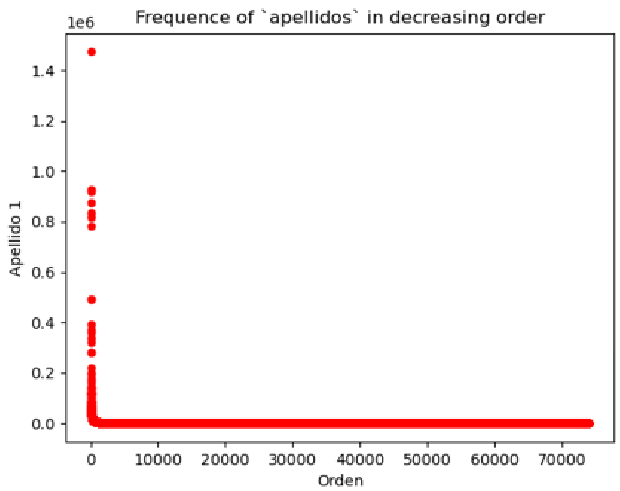
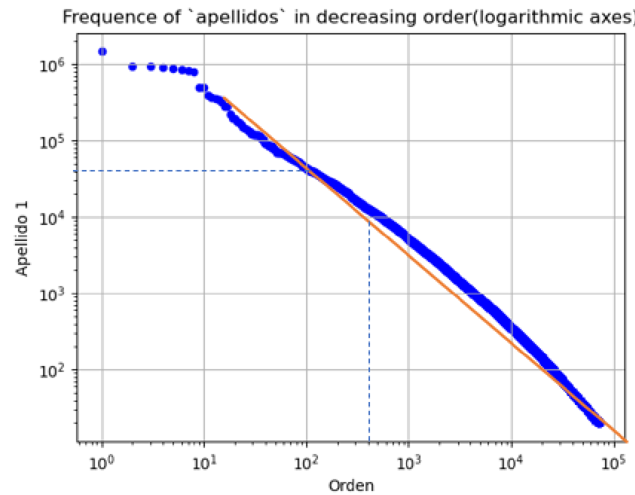
Subject: ***Information Retrieval and Analysis***
Student: ***Stanciu Iulia-Cristina***
Group: *12*
Practicals: ***Lab1 – Powerlaws – 26.09.2022***

Practical 1 - Powerlaws

2. Distribution of family names





Answer: A powerlaw is defined as $y = c * (x + b)^a$, with a, b and c constants. Looking at the plot of frequence of family names in the Spanish census of 2015, we can see that the logarithmic representation is almost a linear function. This means that it can be considered a powerlaw with $b \cong 0$.

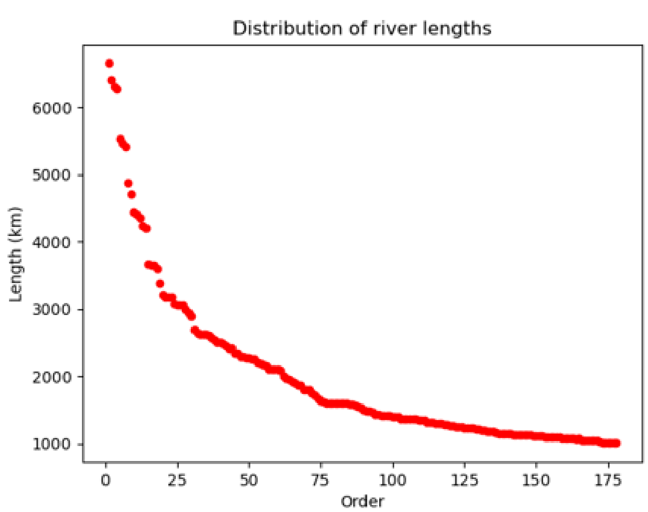
$\log c \cong 1.4 * 10^5 => c = 4.05 * e^{10000}$

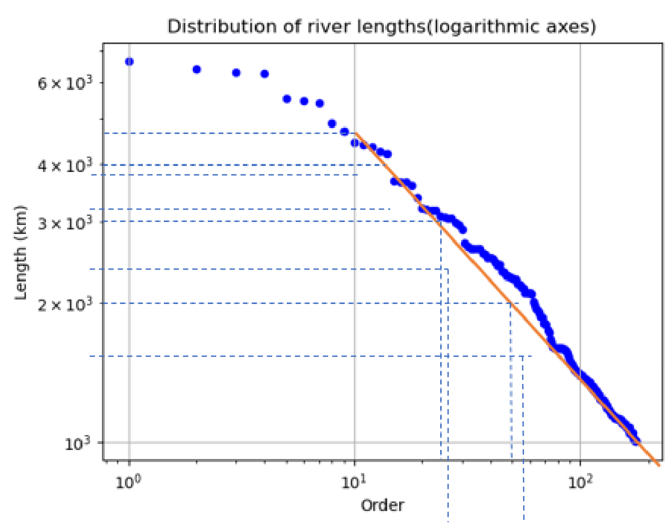
Considering b = 0, $\log y = a * \log x + \log c$.

⇨ We can verify by choosing two points (x1, y1) = $(3 * 10^2, 10^4)$ and (x2, y2) = $(10^2, 3 * 10^4)$

⇨ $=> a = \frac{y1-y2}{x1-x2} = \frac{-2*10^4}{10^2} = -200$

3. Distribution of river length





Subject: ***Information Retrieval and Analysis***
Student: ***Stanciu Iulia-Cristina***
Group: *12*
Practicals: ***Lab1 – Powerlaws – 26.09.2022***

A powerlaw is defined as $y = c * (x + b)^a$, with a, b and c constants. The logarithmic plot of the distribution of river lengths resembles a linear function, except the points for a low value of x. This means that it can be considered a powerlaw.

$\log c \cong 2 * 10^2 => c = 7.39 * e^{200}$

Considering b = 0, $\log y = a * \log x + \log c$.

⇨ We can verify by choosing two points (x1, y1) = $(1.3 * 10^1, 3*10^3)$ and (x2, y2) = $(5 * 10^1, 2 * 10^3)$

⇨ $=> a = \frac{y1-y2}{x1-x2} = \frac{-10^3}{37} = -27.02$

4. Text Laws

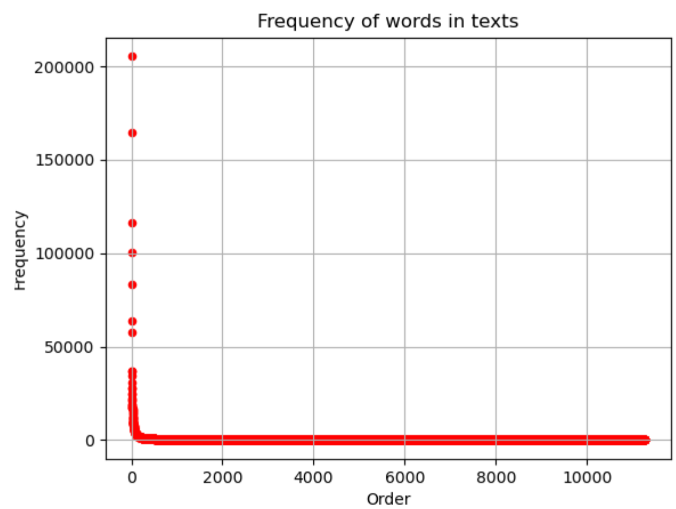
Python code:

```
def readWords(path, output_filename_full, output_filename_k):
    dictionary = {}
    unique_words = []
    count = 0;

    for f in listdir(path):
        ff = join(path,f)
        print("processing ",ff)
        for text in open(ff, "r", encoding="utf8"):
            # transform punctuation to spaces in line
            # text = text.read()
            skips = [".", ",", ":", ";", "-", "_", "'", '"', "\n", "\r", "?", "!", "(", ")", "*", "/", "[", "]",
                     "https", "1", "2", "3", "4", "5", "6", "7", "8", "9", "0"]
            for ch in skips:
                text = text.replace(ch, "")

            # translate line to lowercase
            text = text.lower()
            for word in text.split(" "):
                count +=1
                if word in dictionary:
                    dictionary[word] += 1
                else:
                    dictionary[word] = 1


    i = 0;
    with open(output_filename_full, 'w') as of:
        of.write("Order" + ";" + "Word" + ";" + "Frequency" + "\n")
        for word in sorted(dictionary, key=dictionary.get, reverse=True):
            print(word, dictionary[word])
            i += 1
            of.write(str(i) + ";" + str(word) + ";" + str(dictionary[word]) + "\n")
```



The distribution of words in the given novels is also a powerlaw.

$\log c \cong 2 * 10^4 => c = 7.39 * e^{10000}$

Considering b = 0, $\log y = a * \log x + \log c$.

⇨ We can verify by choosing two points (x1, y1) = $(3 * 10^2, 10^3)$ and (x2, y2) = $(3 * 10^1, 10^4)$

⇨ $=> a = \frac{y1-y2}{x1-x2} = \frac{-9*10^3}{270} = -33.33$

The distribution of unique words per number of words is also a powerlaw, but almost linear. a = 1; b=0; c= 0.035