CAI Lab1

Jordi Puig Rabat & Jose Pérez Cano

September 2020

1 Distribution of family names

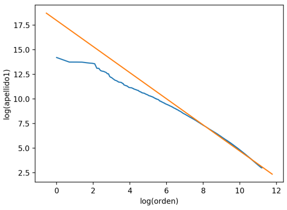
Exercise 1

We can observe that it is not any perfect mathematical function, because at the beginning it has some sharp changes. But in overall it looks like a power law.

Exercise 2

In order to see if it is a power law we apply a double log transformation and watch that it is almost a line, except for a little curvature at the beginning, which seems reasonable according to previous observations.

Exercise 3



Exercise 4

We fitted a linear model with the library ‘scipy.stats‘ of python to the data resulting in an r^2 value of 0.9914, which supports the idea that the data follows a straight line. The values of the slope and the intercept are respectively: -1.33, 17.97.

Exercise 5

In any spreadsheet (and in this case Excel) there is an option to create a tendency line on any graphic. So in order to replicate the above result it is needed to first apply a log transformation to both columns, then create a graphic with the data and finally add a tendency line to it.

2 Distribution of river lengths

Exercise 6

Again, repeating the process of plotting in loglog scale we don’t get a straight line at the beginning but if we remove the longest and shortest rivers we get a proper straight line. The reasoning behind doing so it that normally the tails of the distribution have anomalies. One of those in this case is the Amazon river has roughly the double of the drainage area than the longest river.

3 Words in text

Exercise 7

First of all it is needed to convert the csv to a more suitable format. We used the library pandas to read the csv and create a dataframe to work with. To convert the punctuation marks to white spaces we use

```
import string
translator = str.maketrans(string.punctuation, \
                           ' '*len(string.punctuation))
```

and then inside the loop

```
line = line.translate(translator)
line = line.lower()
```

The part of the dictionary is simple too:

```
for word in line.split():
    if word in dicc.keys():
        dicc[word] += 1
    else:
        dicc[word] = 1
```

Last of all, the result is written, again using pandas, to a csv.

```
l = []
for item in dicc.keys():
    l.append([item, dicc[item]])
l.sort(key=lambda x:-x[1])
df = pd.DataFrame(l, columns = ["word", "count"])
df["index"] = range(1, len(l)+1)
df.to_csv(output_filename + '.csv')
```

Exercise 8

The log-log plot looks similar to the one in exercise 3, even more akin to a line.

Exercise 9

The plot looks like a line but this time the slope is positive. Concretely, the slope is 0.55 which is stating that the number of new words found follows a square-root law.