Estimating the distribution of English alphabets in a set of books

Ramkumar

31-10-2024

Problem definition

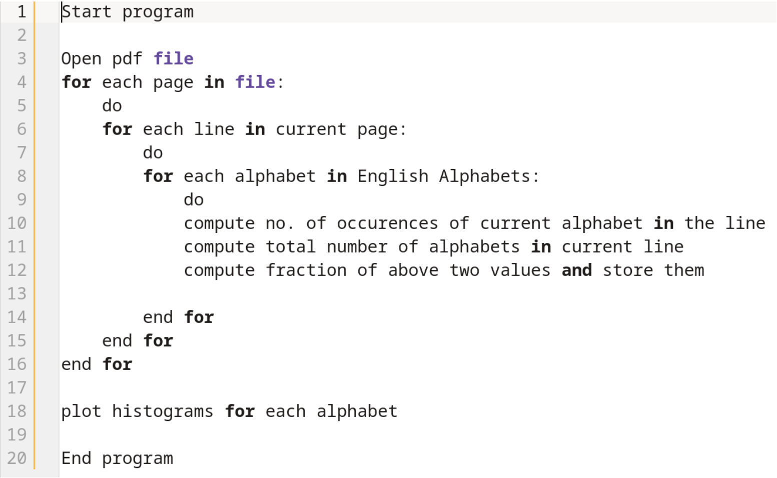
Aim is to estimate the distribution of English alphabets in a set of books and check if it is approaching Normal distribution in Probability.

The following books were taken for this work

- Animal Farm, by George Orwell
- Around the world in eighty days, by Jules Verne
- Flow, by Philip Ball
- For the love of Physics, by Walter Lewin
- Ikigai, by Hector Garcia and Francesc Miralles
- The power of positive thinking, by Norman Vincent

Solution approach

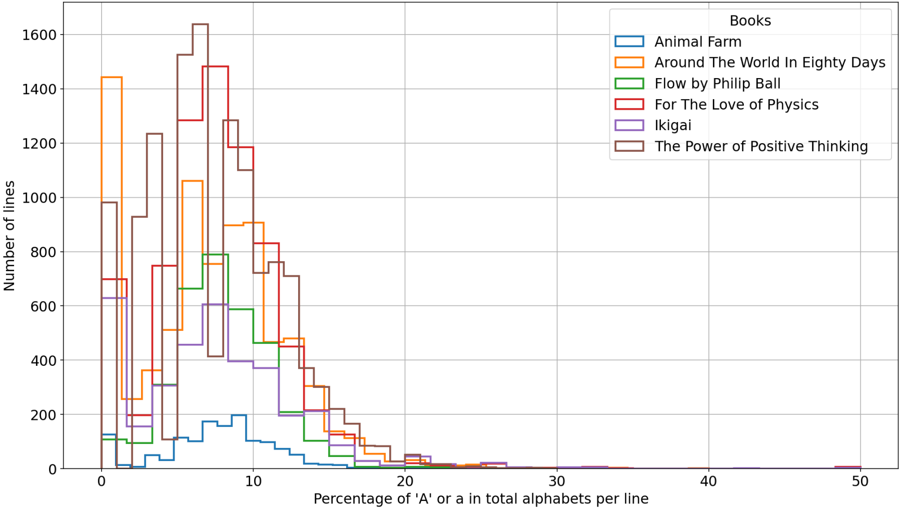
A couple of Python codes were generated to read pdf files and process line-wise data with the following Pseudocode.



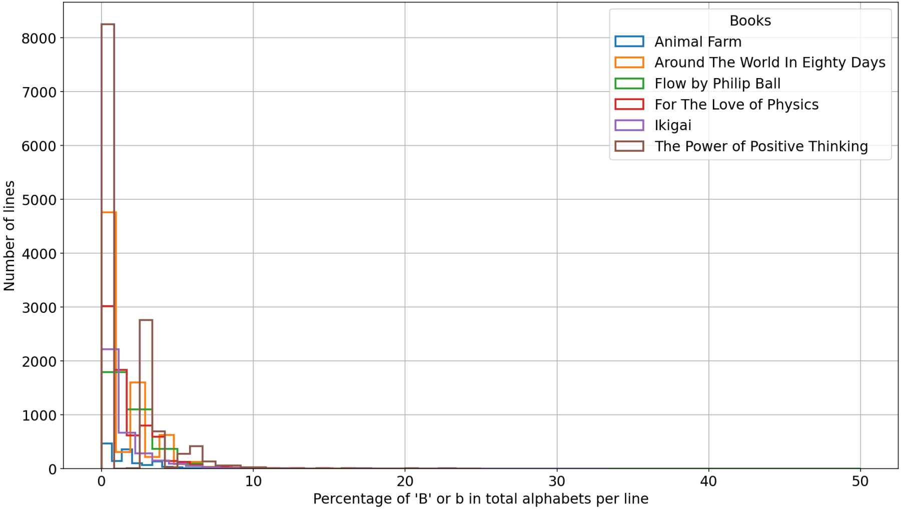
```
Start program

Open pdf file
for each page in file:
    do
    for each line in current page:
        do
        for each alphabet in English Alphabets:
            do
            compute no. of occurences of current alphabet in the line
            compute total number of alphabets in current line
            compute fraction of above two values and store them

        end for
    end for
end for

plot histograms for each alphabet

End program
```

Histograms of all alphabets compared against all chosen books

A or a

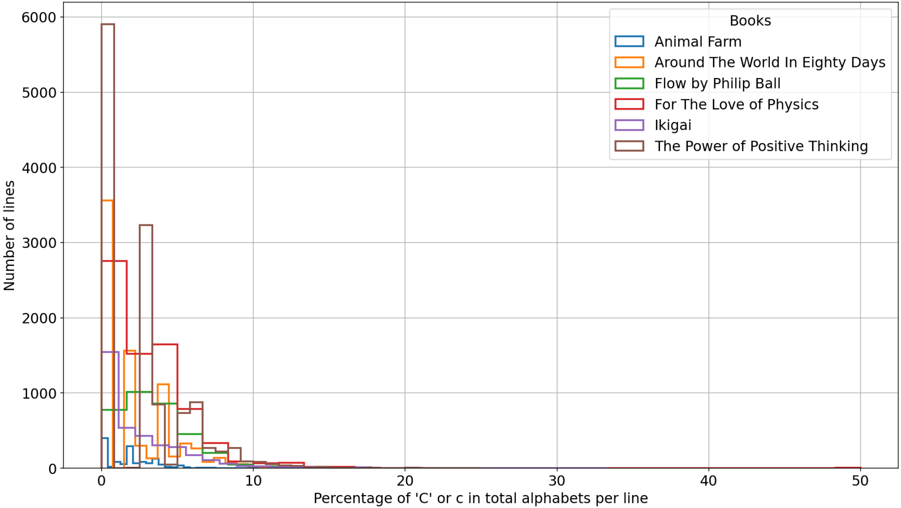

Books
Animal Farm
Around The World In Eighty Days
Flow by Philip Ball
For The Love of Physics
Ikigai
The Power of Positive Thinking
Number of lines
0
200
400
600
800
1000
1200
1400
1600
0
10
20
30
40
50
Percentage of 'A' or a in total alphabets per line

B or b



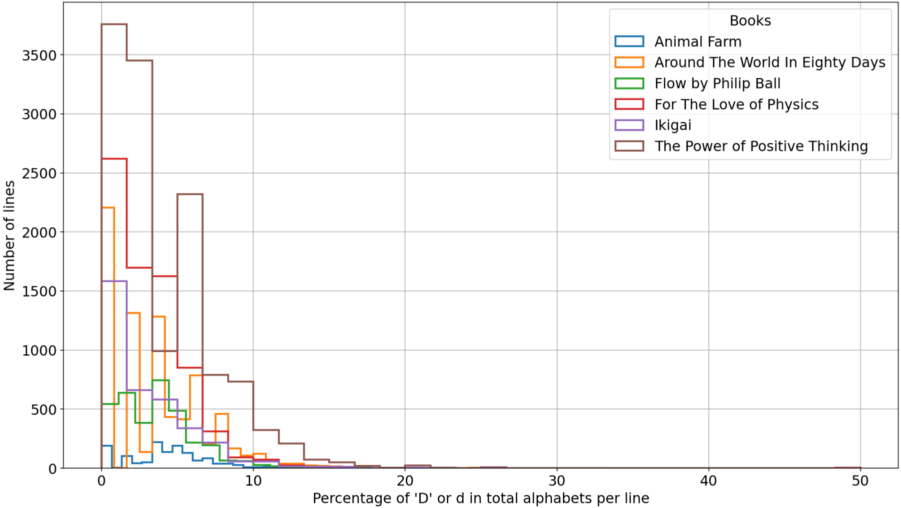
Books
Animal Farm
Around The World In Eighty Days
Flow by Philip Ball
For The Love of Physics
Ikigai
The Power of Positive Thinking
Number of lines
0
1000
2000
3000
4000
5000
6000
7000
8000
0
10
20
30
40
50
Percentage of 'B' or b in total alphabets per line

C or c

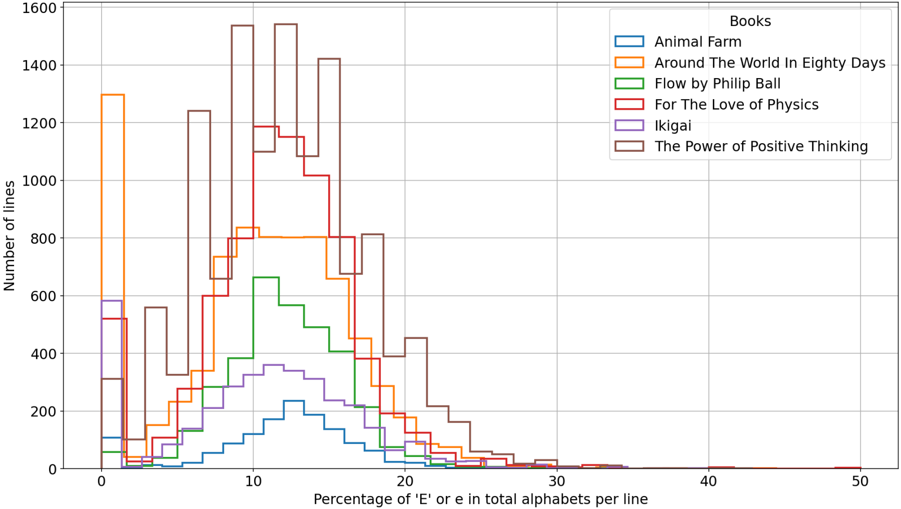


Books
Animal Farm
Around The World In Eighty Days
Flow by Philip Ball
For The Love of Physics
Ikigai
The Power of Positive Thinking
Number of lines
0
1000
2000
3000
4000
5000
6000
0
10
20
30
40
50
Percentage of 'C' or c in total alphabets per line

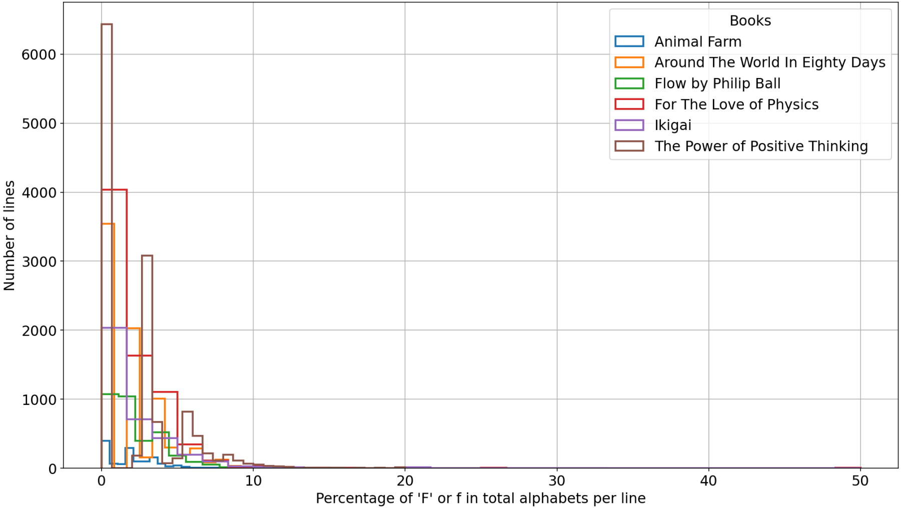
D or d


Books
Animal Farm
Around The World In Eighty Days
Flow by Philip Ball
For The Love of Physics
Ikigai
The Power of Positive Thinking
Number of lines
0
500
1000
1500
2000
2500
3000
3500
0
10
20
30
40
50
Percentage of 'D' or d in total alphabets per line

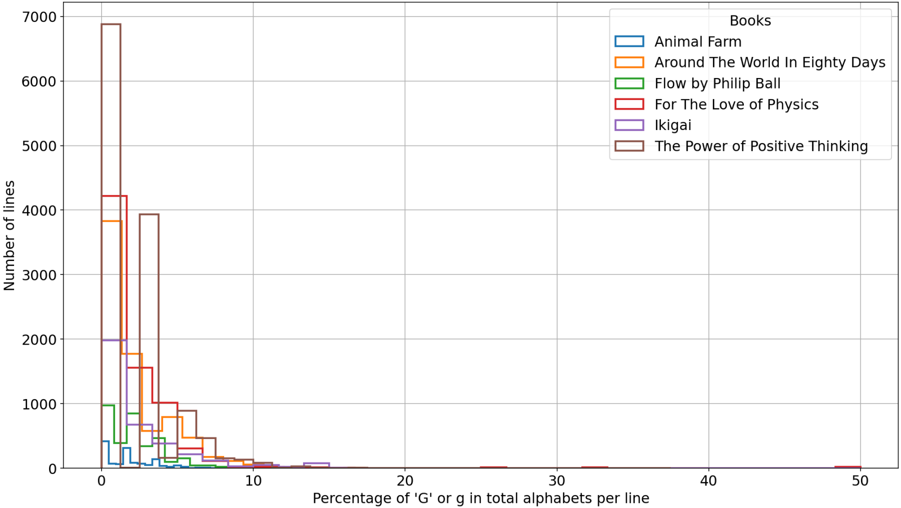
E or e


Books
Animal Farm
Around The World In Eighty Days
Flow by Philip Ball
For The Love of Physics
Ikigai
The Power of Positive Thinking
Number of lines
0
200
400
600
800
1000
1200
1400
1600
0
10
20
30
40
50
Percentage of 'E' or e in total alphabets per line

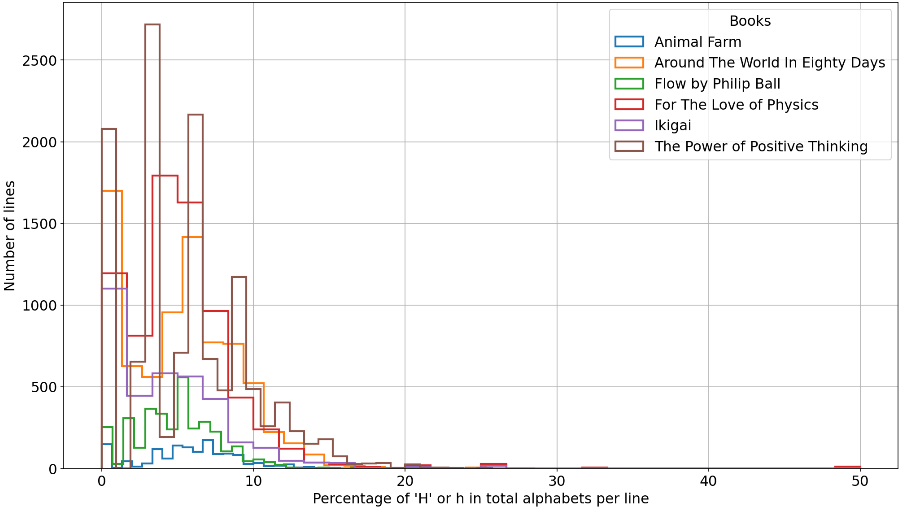
F or f


Books
Animal Farm
Around The World In Eighty Days
Flow by Philip Ball
For The Love of Physics
Ikigai
The Power of Positive Thinking
Number of lines
0
1000
2000
3000
4000
5000
6000
0
10
20
30
40
50
Percentage of 'F' or f in total alphabets per line

G or g

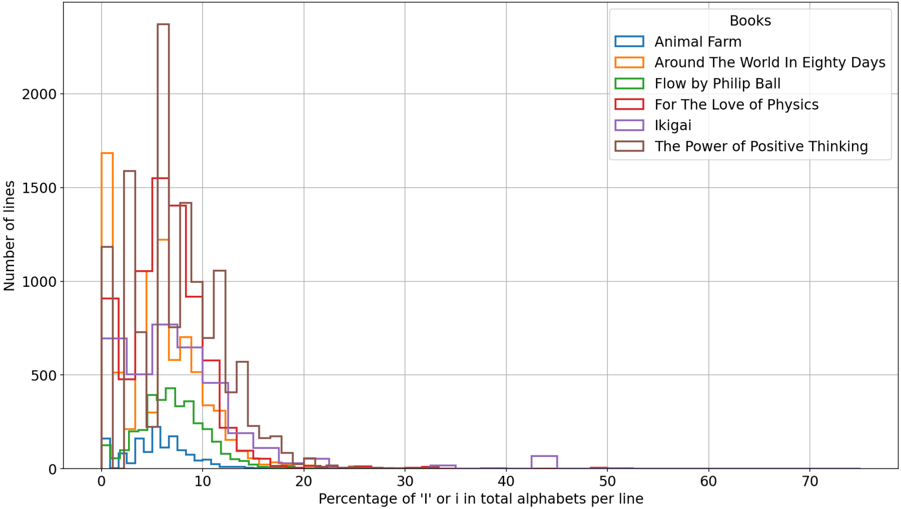

Books
Animal Farm
Around The World In Eighty Days
Flow by Philip Ball
For The Love of Physics
Ikigai
The Power of Positive Thinking
Number of lines
0
1000
2000
3000
4000
5000
6000
7000
0
10
20
30
40
50
Percentage of 'G' or g in total alphabets per line

H or h

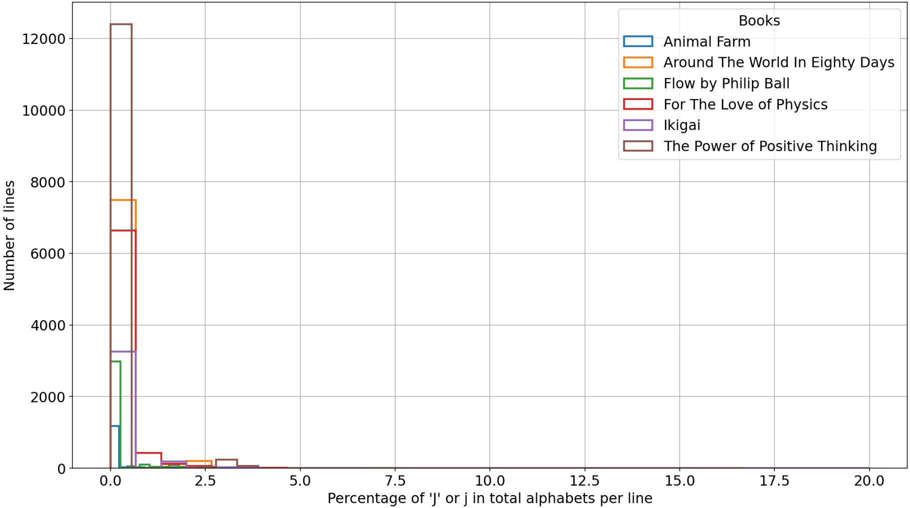


Books
Animal Farm
Around The World In Eighty Days
Flow by Philip Ball
For The Love of Physics
Ikigai
The Power of Positive Thinking
Number of lines
0
500
1000
1500
2000
2500
0
10
20
30
40
50
Percentage of 'H' or h in total alphabets per line

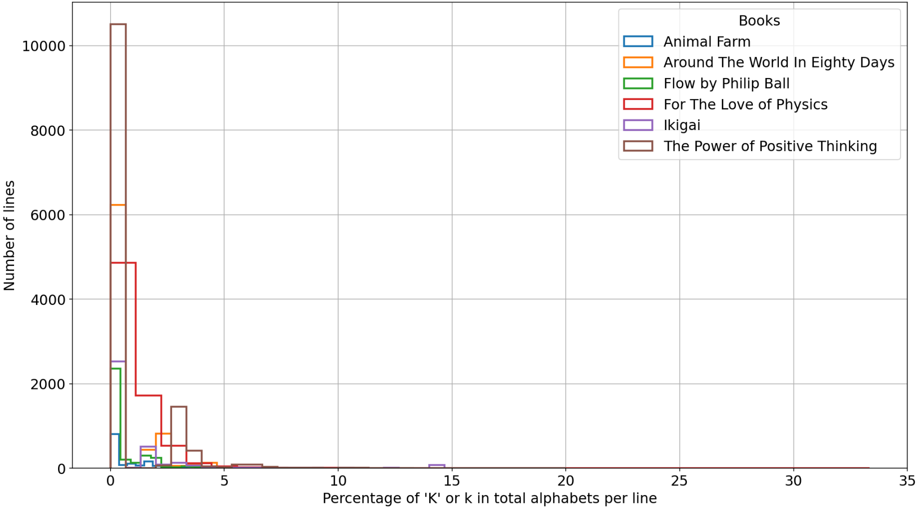
I or i


Books
Animal Farm
Around The World In Eighty Days
Flow by Philip Ball
For The Love of Physics
Ikigai
The Power of Positive Thinking
Number of lines
0
500
1000
1500
2000
0
10
20
30
40
50
60
70
Percentage of 'I' or i in total alphabets per line

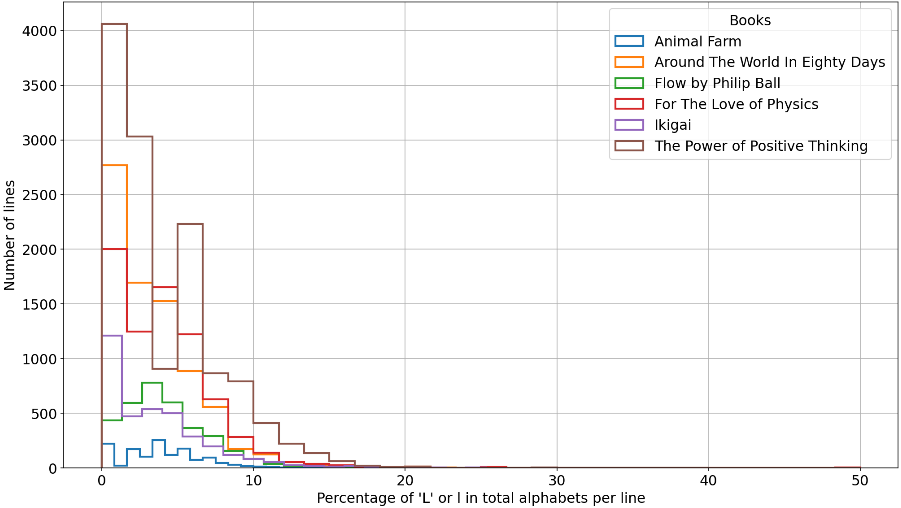
J or j


Books
Animal Farm
Around The World In Eighty Days
Flow by Philip Ball
For The Love of Physics
Ikigai
The Power of Positive Thinking
Number of lines
0
2000
4000
6000
8000
10000
12000
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Percentage of 'J' or j in total alphabets per line

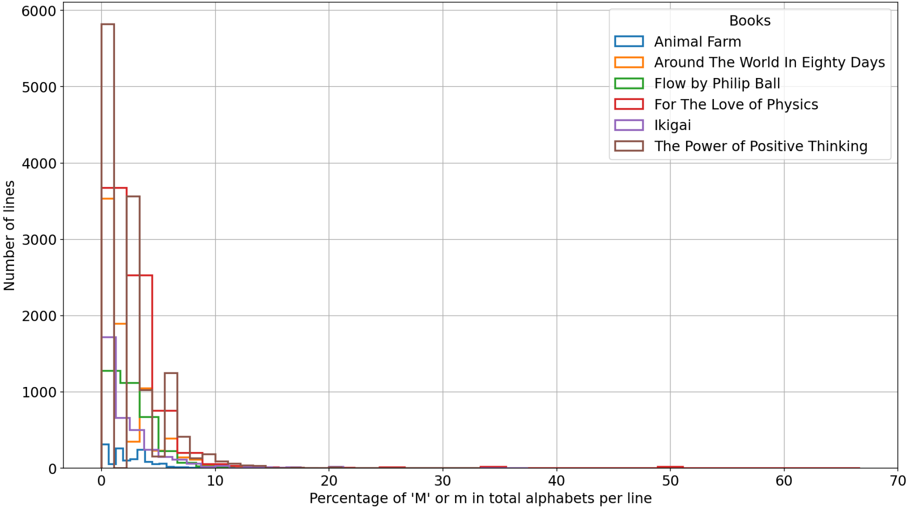
K or k


Books
Animal Farm
Around The World In Eighty Days
Flow by Philip Ball
For The Love of Physics
Ikigai
The Power of Positive Thinking
Number of lines
0
2000
4000
6000
8000
10000
0
5
10
15
20
25
30
35
Percentage of 'K' or k in total alphabets per line

L or l

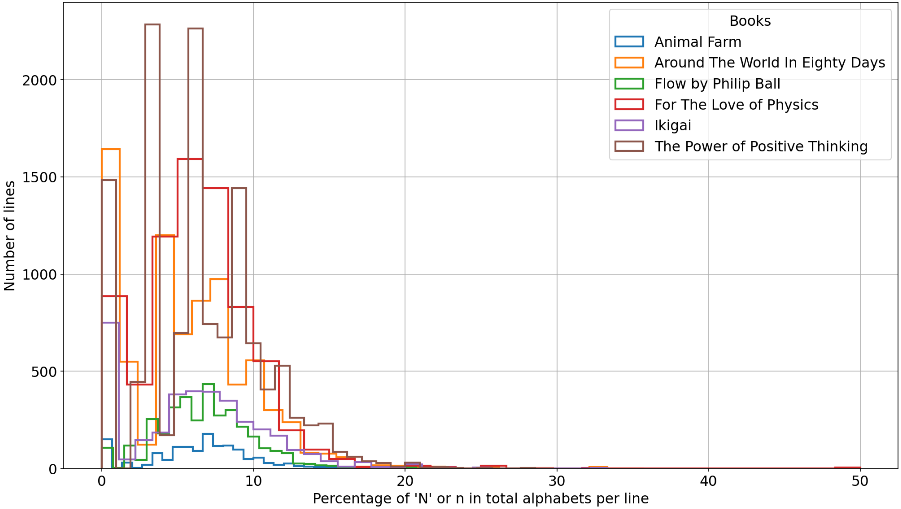

Books
Animal Farm
Around The World In Eighty Days
Flow by Philip Ball
For The Love of Physics
Ikigai
The Power of Positive Thinking
Number of lines
0
500
1000
1500
2000
2500
3000
3500
4000
0
10
20
30
40
50
Percentage of 'L' or l in total alphabets per line

M or m

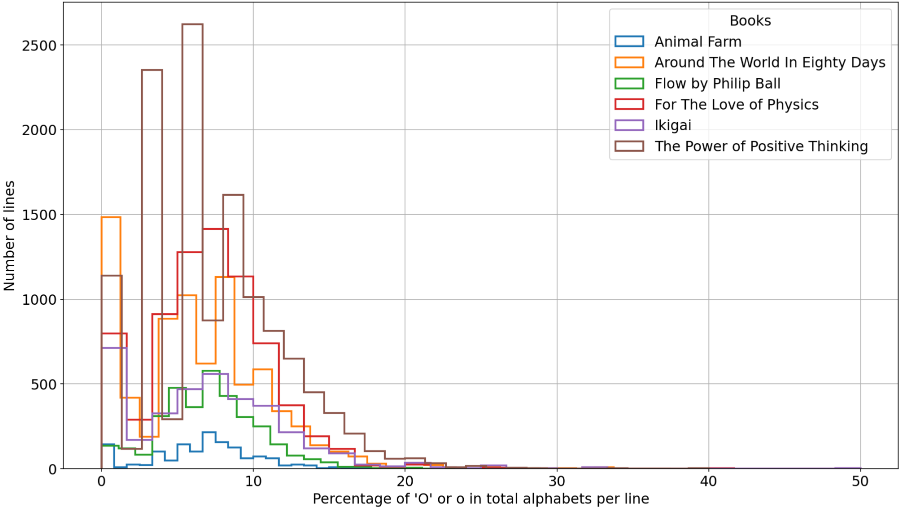


Books
Animal Farm
Around The World In Eighty Days
Flow by Philip Ball
For The Love of Physics
Ikigai
The Power of Positive Thinking
Number of lines
0
1000
2000
3000
4000
5000
6000
0
10
20
30
40
50
60
70
Percentage of 'M' or m in total alphabets per line

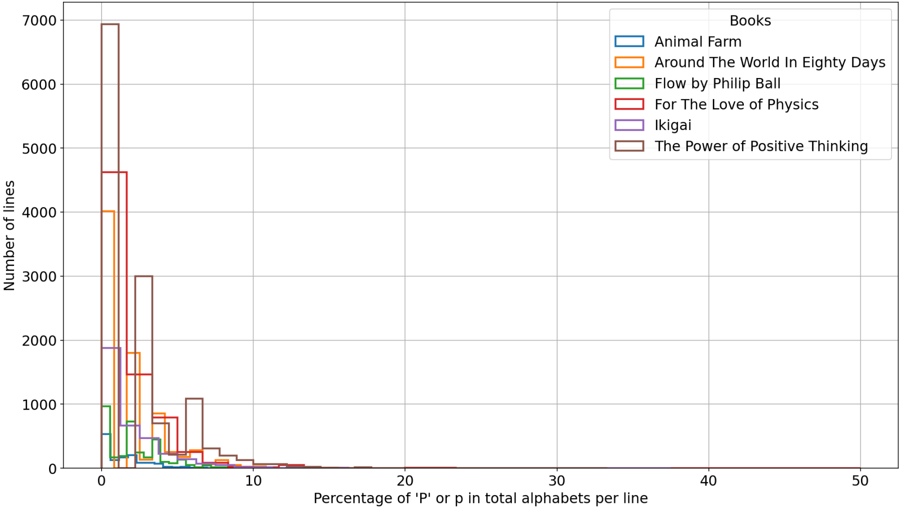
N or n


Books
Animal Farm
Around The World In Eighty Days
Flow by Philip Ball
For The Love of Physics
Ikigai
The Power of Positive Thinking
Number of lines
0
500
1000
1500
2000
0
10
20
30
40
50
Percentage of 'N' or n in total alphabets per line

O or o

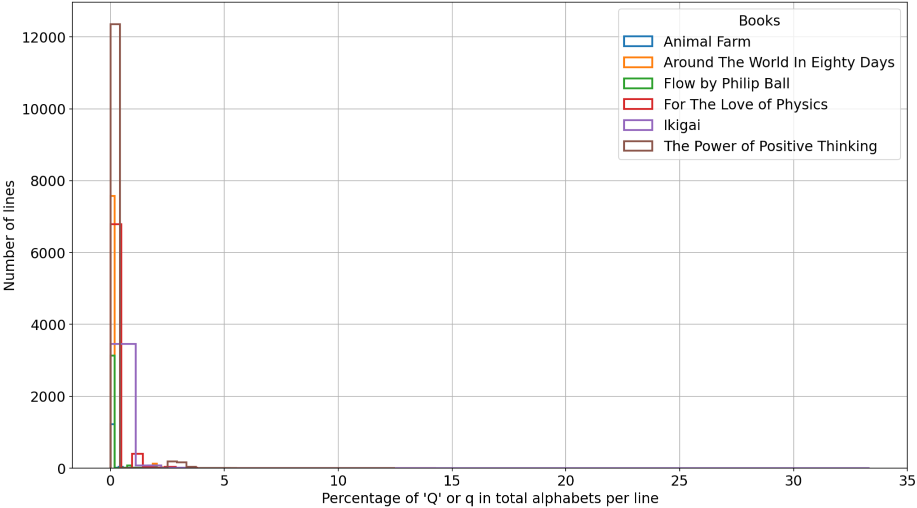

Books
Animal Farm
Around The World In Eighty Days
Flow by Philip Ball
For The Love of Physics
Ikigai
The Power of Positive Thinking
Number of lines
0
500
1000
1500
2000
2500
0
10
20
30
40
50
Percentage of 'O' or o in total alphabets per line

P or p



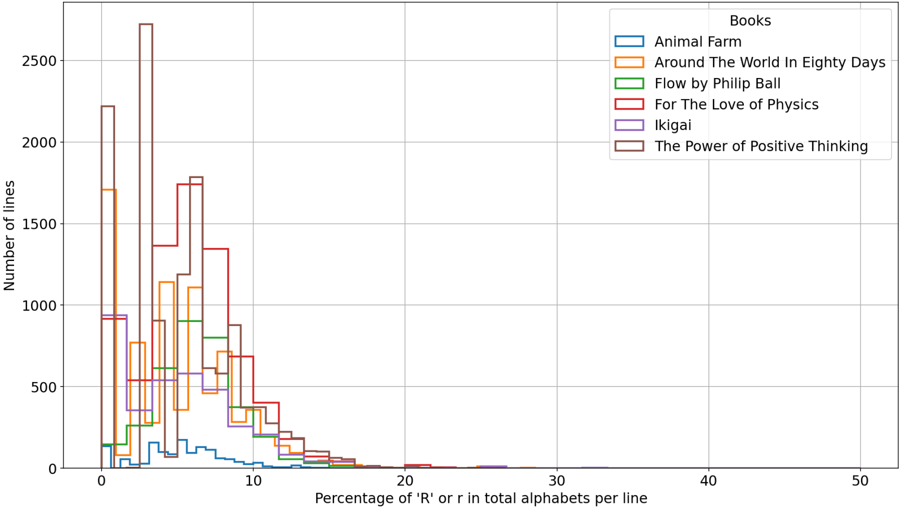
Books
Animal Farm
Around The World In Eighty Days
Flow by Philip Ball
For The Love of Physics
Ikigai
The Power of Positive Thinking
Number of lines
0
1000
2000
3000
4000
5000
6000
7000
0
10
20
30
40
50
Percentage of 'P' or p in total alphabets per line

Q or q



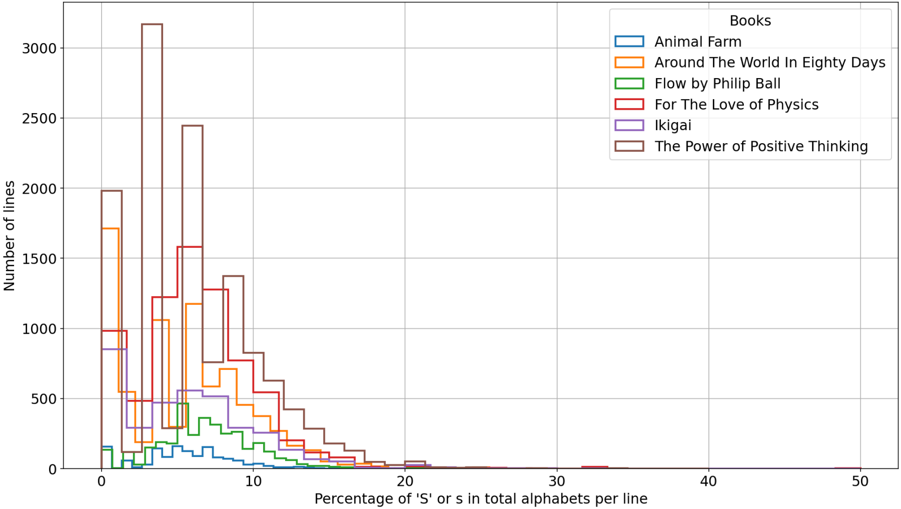
Books
Animal Farm
Around The World In Eighty Days
Flow by Philip Ball
For The Love of Physics
Ikigai
The Power of Positive Thinking
Number of lines
0
2000
4000
6000
8000
10000
12000
0
5
10
15
20
25
30
35
Percentage of 'Q' or q in total alphabets per line

R or r



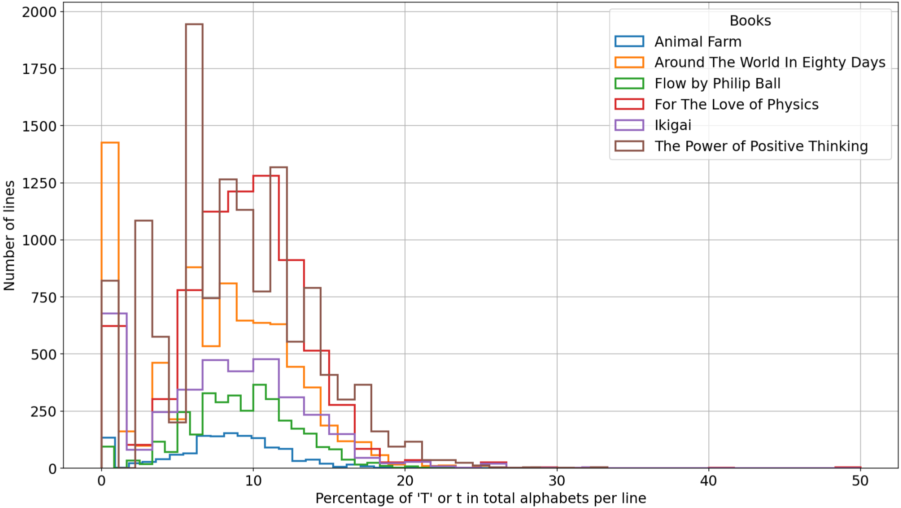
Books
Animal Farm
Around The World In Eighty Days
Flow by Philip Ball
For The Love of Physics
Ikigai
The Power of Positive Thinking
Number of lines
0
500
1000
1500
2000
2500
0
10
20
30
40
50
Percentage of 'R' or r in total alphabets per line

S or s

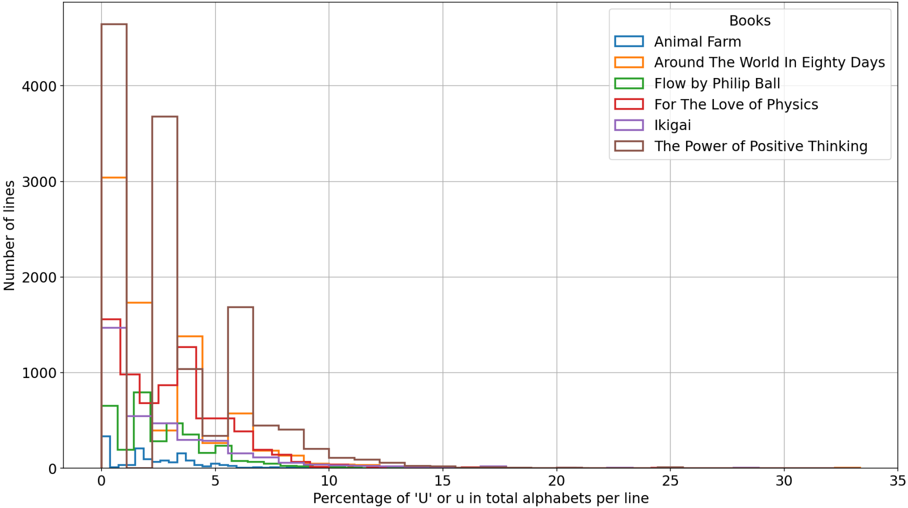


Books
Animal Farm
Around The World In Eighty Days
Flow by Philip Ball
For The Love of Physics
Ikigai
The Power of Positive Thinking
Number of lines
0
500
1000
1500
2000
2500
3000
0
10
20
30
40
50
Percentage of 'S' or s in total alphabets per line

T or t

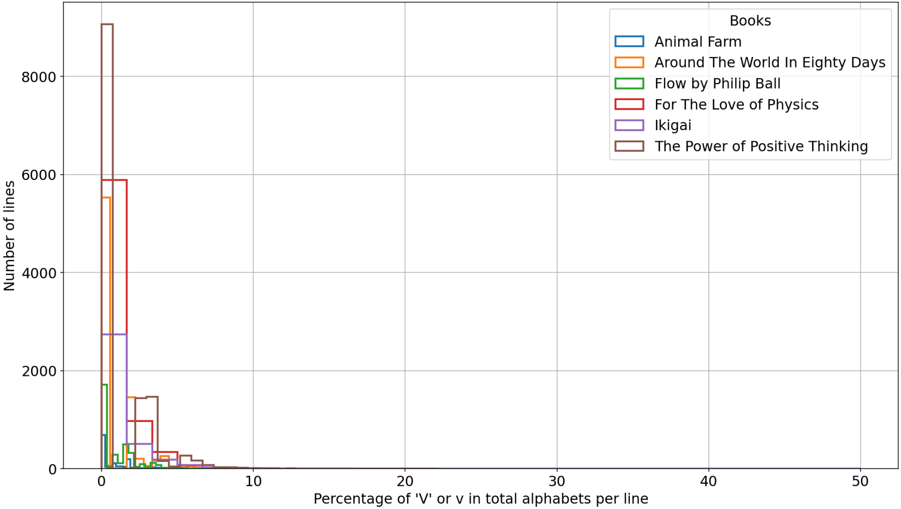

Books
Animal Farm
Around The World In Eighty Days
Flow by Philip Ball
For The Love of Physics
Ikigai
The Power of Positive Thinking
Number of lines
0
250
500
750
1000
1250
1500
1750
2000
0
10
20
30
40
50
Percentage of 'T' or t in total alphabets per line

U or u

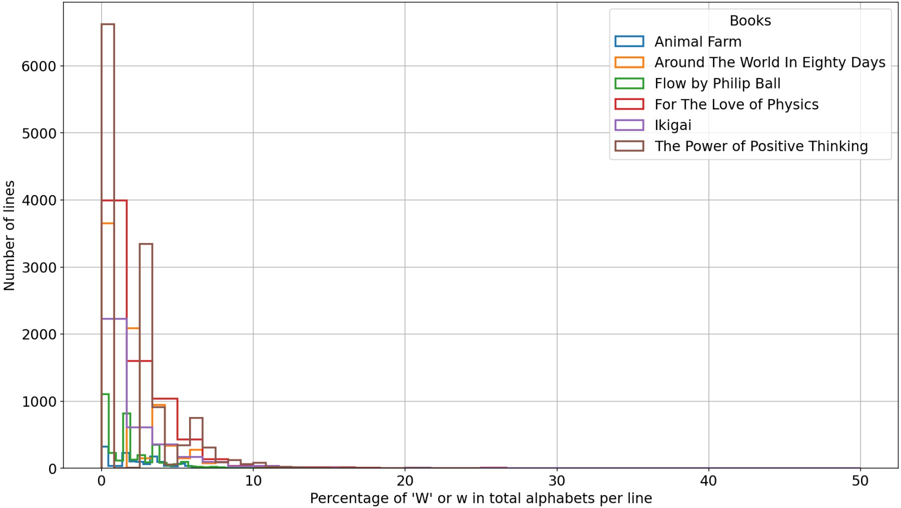


Books
Animal Farm
Around The World In Eighty Days
Flow by Philip Ball
For The Love of Physics
Ikigai
The Power of Positive Thinking
Number of lines
0
1000
2000
3000
4000
0
5
10
15
20
25
30
35
Percentage of 'U' or u in total alphabets per line

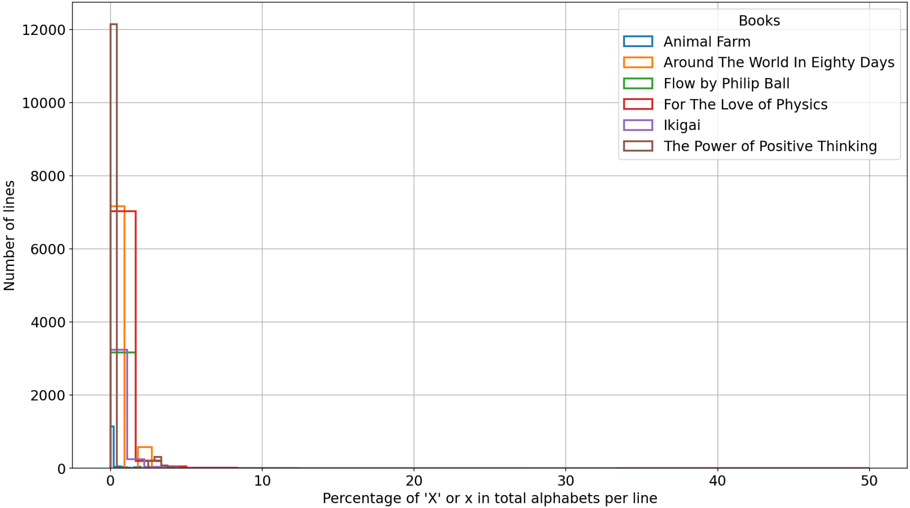
V or v


Books
Animal Farm
Around The World In Eighty Days
Flow by Philip Ball
For The Love of Physics
Ikigai
The Power of Positive Thinking
Number of lines
0
2000
4000
6000
8000
0
10
20
30
40
50
Percentage of 'V' or v in total alphabets per line

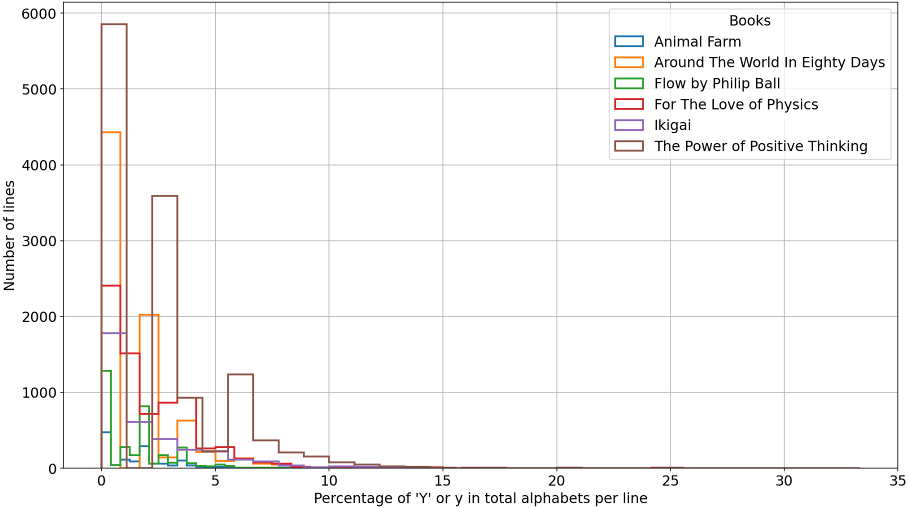
W or w


Books
Animal Farm
Around The World In Eighty Days
Flow by Philip Ball
For The Love of Physics
Ikigai
The Power of Positive Thinking
Number of lines
0
1000
2000
3000
4000
5000
6000
0
10
20
30
40
50
Percentage of 'W' or w in total alphabets per line

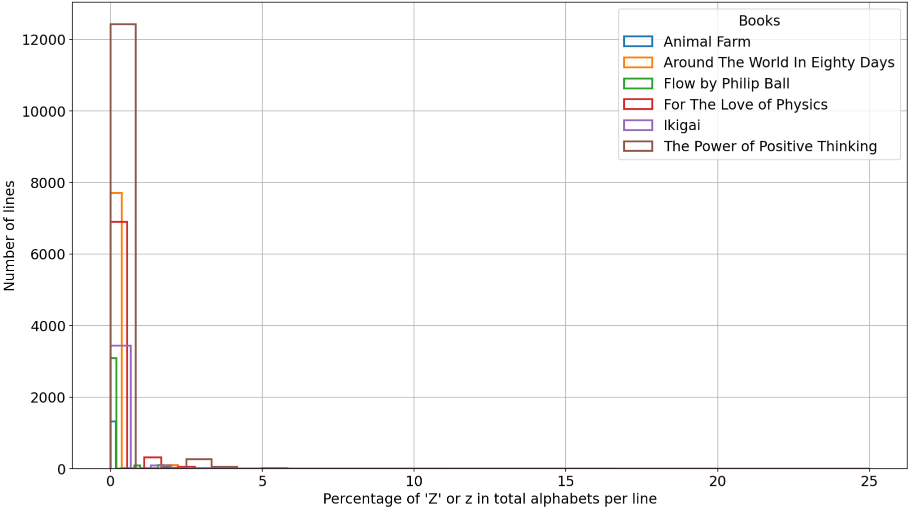
X or x


Books
Animal Farm
Around The World In Eighty Days
Flow by Philip Ball
For The Love of Physics
Ikigai
The Power of Positive Thinking
Number of lines
0
2000
4000
6000
8000
10000
12000
0
10
20
30
40
50
Percentage of 'X' or x in total alphabets per line

Y or y


Books
Animal Farm
Around The World In Eighty Days
Flow by Philip Ball
For The Love of Physics
Ikigai
The Power of Positive Thinking
Number of lines
0
1000
2000
3000
4000
5000
6000
0
5
10
15
20
25
30
35
Percentage of 'Y' or y in total alphabets per line

Z or z


Books
Animal Farm
Around The World In Eighty Days
Flow by Philip Ball
For The Love of Physics
Ikigai
The Power of Positive Thinking
Number of lines
0
2000
4000
6000
8000
10000
12000
0
5
10
15
20
25
Percentage of 'Z' or z in total alphabets per line

Observations

- Least used letters in the chosen books are X, followed by Q and then Z

- Most used letters in the chosen books are E, followed by A, then O and then T

- Most used letters appear to follow **Normal distribution**

- Least used letters appear to follow **Exponential distribution**

Python code - Reader script

Reader script I

```
#!/bin/python3
"""==========================================================================
Alphabet statistics generator from pdf books

Ramkumar
Wed Oct 30 04:55:14 PM IST 2024
=========================================================================="""

# importing needed modules
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from pypdf import PdfReader
import os,glob

#===========================================================================

# specifying names of books and its contained directory
bookNames = ["For The Love of Physics.pdf",
             "Around The World In Eighty Days.pdf",
             "Flow by Philip Ball.pdf",
             "Animal Farm.pdf",
             "The Power of Positive Thinking.pdf",
             "Ikigai.pdf"]
directory = "books/"

# specifying starting and ending page numbers to exclude title, index etc...
pageStart = [10,4,12,5,5,8]
pageEnd   = [161,320,189,83,700,115]

```

Reader script II

```python
# creating a directory to store line-wise data files
os.system("rm -rf linewise_data && mkdir linewise_data")

# looping through books
for I in range(len(bookNames)):
    # preparing book name
    bookName = directory + "/" + bookNames[I]

    print("reading book : ",bookNames[I])

    # reading book
    reader = PdfReader(bookName)

    # getting number of pages
    N_pages = len(reader.pages)

    # setting index to start page and end page numbers in python indexing
    start = pageStart[I]
    end   = pageEnd[I]

    # extracting lines
    total_lines = []
    for i in range(start,end):
        # extracting text from current page
        page = reader.pages[i]
        content = page.extract_text()

        # extracting lines from the content
        lines = content.split("\n")
        total_lines.extend(lines)

```

Reader script III

```
        print("reading page : ",i-start+1," of ",end-start+1)

    # counting number of characters in each line
    a_list = []; b_list = []; c_list = []; d_list = []; e_list = []; f_list = []
    g_list = []; h_list = []; i_list = []; j_list = []; k_list = []; l_list = []
    m_list = []; n_list = []; o_list = []; p_list = []; q_list = []; r_list = []
    s_list = []; t_list = []; u_list = []; v_list = []; w_list = []; x_list = []
    y_list = []; z_list = []; line_length = [];

    idx = 1
    for line in total_lines:
        # extracting total length of alpha characters
        length = len([char for char in line if char.isalpha()])

        # extracting total number of each alphabets
        a_len = len([char for char in line if char == 'a' or char == 'A'])
        b_len = len([char for char in line if char == 'b' or char == 'B'])
        c_len = len([char for char in line if char == 'c' or char == 'C'])
        d_len = len([char for char in line if char == 'd' or char == 'D'])
        e_len = len([char for char in line if char == 'e' or char == 'E'])
        f_len = len([char for char in line if char == 'f' or char == 'F'])
        g_len = len([char for char in line if char == 'g' or char == 'G'])
        h_len = len([char for char in line if char == 'h' or char == 'H'])
        i_len = len([char for char in line if char == 'i' or char == 'I'])
        j_len = len([char for char in line if char == 'j' or char == 'J'])
        k_len = len([char for char in line if char == 'k' or char == 'K'])
        l_len = len([char for char in line if char == 'l' or char == 'L'])
        m_len = len([char for char in line if char == 'm' or char == 'M'])
        n_len = len([char for char in line if char == 'n' or char == 'N'])
        o_len = len([char for char in line if char == 'o' or char == 'O'])
        p_len = len([char for char in line if char == 'p' or char == 'P'])
```

Reader script IV

```
        q_len = len([char for char in line if char == 'q' or char == 'Q'])
        r_len = len([char for char in line if char == 'r' or char == 'R'])
        s_len = len([char for char in line if char == 's' or char == 'S'])
        t_len = len([char for char in line if char == 't' or char == 'T'])
        u_len = len([char for char in line if char == 'u' or char == 'U'])
        v_len = len([char for char in line if char == 'v' or char == 'V'])
        w_len = len([char for char in line if char == 'w' or char == 'W'])
        x_len = len([char for char in line if char == 'x' or char == 'X'])
        y_len = len([char for char in line if char == 'y' or char == 'Y'])
        z_len = len([char for char in line if char == 'z' or char == 'Z'])

        # appending to the lists
        a_list.append(a_len); b_list.append(b_len); c_list.append(c_len)
        d_list.append(d_len); e_list.append(e_len); f_list.append(f_len)
        g_list.append(g_len); h_list.append(h_len); i_list.append(i_len)
        j_list.append(j_len); k_list.append(k_len); l_list.append(l_len)
        m_list.append(m_len); n_list.append(n_len); o_list.append(o_len)
        p_list.append(p_len); q_list.append(q_len); r_list.append(r_len)
        s_list.append(s_len); t_list.append(t_len); u_list.append(u_len)
        v_list.append(v_len); w_list.append(w_len); x_list.append(x_len)
        y_list.append(y_len); z_list.append(z_len); line_length.append(length)

        print("processing line = ",idx," of ",len(total_lines))
        idx += 1

    # preparing pandas dataframe to store the results
    filename = "alphabetCount_"+bookNames[l].split(".pdf")[0]+".csv"
    fid = pd.DataFrame(np.transpose([
                    a_list, b_list, c_list, d_list, e_list, f_list, g_list,
                    h_list, i_list, j_list, k_list, l_list, m_list, n_list,
                    o_list, p_list, q_list, r_list, s_list, t_list, u_list,
```

Reader script V

```
                    v_list, w_list, x_list, y_list, z_list, line_length]),
                       columns = ["a","b","c","d","e","f","g",
                                  "h","i","j","k","l","m","n",
                                  "o","p","q","r","s","t","u",
                                  "v","w","x","y","z","line_length"])
    fid.to_csv("linewise_data/"+filename, index = None)

print("done")

#==============================================================================
```

Python code - Post-processing script

Post-processing script I

```python
#!/bin/python3
"""==========================================================================
post processing for the script_reader.py

Ramkumar
Wed Oct 30 06:12:10 PM IST 2024
=========================================================================="""

# importing needed modules
import pandas as pd
import matplotlib.pyplot as plt
import os,glob

#==========================================================================

# reading data files
fileNames = sorted(glob.glob1(os.getcwd()+"/linewise_data/","*.csv"))

# preparing booknames
bookNames = [name.split("alphabetCount_")[1].split(".csv")[0] for name in fileNames]

# preparing list to store line-normalized character values
a_list = []; b_list = []; c_list = []; d_list = []; e_list = []; f_list = []
g_list = []; h_list = []; i_list = []; j_list = []; k_list = []; l_list = []
m_list = []; n_list = []; o_list = []; p_list = []; q_list = []; r_list = []
s_list = []; t_list = []; u_list = []; v_list = []; w_list = []; x_list = []
y_list = []; z_list = []

charArray = ["a","b","c","d","e","f","g","h","i","j","k","l","m","n",
```

Post-processing script II

```
                "o","p","q","r","s","t","u","v","w","x","y","z"]

# looping through the data files
for file in fileNames:
    # reading data
    fid = pd.read_csv("linewise_data/"+file)

    # looping through characters, normalizing and storing 'em in the lists
    for char in charArray:
        fid[char] = fid[char]/(fid["line_length"]+1)*100
        eval(char+"_list.append(fid[\""+char+"\"])")

# preparing directory to store graphs
os.system("rm -rf histograms && mkdir histograms")

# preparing plots and saving them
for char in charArray:

    # getting the current list
    exec("curr_list = "+char+"_list")

    # plotting histogram
    plt.rcParams.update({"font.size":15})
    plt.figure(figsize=(16,9))
    for i in range(len(bookNames)):
        plt.hist(curr_list[i],bins=30,histtype="step",label=bookNames[i],
                 density=False,linewidth=2)
    plt.grid()
    plt.xlabel("Percentage of \'"+char.upper()+"\' or "+char+" in total
     alphabets per line")
    plt.ylabel("Number of lines")
```

Post-processing script III

```
    #  plt.title("Alphabet : "+char.upper()+" or "+char)
    #  plt.legend(loc=(1.01,0.75))
    plt.legend(title="Books")
    plt.savefig("histograms/"+char+".png",dpi=150,bbox_inches="tight")
    plt.close()

    print("Character : ",char.upper())

print("done")

#==============================================================================
```

End