This is obviously a linear form, and should appear that way when plotted. For this we need to count the frequencies of each word.



Zipf's law is most easily observed by plotting the data on a log-log graph, with the axes being log (rank order) and log (frequency). For example, the word *"the"* (as described above) would appear at $x = \log(1)$, $y = \log(69971)$. It is also possible to plot reciprocal rank against frequency or reciprocal frequency or interword interval against rank.[2] The data conform to Zipf's law to the extent that the plot is linear.

Formally, let:

- N be the number of elements;
- k be their rank;
- s be the value of the exponent characterizing the distribution.

Zipf's law then predicts that out of a population of N elements, the normalized frequency of the element of rank k, $f(k;s,N)$, is:

$$f(k;s,N) = \frac{1/k^s}{\sum_{n=1}^{N}(1/n^s)}$$

⇨ In the code an estimation for s was found

Estimation of parameter s is: -1.104
By putting this value in Zipf's law we get a number close to the actual value so it is working well.
From the graph plot you can observe that it is in accordance with Zipf's law.

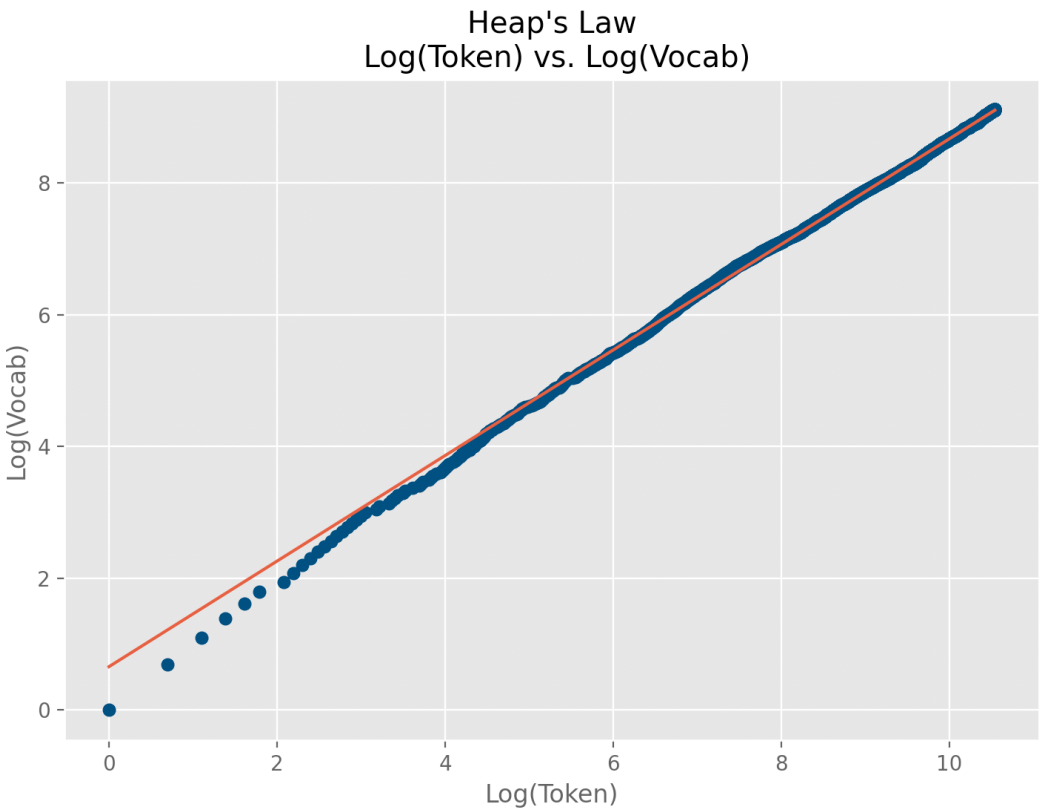
Heap's law:

First 25 vocab words
VocabSize: | CollectionSize:

1	1
2	2
3	3
4	4
5	5
6	6
7	8
8	9
9	10
10	11
11	12
12	13
13	14

14	15
15	16
16	17
17	18
18	19
19	20
20	21
21	24
22	25
23	28
24	29
25	30
26	31
27	33
28	34
29	37
30	40
31	41
32	42
33	45
34	46
35	47
36	49
37	52
38	53
39	54
40	55
41	56
42	57
43	60
44	62
45	63
46	64
47	65
48	66
49	67
50	69

The line equation is: $y = 0.8x + 0.66$
(calculated in the code)



Heaps' law (also called **Herdan's law**) is an empirical law which describes the number of distinct words in a document (or set of documents) as a function of the document length (so called type-token relation). It can be formulated as where V_R is the number of distinct words in an instance text of size n. K and β are free parameters determined empirically.

Now, we can be able to estimate the *vocabulary size(unique words)* of the same document if we are provided with the *n(total number of the words)*. To predict that we first have to calculate the constant value K and β, for that we make use of the calculated value of **V(n)** and **n** and put that in V(n) = K n^β

V(n) = K n^β => log V(n) = β log(n) + K

By plugging in n and V(n) based off the graph values (fitted line)
By solving systems of equations, we get:
0.879 = B , 1.134 = K

now if we plug in n by these parameters we get a predicted result which is very close to the actual value.
The parameter k is quite variable and will reduce the growth rate of the vocabulary if stemming or lemmatization is used whereas including numbers and spelling errors can increase it.
Heaps' law suggests that (i) the vocabulary size continues to expand with more documents in the collection, but the growth rate slows down and (ii) the size of the vocabulary is quite large for large collections.