1. Power Laws

Aleix Torres i Camps, Àlex Batlle Casellas

17/09/2021

Family names.

Exercici 1. Use python + matplotlib to plot the frequence of `apellidos.csv` in decreasing order.

This is the plot of the rank versus the absolute frequency of family names in Spain.

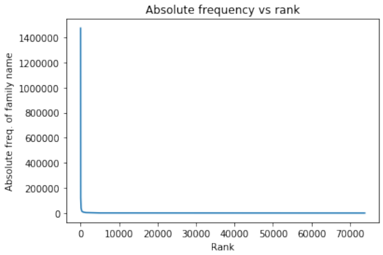

Figure 1: Plot of absolute frequency of family names against rank.

Exercici 2. (joke) RESIST THE TEMPTATION to say that this is an exponential distribution "because it decrases very fast". Is it a powerlaw? Or, can it be approximated by a powerlaw?

We can't extract much information of the distribution, but nontheless we can observe that the frequency falls rapidly. For what we know untill now, we can't say for sure if it is a powerlaw or not, or if it can be approximated by one. Nevertheless, in the next exercices we will try it.

Exercici 3. Tell matplotlib to use logarithmic x and y axes.

Exercici 4. Let's find a and c more analytically. Assume we have $\log y = a \cdot \log x + \log c$. Take two distinct large values of x, find their corresponding values of y, set up a system of two linear equations, and solve for a and c.

We have chosen $x_1 = 100$ and $x_2 = 10000$, and this ranks correspond to the absolute frequencies $y_1 = 43663$ and $y_2 = 356$. Hence,

$$\hat{a} = \frac{\log y_2 - \log y_1}{\log x_2 - \log x_1} \approx -1.044, \quad \log \hat{c} = \log y_2 - \hat{a} \cdot \log x_2 \approx 15.493 \implies \hat{c} = \exp \log \hat{c} \approx 5355217.890.$$

We tried some values of b with the calculated $\hat{a}$ and $\hat{c}$. For $\hat{b} = 4$, the curve $y = \hat{c}(x + \hat{b})^{\hat{a}}$ adjusts pretty well the first part of the loglog-plot (which is almost horizontal), and the linear part as well. These are the resulting plot:

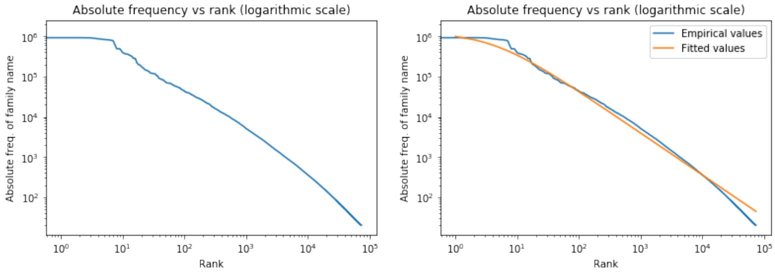

Figure 2: Plot of absolute frequency of family names against rank (logarithmic scale on both axes).

Exercici 5. All of the above can also be done pretty easily with a spreadsheet. Think how.

First of all, everytime we want a new plot, we just have to select the column that will be the x-axis, then the y-axis and ask the program to do the plot that we like. In the exercices above we focused in two columns ("Orden" and "Apellido 1"), from these we can generate our first plot. After that, we should compute the logarithm of these two columns (something like "=log(position of a cell)" and filling the cells below will do the job), and again, we can generate a plot from these two new columns. Finally, we want to find a and c. In some spreadsheets, for example MS Excel, we can fit power laws directly. Alternatively, we can do the same that we have done in exercise 4.

Rivers.

The file `rivers.csv` contains info about the longest rivers on Earth.

Exercici 6. Check whether the distribution of lengths follows something like a powerlaw. For curiosity, see the humongous outlier in basin area.

The distribution of lengths shows some signs of following a power law that would be clearer if there was more data. The log-log plot tends to a line for big values of the rank, so we could approximate this distribution by a power law.

There is a big outlier in terms of basin area, and it is the Amazon river, one of the longest rivers on earth, that crosses multiple countries.

Words in text.

Exercici 7. Write Python code that creates a .csv file with information about the rank of a word, the word and its absolute frequency, when accounting for all the words inside all files within a given directory.

Exercici 8. Try your code on the file collection in `novels.zip`. Proceeding as before, do the word frequencies look like a power law?

Now, when plotting the results (absolute frequency against rank) we see no conclusive information. The plot could be one of a power law, but we cannot say so for sure. Hence, we do a logarithmic scaling of the axes, and now the results do look like some sort of power law, as the plot shows some pseudo-linear behaviour.

We can again try to guess the parameters by picking two large values of x and calculating the gradient from the empirical values of y. If we do this for $x_1 = 100$ and $x_2 = 10000$, the resulting a is around -1.114, the resulting c is approximately 542659 and if we try to fit b by observation, we conclude that $b \approx 2.75$ fits the empirical curve somewhat well.

Exercici 9. Now, change your program so that for some k, it prints the number of different words in the collection after it has read k, $2k$, $3k$, $4k$ words from the collection. Collect the output of pairs `(ik, distinct words)`. Create a plot and check whether the distribution looks like a powerlaw.

When asking for a log-log plot of the results with this modification for $k = 100$, we can see that the resulting graph of the number of different words (d) against the number of processed words (N) does look something like a line. We can even try to approximate the values for β and α in Heaps' law, $d = \alpha \cdot N^{\beta}$, taking for example $N_1 = 10^3$ and $N_2 = 10^6$, and d_1 and d_2 their corresponding empirical values. Then,

$$\hat{\beta} = \frac{\log d_2 - \log d_1}{\log N_2 - \log N_1} \approx 0.618, \quad \hat{\alpha} = \exp\left(\log d_2 - \hat{\beta} \log N_2\right) \approx 4.222.$$

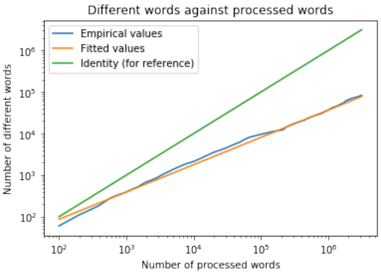

Figure 3: Plot of number of unique words against number of processed words.