char-frequency-visualization

June 25, 2018

```
In [20]: from collections import OrderedDict
         import matplotlib.pyplot as plt

         %matplotlib inline
```

Output of the hadoop job has the following format:

```
In [2]: !head part-r-00000
```

```
!       1575
"       1
#       2
$       2
%       6
&       4
'       1
(       1796
)       1808
*       28
```

Parse file:

```
In [24]: char_occurences = {}
         sum_count = 0
         with open('part-r-00000') as f:
             for line in f:
                 char, count = line.split()
                 count = int(count)
                 char_occurences[char] = count
                 sum_count += count
```

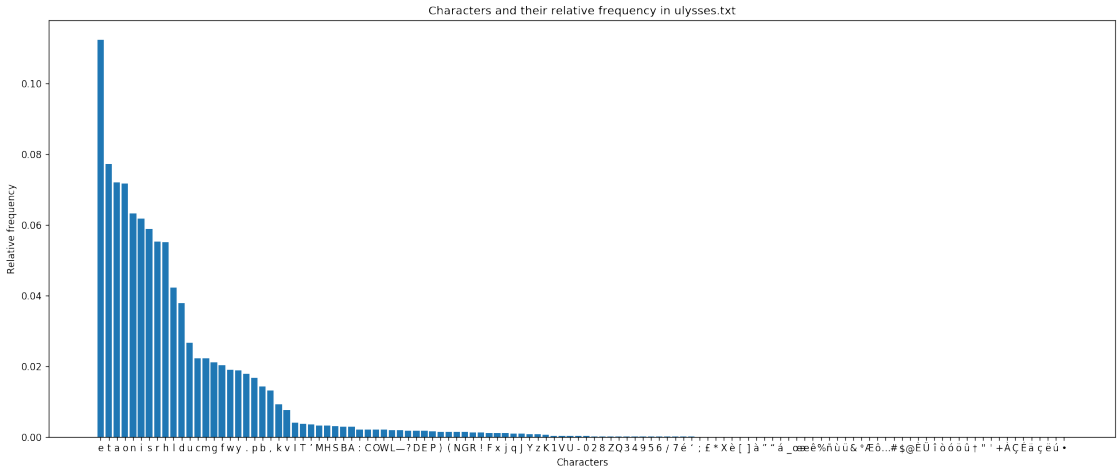
Divide absolute frequencies by total amount of chars to get relative frequency in text:

```
In [31]: for key, val in char_occurences.items():
             char_occurences[key] /= sum_count
```

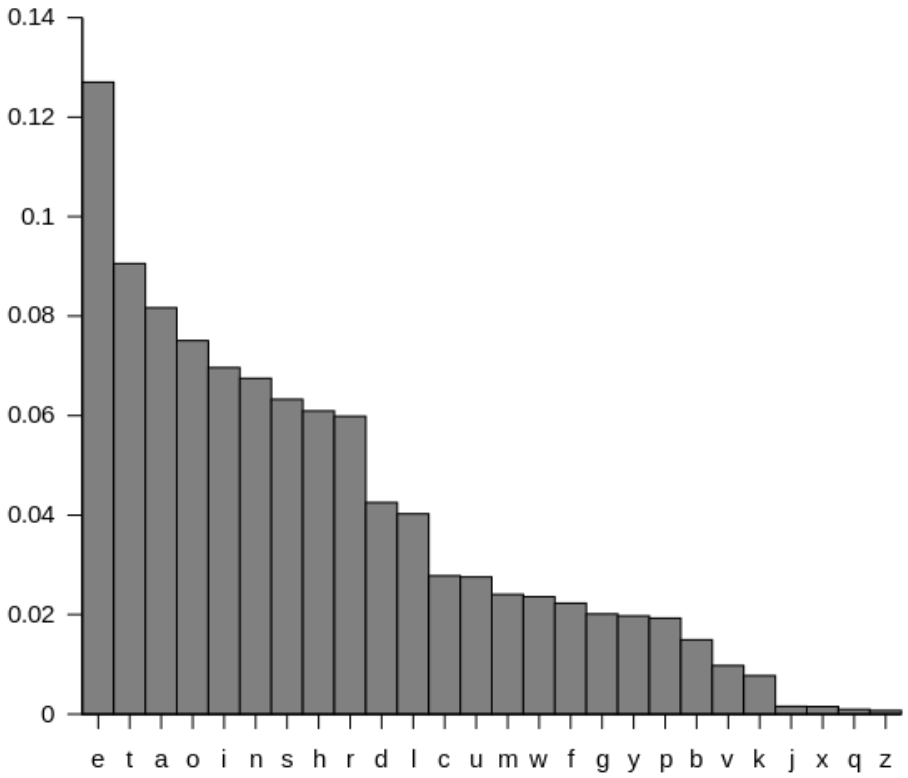
Sort by declining frequency:

```
In [26]: sorted_occurences = sorted(char_occurences.items(), key=lambda x: -x[1])
```

```
In [30]: plt.figure(figsize=(20,8), dpi=120)
         chars, occurences = list(zip(*sorted_occurences))
         plt.title('Characters and their relative frequency in ulysses.txt')
         plt.xlabel('Characters')
         plt.ylabel('Relative frequency')
         plt.bar(chars, occurences);
```



Relative frequencies of characters. (taken from https://en.wikipedia.org/wiki/Letter_frequency#Relative_fre
The frequencies that are outputted when analyzing the `ulysses.txt` file resemble the follow-



ing one: