The Frequency of a Word by its Length

Computational Science

Written by

Tom Erez

Submitted on 29th of September 2024

Project written for unit 1 in Computational Science at Hemda,

Schwartz-Reisman Science Education Center, Tel Aviv.

This project was written under guidance by Shlomo Rozenfield in 2024.

Abstract

In this project, I aim to explore various aspects linked to the distribution of word lengths in literature across different eras and authors. The key objective is to investigate whether specific authors or time periods exhibit distinctive characteristics of word length frequency and relative frequency. Furthermore, based on these properties I aim to predict the era or author of multiple books. For my investigation I will use literature texts from the **Gutenberg Project**.

Table of Contents:

Introduction: Historical & Theoretical Background

A linguist is a person who studies languages and their structure also called linguistics. For many years, there was an unsolved mystery dealing with the description of how the frequency of a word length in a natural language is dependent on its rank in the frequency table. In the early 1920's Felix Auerbach, a German physicist, observed an inverse proportionality between the population sizes of cities, and their ranks when sorted by decreasing order of that variable. Later in 1935, George Zipf, a famous linguist, observed that the same law applies to the frequency of a word length and their rank in a frequency table. Today this law is called Zipf's Law which is an empirical law that often holds, approximately, when a list of measured values is sorted in decreasing order. It states that the value of the *n*th entry is inversely proportional to *n*.

$$\text{frequency} \propto \frac{1}{(\text{rank} + b)^a}$$

Above is the mathematical form of Zipf's Law where ***a*** and ***b*** are constants and

a ≈ 1

b ≈ 2.7

Zipf's Law doesn't apply only to linguistics, it also applies to subjects such as city population ,as Felix Auerbach discovered and economic activity (a few large entities dominate the market, while numerous entities make up the rest).

As of right now, Zipf's Law hasn't been explained and scientists are unsure about the reason for its occurrence. Many researchers have tried to explain this phenomenon including Zipf himself.

Wentian Li attempted to explain this phenomenon by creating randomly generated texts and later seeing that these texts follow the macro-trend of Zipf's Law (the more probable words are shorter and have equal probability). Although this doesn't explain Zipf's Law it shows that even in randomly generated texts shorter words are more probable. When Zipf tried to explain his observation he argued that the principle of least effort is the reason for this law. He proposes that the speaker and listener want to use the least effort in order to reach an understanding when speaking which therefore leads to approximately equal distribution of effort leads to Zipf's Law. Understanding this relationship is important because it demonstrates a fundamental property found in natural languages across the world.

My Code

General Approach:

In this project I used the following approach to investigate the link between word length and its frequency.

- Firstly I chose books from the Gutenberg Project.
- Each book is given in txt file format. For each file I extracted a list containing all the words. When relevant I combined in the list words from multiple books with similar traits such as author or era.
- For each list of words, I computed a dictionary that includes the word length and the frequency of word length.
- I investigated the relationship between word length and its frequency using the principles of Zipf's Law and visualizations such as histograms, tables and graphs.
- Finally I used the results of my analysis to predict the author or era of a given book.

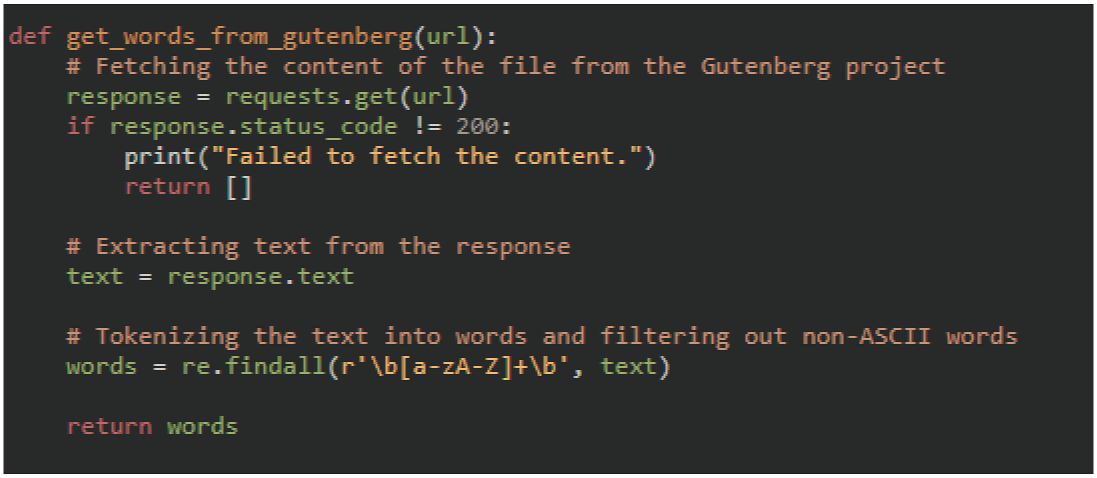
Below I will describe the code and details according to this approach.

Creating Dictionaries from Txt Files:

Below are the basic functions I used to render a txt file into dictionaries of frequency tables with the lengths of the word and their relative frequency:

get_words_from_gutenberg:

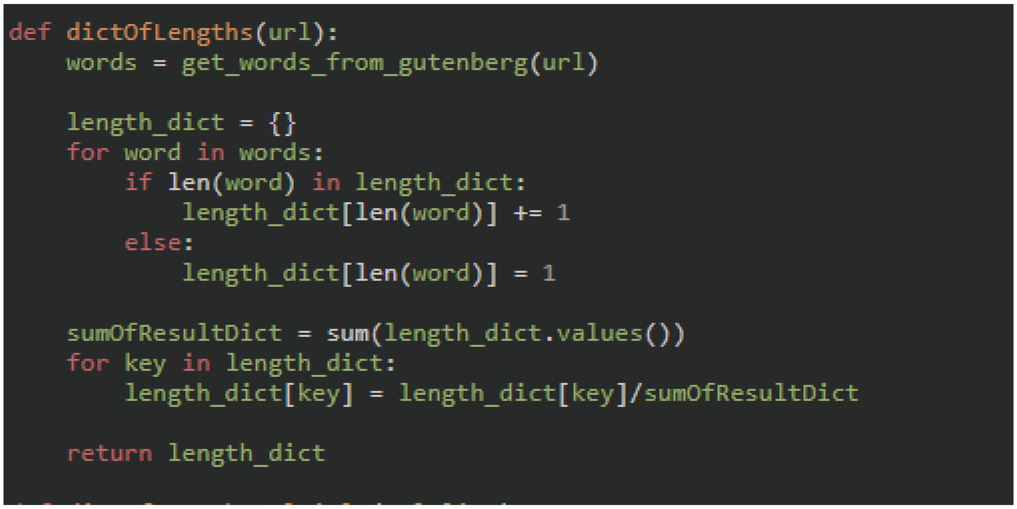
The function receives a txt file from the Gutenberg Project and returns a list of all the words in the file.



```python
def get_words_from_gutenberg(url):
    # Fetching the content of the file from the Gutenberg project
    response = requests.get(url)
    if response.status_code != 200:
        print("Failed to fetch the content.")
        return []

    # Extracting text from the response
    text = response.text

    # Tokenizing the text into words and filtering out non-ASCII words
    words = re.findall(r'\b[a-zA-Z]+\b', text)

    return words
```

dictOfLengths:

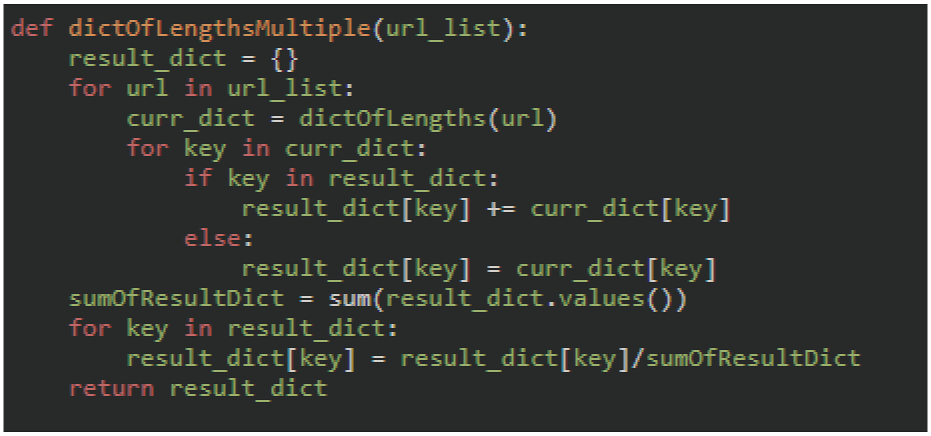
The function receives a txt file from the Gutenberg Project and returns a dictionary in which the keys are the lengths of the words and the values are the relative frequency of each word length in the given file.



```python
def dictOfLengths(url):
    words = get_words_from_gutenberg(url)

    length_dict = {}
    for word in words:
        if len(word) in length_dict:
            length_dict[len(word)] += 1
        else:
            length_dict[len(word)] = 1

    sumOfResultDict = sum(length_dict.values())
    for key in length_dict:
        length_dict[key] = length_dict[key]/sumOfResultDict

    return length_dict
```

dictOfLengthsMultiple:

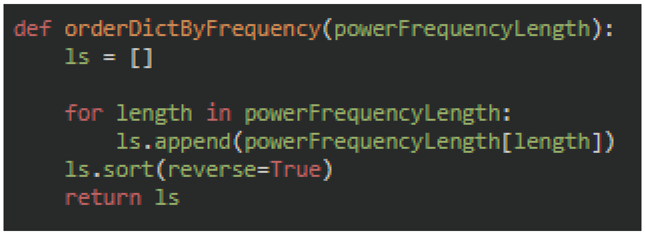
The function receives a list of txt files and returns a dictionary in which the keys are the lengths of the words and the values are the relative frequency of each word length across all the files.



```
def dictOfLengthsMultiple(url_list):
    result_dict = {}
    for url in url_list:
        curr_dict = dictOfLengths(url)
        for key in curr_dict:
            if key in result_dict:
                result_dict[key] += curr_dict[key]
            else:
                result_dict[key] = curr_dict[key]
    sumOfResultDict = sum(result_dict.values())
    for key in result_dict:
        result_dict[key] = result_dict[key]/sumOfResultDict
    return result_dict
```

orderDictByFrequency:

The function receives a dictionary containing the length and frequency of words in a file or files and returns a list of the frequency ordered by rank. The rank of an item is its position in an ordered series.



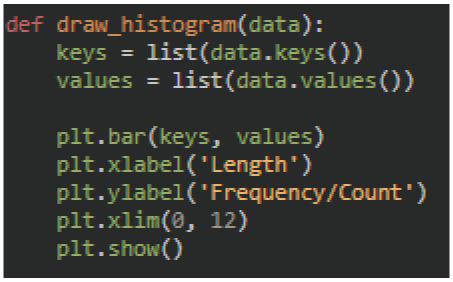
```
def orderDictByFrequency(powerFrequencyLength):
    ls = []

    for length in powerFrequencyLength:
        ls.append(powerFrequencyLength[length])
    ls.sort(reverse=True)
    return ls
```

Visualization and Analysis of the Data:

Below are the functions used to visualize and analyze the data after being rendered:

draw_histogram:

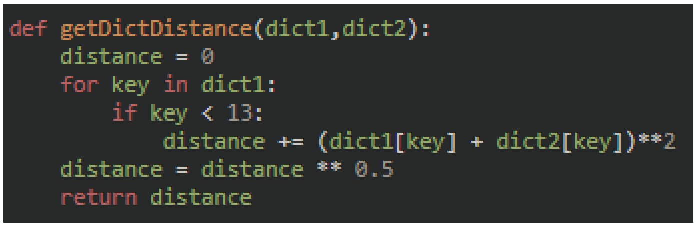
The function receives a dictionary and plots a histogram of the length of words. The y-axis is the relative frequency of word length, meaning the frequency of length words divided by the total number of words.



```
def draw_histogram(data):
    keys = list(data.keys())
    values = list(data.values())

    plt.bar(keys, values)
    plt.xlabel('Length')
    plt.ylabel('Frequency/Count')
    plt.xlim(0, 12)
    plt.show()
```

getDictDistance:

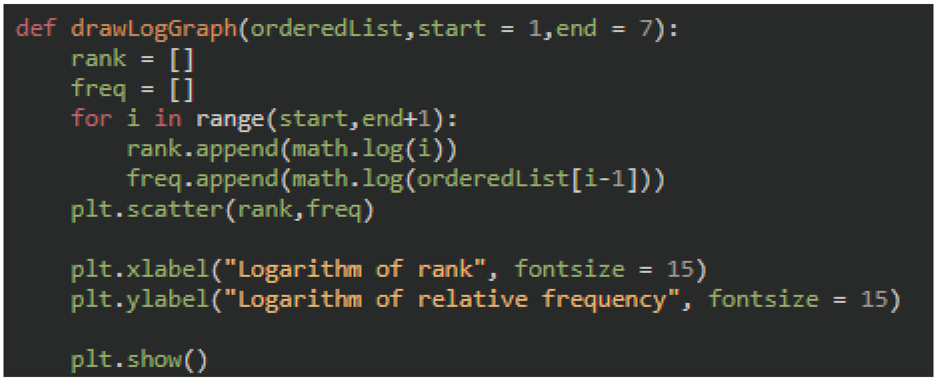
The function receives two dictionaries and returns the geometric distance between the dictionaries. Words with length 13 or more were not included in the calculation because they are anomalies such as url links and text notations.



```
def getDictDistance(dict1,dict2):
    distance = 0
    for key in dict1:
        if key < 13:
            distance += (dict1[key] + dict2[key])**2
    distance = distance ** 0.5
    return distance
```

drawLogGraph:

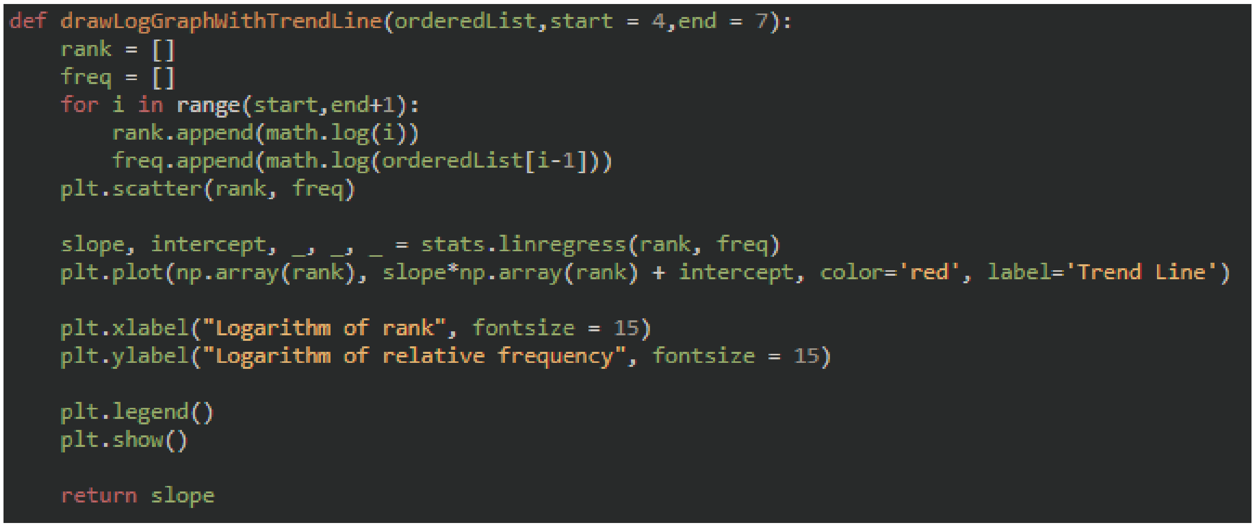
The function receives a list of frequencies ordered by their rank and plots a graph of the log of the frequency depending on the log of the rank of the 7 highest frequencies.



```
def drawLogGraph(orderedList,start = 1,end = 7):
    rank = []
    freq = []
    for i in range(start,end+1):
        rank.append(math.log(i))
        freq.append(math.log(orderedList[i-1]))
    plt.scatter(rank,freq)

    plt.xlabel("Logarithm of rank", fontsize = 15)
    plt.ylabel("Logarithm of relative frequency", fontsize = 15)

    plt.show()
```

drawLogGraphWithTrendLine:

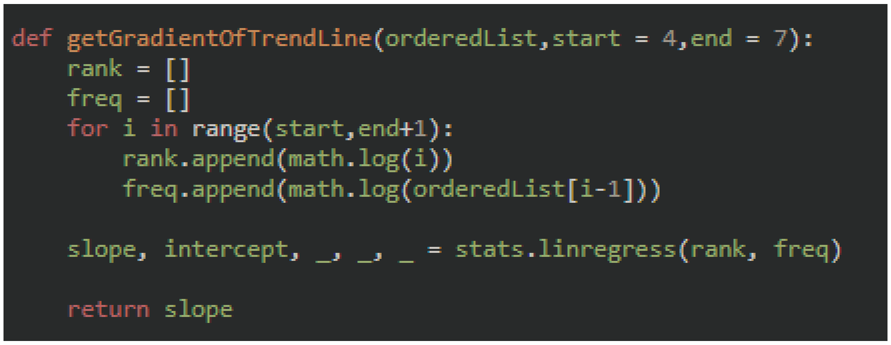
The function receives a list with frequencies ordered by rank and returns a graph of the log of the frequency depending on the log of the rank and the trend line of the graph in the range between the ranks 4 and 7. The function returns the slope of the trend line.



```
def drawLogGraphWithTrendLine(orderedList,start = 4,end = 7):
    rank = []
    freq = []
    for i in range(start,end+1):
        rank.append(math.log(i))
        freq.append(math.log(orderedList[i-1]))
    plt.scatter(rank, freq)

    slope, intercept, _, _, _ = stats.linregress(rank, freq)
    plt.plot(np.array(rank), slope*np.array(rank) + intercept, color='red', label='Trend Line')

    plt.xlabel("Logarithm of rank", fontsize = 15)
    plt.ylabel("Logarithm of relative frequency", fontsize = 15)

    plt.legend()
    plt.show()

    return slope
```

getGradientOfTrendLine:

This function receives a list of frequencies ordered by their rank and returns the slope of the trend line of the graph of the log of the frequency depending on the log of the rank between the ranks 4 and 7.



```
def getGradientOfTrendLine(orderedList,start = 4,end = 7):
    rank = []
    freq = []
    for i in range(start,end+1):
        rank.append(math.log(i))
        freq.append(math.log(orderedList[i-1]))

    slope, intercept, _, _, _ = stats.linregress(rank, freq)

    return slope
```

Investigating Books from Different Eras

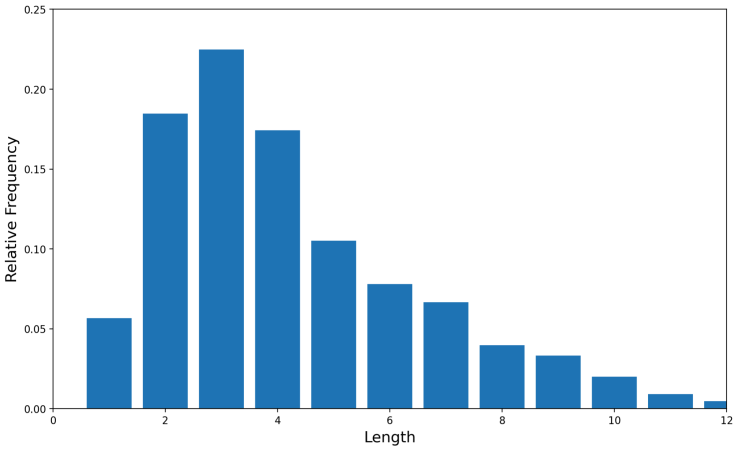
In this section, I am looking into books from different eras to see their similarities and differences. I gathered lists of books from certain periods, all from Project Gutenberg. I want to see if books from similar eras have similar features and if books from different eras have different characteristics.

The books that I used to gather the data for this section appear in Appendix 1.

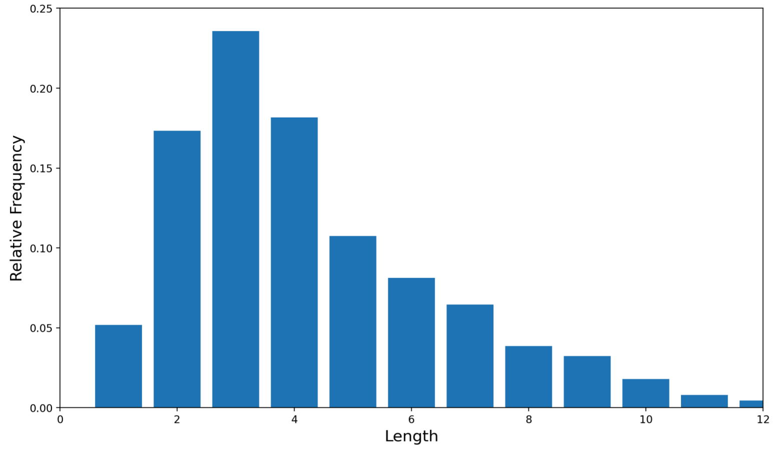
Histogram Analysis

Below are the histograms that describe the data. Overall, the histograms for the three eras are similar. They show that the most frequent word length for all periods is 3 letters. Lengths of 2 and 4 letters are also very frequent. In addition, there is a "tail" for words longer than 4 with decreasing frequency.

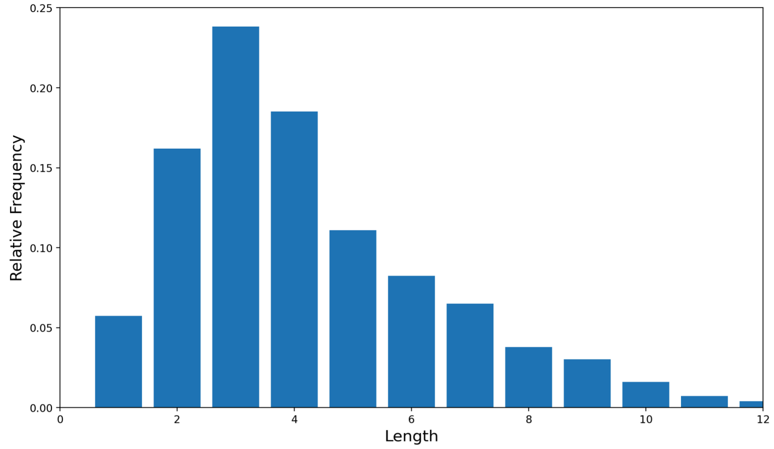
1700-1800:



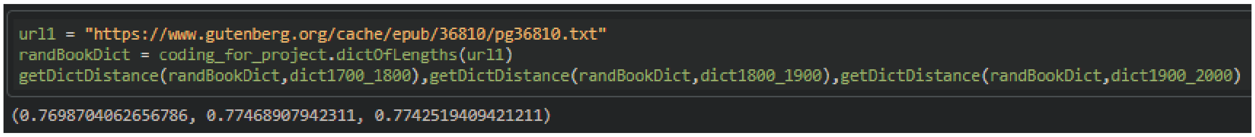
1800-1900:



1900-2000:



Next, even though the histograms are similar, I tried to see if I can identify the era of a book by the histogram of its data. I calculated the geometric distance between the book's data to each era and checked which histogram it is closest to.



```
url1 = "https://www.gutenberg.org/cache/epub/36810/pg36810.txt"
randBookDict = coding_for_project.dictOfLengths(url1)
getDictDistance(randBookDict,dict1700_1800),getDictDistance(randBookDict,dict1800_1900),getDictDistance(randBookDict,dict1900_2000)
```

```
(0.7698704062656786, 0.77468907942311, 0.7742519409421211)
```

The numbers in white in the bottom row represent the geometric distance between the histogram and each of the histograms of the eras 1700-1800, 1800-1900, 1900-2000, respectively.

As you can see the closest histogram is the one from the era 1700-1800 even though the book was written in the 1800-1900 era.

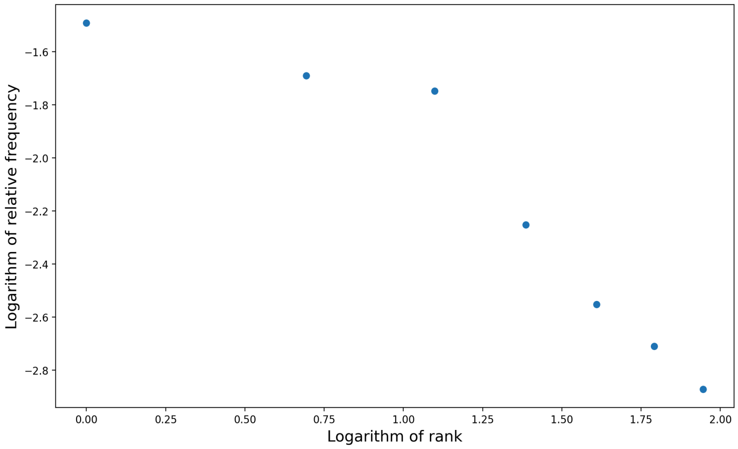
The similar histogram and this demonstration show that the distribution of word length is not a good enough indication to distinguish between the eras.

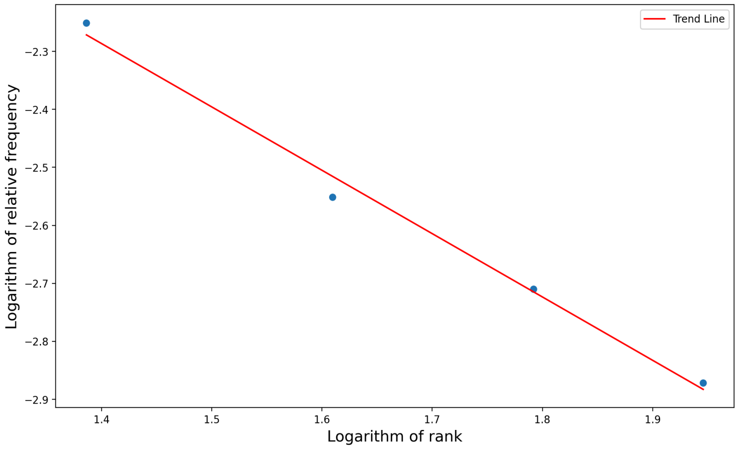
Zipf’s Law Analysis

Next I applied Zipf’s Law and used it to find observable differences between books written in different eras.

Below are graphs of the natural log of the relative frequency depending on the natural log of the frequency rank. For each era, the first graph contains the 7 most frequent word lengths. A linear relationship is most observable in ranks 3 to 7. Therefore, in order to accurately capture this relationship, I focused on ranks 3 to 7 to compute the trend line.

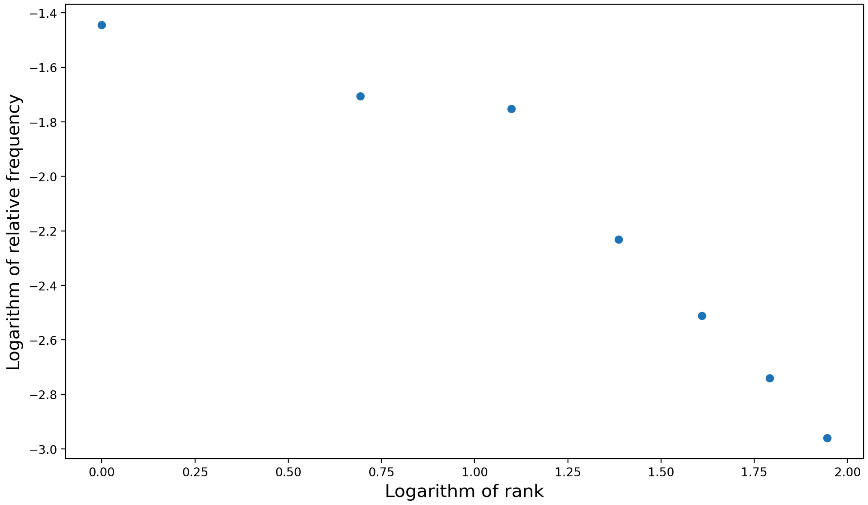
1700-1800:

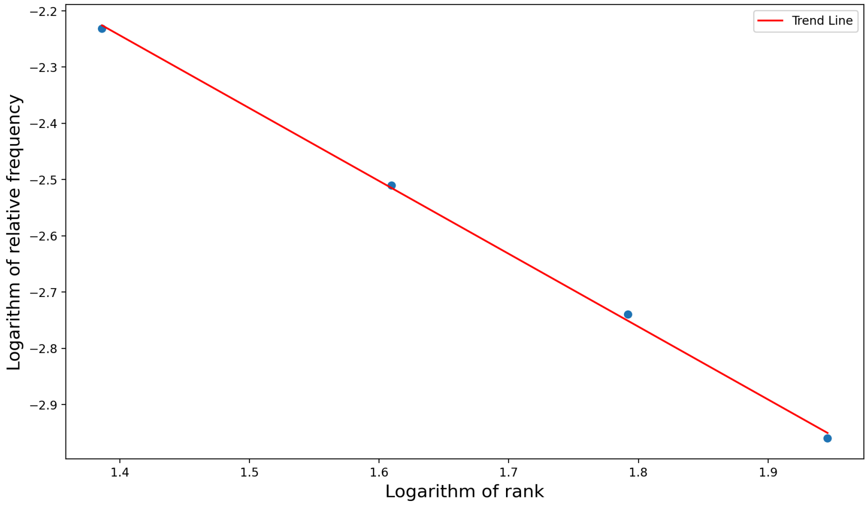




The slope of the above trend line is **-1.092.**

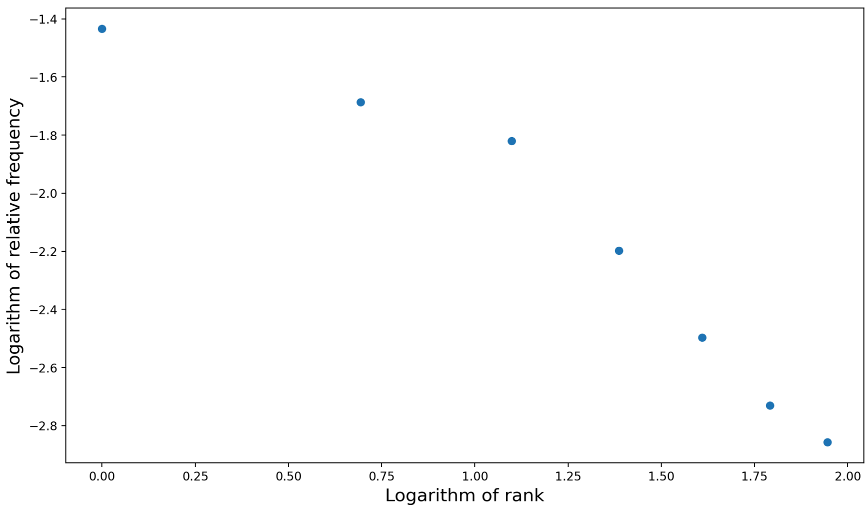
1800-1900:

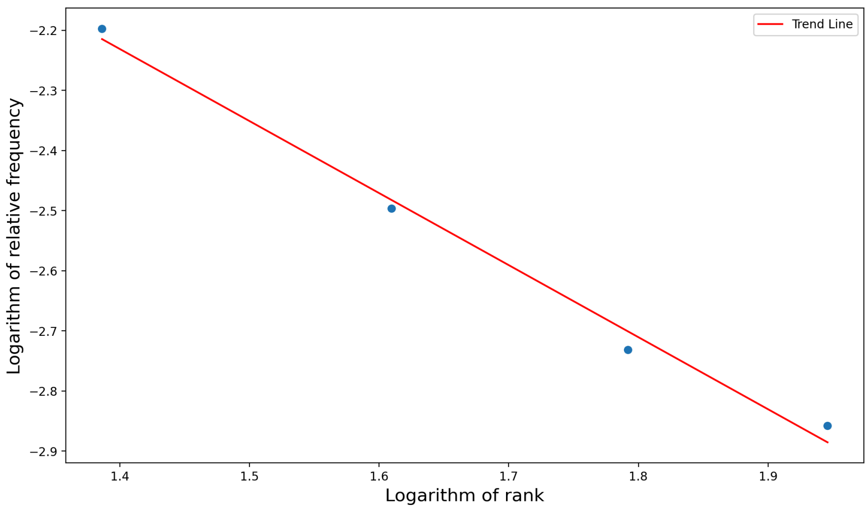




The slope of the above trend line is **-1.295**.

1900-2000:





The slope of the above trend line is **-1.199**.

Era	Gradient of trend line between ranks 4 and 7
1700-1800	**-1.092**
1800-1900	**-1.295**
1900-2000	**-1.199**

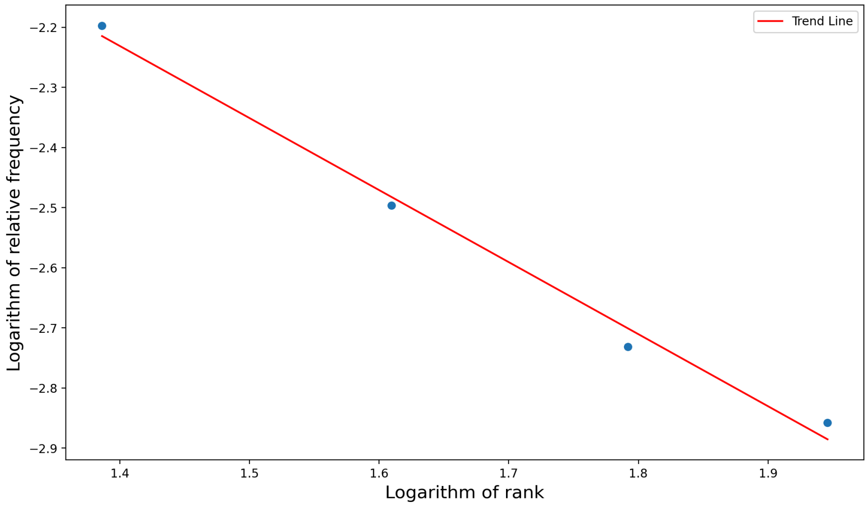
As expected all three plots show a negative slope of the trend line.

This is because the relative frequencies (y axis) are ordered by rank (x axis). However the slopes are not identical indicating different relationships between them for the three eras.

Lastly, I selected three books from these time periods and examined whether it was possible to identify the era in which the book was authored based on the slope of the book's respective graph.

Book 1:

https://gutenberg.org/cache/epub/73774/pg73774.txt - "Piracy" : A romantic chronicle of these days by Michael Arlen, written in the era 1900-2000.

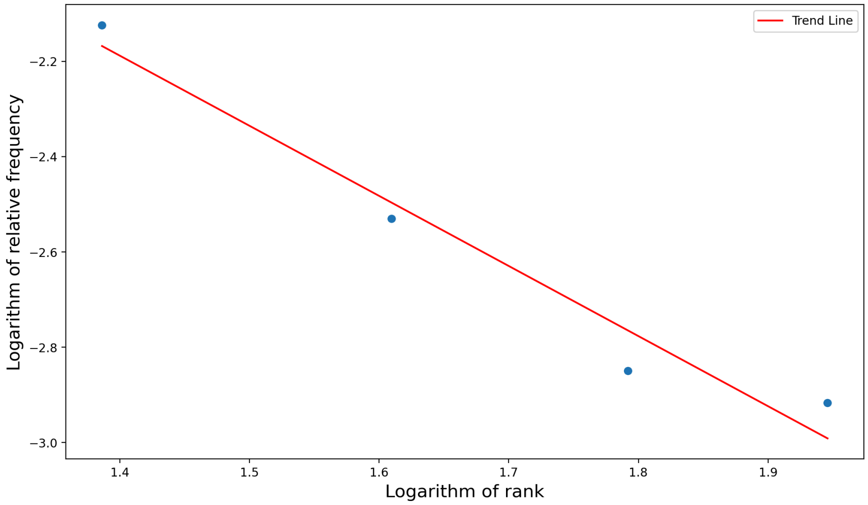


The slope of the above graph is **-1.341.**

This slope is closest to the slope of the era 1800-1900 but the book was written in 1900-2000.

Book 2:

https://gutenberg.org/cache/epub/21364/pg21364.txt - The Rajah of Dah by George Manville Fenn, written in the era 1800-1900.

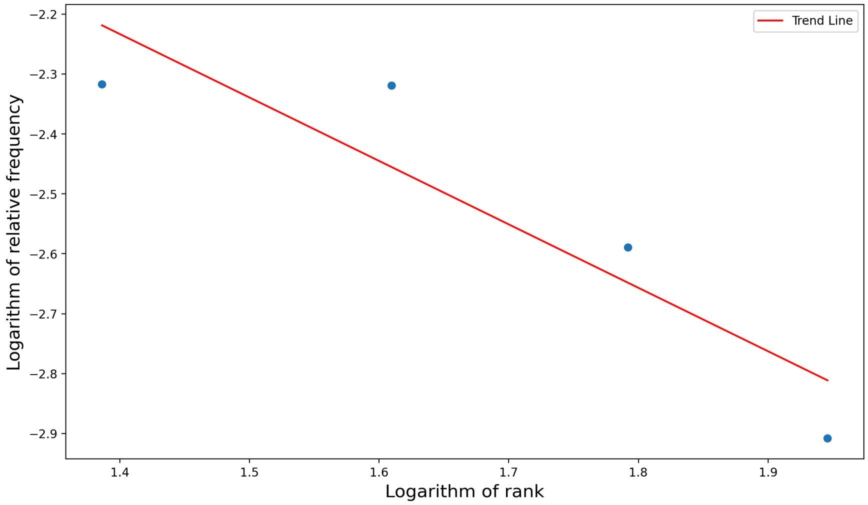


The slope of the above graph is **-1.471**.

This slope is closest to the slope of the era in which the book was written, 1800-1900. However, the book’s slope is much larger than the slope of the books from the 1800-1900 era.

Book 3:

https://gutenberg.org/cache/epub/32134/pg32134.txt - The Dictator by Stephen Marlowe, written in the era 1900-2000.



The slope of the above graph is **-1.059**.

This slope is closest to the slope of the era 1700-1800 but the book was written in 1900-2000.

Based on these findings, it appears that a collection of books from a given century cannot be reliably distinguished solely by their histograms or the slopes of graphs.

Investigating Books by Different Authors

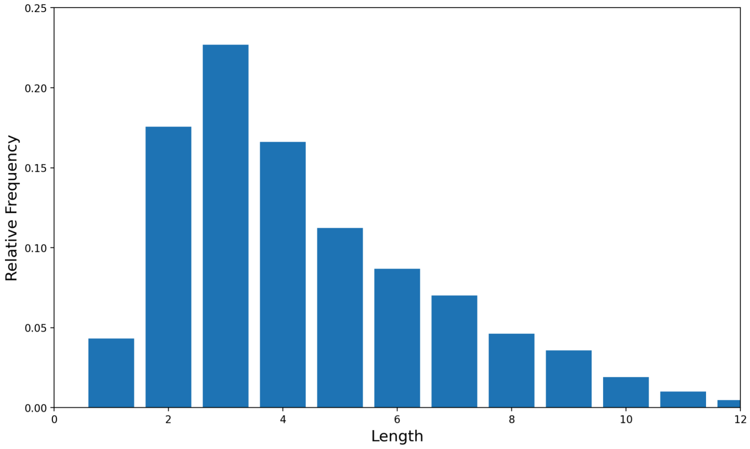
In this section, I am looking into books written by different authors to see their similarities and differences. I gathered lists of books from certain authors, all from Project Gutenberg. I want to see if books written by the same author have similar features and if books written by different authors have different characteristics.

The books that I used to gather the data for this section appear in Appendix 2.

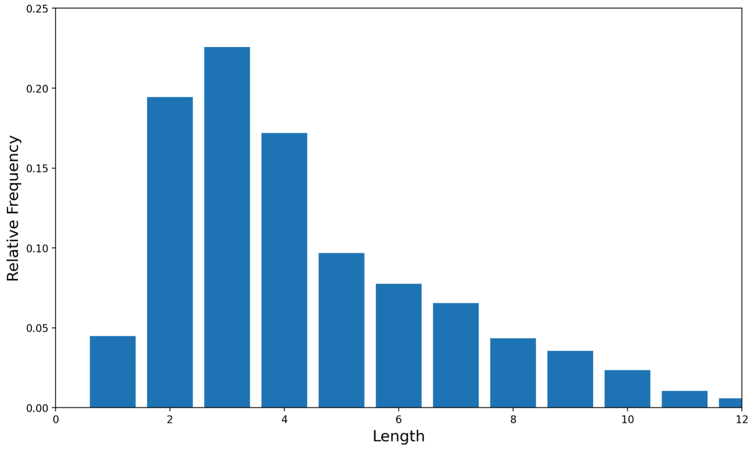
Histogram Analysis

Below are the histograms that describe the data:

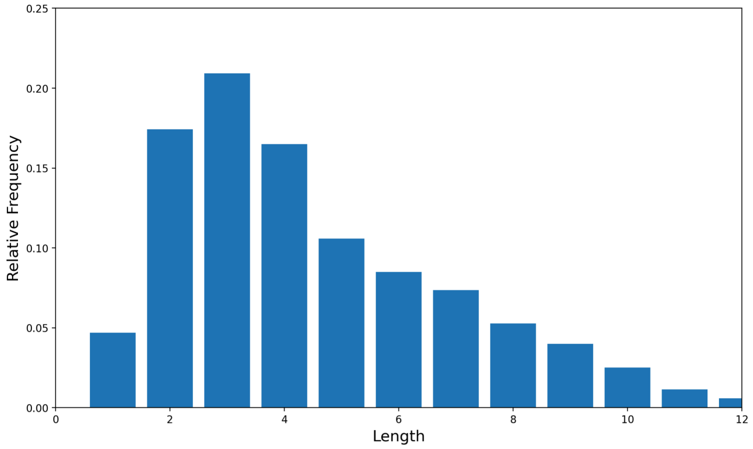
Mary Wollstonecraft Shelley:



Jane Austen:



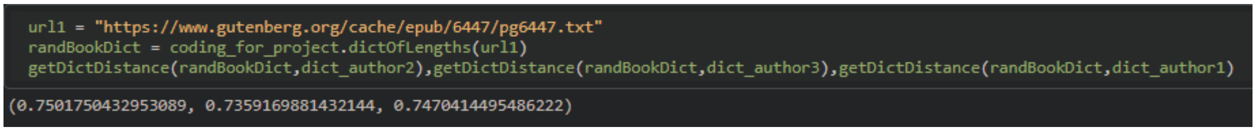
Herman Melville:



Next, even though the histograms are similar, I tried to see if I can identify the author of a book by the histogram of its data. I calculated the geometric distance between the book's data to each author's data and checked which histogram it is closest to.

The random book I chose was:

https://www.gutenberg.org/cache/epub/6447/pg6447.txt - Proserpine and Midas by Mary Wollstonecraft Shelley.



```
url1 = "https://www.gutenberg.org/cache/epub/6447/pg6447.txt"
randBookDict = coding_for_project.dictOfLengths(url1)
getDictDistance(randBookDict,dict_author2),getDictDistance(randBookDict,dict_author3),getDictDistance(randBookDict,dict_author1)

(0.7501750432953089, 0.7359169881432144, 0.7470414495486222)
```

The numbers in white in the bottom row represent the geometric distance between the histogram and each of the histograms of the authors Mary Wollstonecraft Shelley, Jane Austen, Herman Melville, respectively.

As you can see the closest histogram is the one that represents Jane Austen's data. However the book was written by Mary Wollstonecraft Shelley.

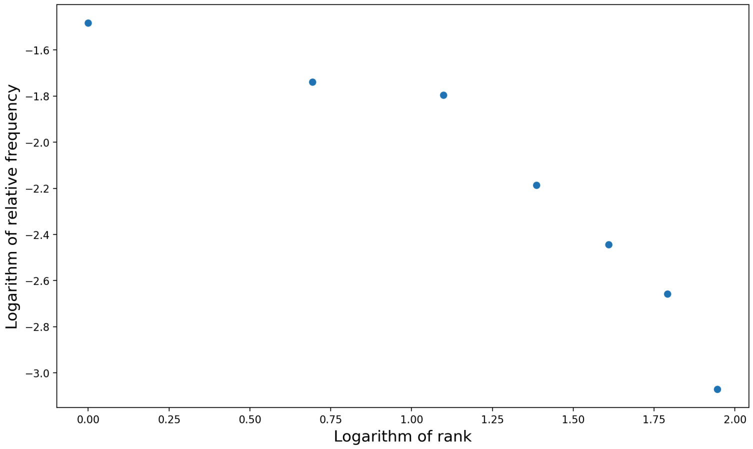
The similar histogram and this demonstration show that the distribution of word length is not a good enough indication to distinguish between the authors.

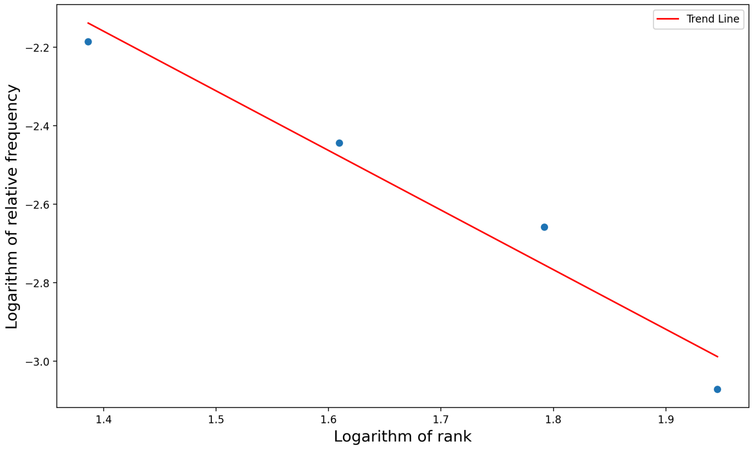
Zipf's Law Analysis

Similarly to how I did previously, I will now apply Zipf's Law and use it to find observable differences between books written in different eras.

Below are graphs of the natural log of the relative frequency depending on the natural log of the frequency rank for each author. A linear relationship is most observable in ranks 3 to 7. Therefore, in order to accurately capture this relationship, I focused on ranks 3 to 7 to compute the trend line.

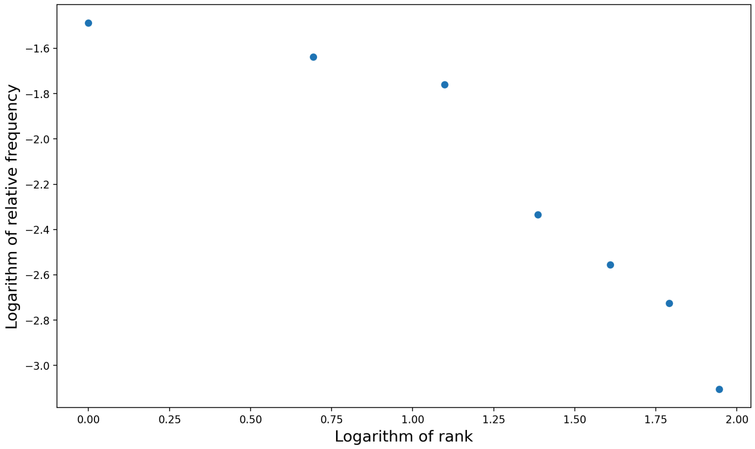
Mary Wollstonecraft Shelley:

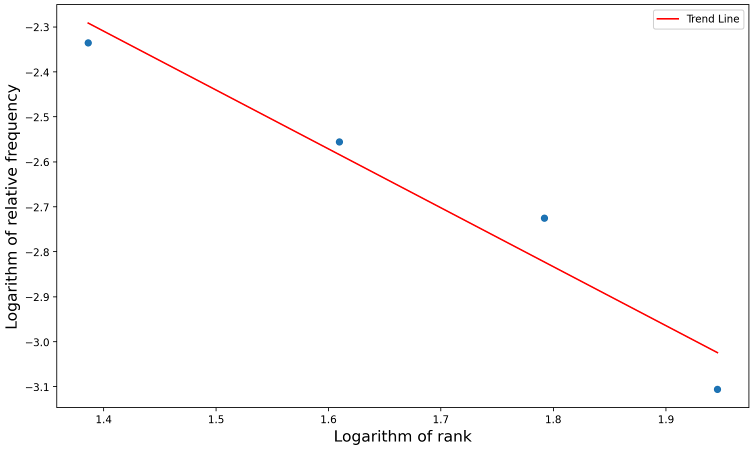




The slope of the above trend line is **-1.519**.

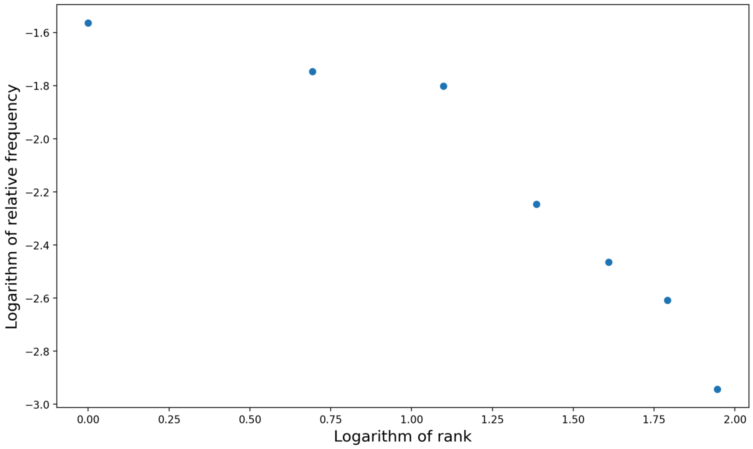
Jane Austen:

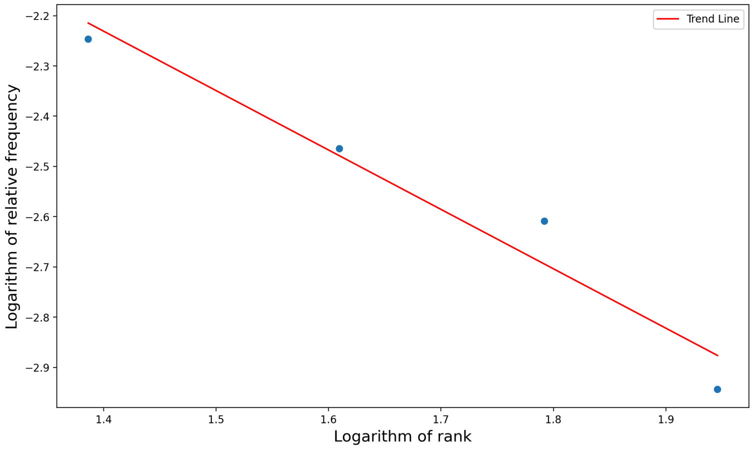




The slope of the above trend line is **-1.309**.

Herman Melville:





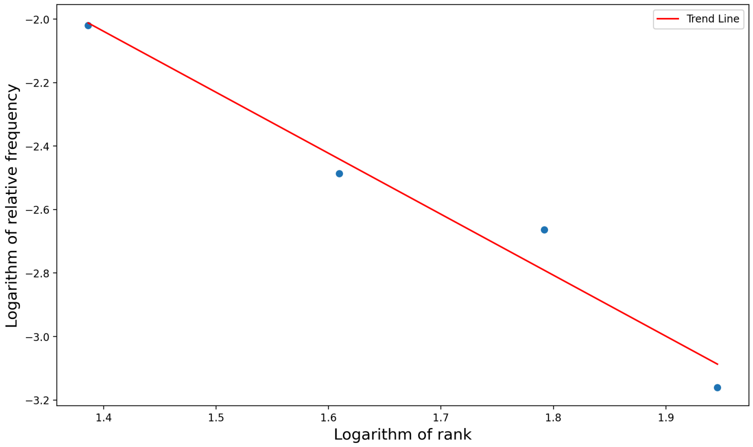
The slope of the above trend line is **-1.182**.

Author	Slope of trend line between ranks 4 and 7
Mary Wollstonecraft Shelley	**-1.519**
Jane Austen	**-1.309**
Herman Melville	**-1.182**

Lastly, I selected three books written by the above authors and examined whether it was feasible to identify the author the book was written by based on the slope of their respective graph.

Book 1:

https://www.gutenberg.org/cache/epub/6447/pg6447.txt - Prosperine and Midas by Mary Wollstonecraft Shelley.

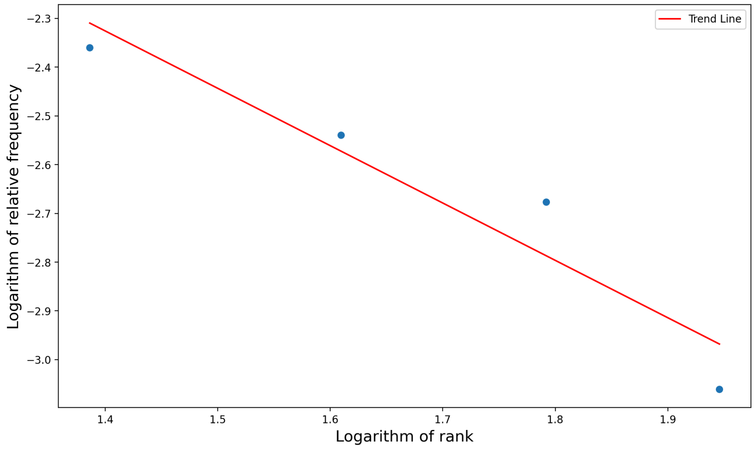


The slope of the above graph is **-1.921**.

This slope is closest to the slope of Mary Wollstonecraft Shelley's graph therefore I was able to correctly predict the author of the book.

Book 2:

https://www.gutenberg.org/cache/epub/161/pg161.txt - Sense and Sensibility by Jane Austen.

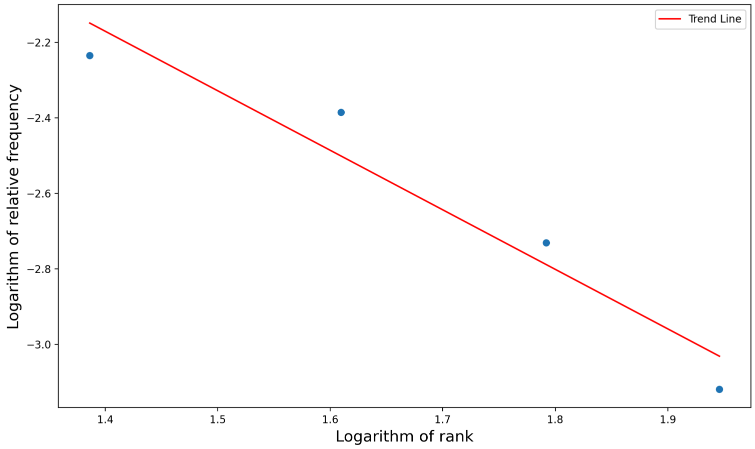


The slope of the above graph is **-1.176**.

This slope is closest to the slope of Herman Melville's graph, however the book was written by Jane Austen.

Book 3:

https://www.gutenberg.org/cache/epub/34970/pg34970.txt - Pierre; or The Ambiguities by Herman Melville.



The slope of the above graph is **-1.577**.

This slope is closest to the slope of Mary Wollstonecraft Shelley’s graph, however the book was written by Herman Melville.

Based on the above findings, it appears that a collection of books from a given author cannot be reliably distinguished solely by their histograms or the slopes of graphs.

Conclusion

In this project, I researched the link between word length and its frequency across different eras and authors. My analysis approach was based on Zipf's findings. My research reveals that there is a correlation between word length and its frequency, and that this correlation is similar across the eras and authors that I studied. Therefore, I was unable to accurately predict the specific era or author of a book based solely on the data included in the project.

This outcome highlights an important observation. While we might assume that similar authors or literary eras possess unique traits that set them apart, my findings suggest that certain characteristics, such as word length frequency, are not necessarily indicators for distinguishing between them. As a result, more advanced analysis approaches or a broader range of features may be required to make accurate predictions in future research.

What’s the next step?

This project presented my research, investigating differences between certain eras and authors. Each era was defined as a century. A possible next step is to reduce the duration of the eras and focus on significant events throughout history. An example for a specific relevant event that could have changed the characteristics of books is the invention of the printing press known as the printing revolution. In addition, other properties that may be investigated in the future include the success of a book, genre and more.

Bibliography

https://en.wikipedia.org/wiki/Linguistics

https://en.wikipedia.org/wiki/Zipf%27s_law

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4176592/

Appendix 1

These are the books that I used to gather the data for my analysis on different eras.

1700-1800:

- https://www.gutenberg.org/cache/epub/27744/pg27744.txt - Remarks on Clarissa by Sarah Fielding
- https://www.gutenberg.org/cache/epub/4085/pg4085.txt - The Adventures of Roderick Random by T. Smollett
- https://www.gutenberg.org/cache/epub/43520/pg43520.txt - The Works of Henry Fielding, vol. 11 by Henry Fielding
- https://www.gutenberg.org/cache/epub/2160/pg2160.txt - The Expedition of Humphry Clinker by T. Smollett
- https://www.gutenberg.org/cache/epub/804/pg804.txt - A Sentimental Journey Through France and Italy by Laurence Sterne
- https://www.gutenberg.org/cache/epub/1292/pg1292.txt - The Way of the World by William Congreve
- https://www.gutenberg.org/cache/epub/128/pg128.txt - The Arabian Nights Entertainments
- https://www.gutenberg.org/cache/epub/35688/pg35688.txt - Alice in Wonderland by Lewis Carroll and Alice Gerstenberg
- https://www.gutenberg.org/cache/epub/9182/pg9182.txt - Villette by Charlotte Brontë
- https://www.gutenberg.org/cache/epub/1022/pg1022.txt - Walking by Henry David Thoreau

1800-1900:

- https://www.gutenberg.org/cache/epub/10724/pg10724.txt - The Store Boy by Jr. Horatio Alger
- https://www.gutenberg.org/cache/epub/84/pg84.txt - Frankenstein; Or, The Modern Prometheus by Mary Wollstonecraft Shelley
- https://www.gutenberg.org/cache/epub/2701/pg2701.txt - Moby Dick; Or, The Whale by Herman Melville
- https://www.gutenberg.org/cache/epub/21223/pg21223.txt - The Carbonels by Charlotte M. Yonge
- https://www.gutenberg.org/cache/epub/30970/pg30970.txt - Miss Cayley's Adventures by Grant Allen
- https://www.gutenberg.org/cache/epub/23853/pg23853.txt - Ran Away to Sea by Mayne Reid
- https://www.gutenberg.org/cache/epub/120/pg120.txt - Treasure Island by Robert Louis Stevenson
- https://www.gutenberg.org/cache/epub/730/pg730.txt - Oliver Twist by Charles Dickens
- https://www.gutenberg.org/cache/epub/902/pg902.txt - The Happy Prince, and Other Tales by Oscar Wilde
- https://www.gutenberg.org/cache/epub/1064/pg1064.txt - The Masque of the Red Death by Edgar Allan Poe

1900-2000:

- https://www.gutenberg.org/cache/epub/42796/pg42796.txt - The Box-Car Children by Gertrude Chandler Warner
- https://www.gutenberg.org/cache/epub/67979/pg67979.txt - The Blue Castle: a novel by L. M. Montgomery
- https://www.gutenberg.org/cache/epub/64317/pg64317.txt - The Great Gatsby by F. Scott Fitzgerald
- https://www.gutenberg.org/cache/epub/14907/pg14907.txt - Living Alone by Stella Benson
- https://www.gutenberg.org/cache/epub/71152/pg71152.txt - A lady and her husband by Amber Reeves Blanco White
- https://www.gutenberg.org/cache/epub/72824/pg72824.txt - The mystery of the Blue Train by Agatha Christie
- https://www.gutenberg.org/cache/epub/4300/pg4300.txt - Ulysses by James Joyce
- https://www.gutenberg.org/cache/epub/72869/pg72869.txt - Meet the Tiger by Leslie Charteris
- https://www.gutenberg.org/cache/epub/32032/pg32032.txt - Second Variety by Philip K. Dick
- https://www.gutenberg.org/cache/epub/58866/pg58866.txt - The Murder on the Links by Agatha Christie

Appendix 2

These are the books that I used to gather the data for my analysis on different authors.

Mary Wollstonecraft Shelley:

- https://www.gutenberg.org/cache/epub/18247/pg18247.txt - The Last Man
- https://www.gutenberg.org/cache/epub/56665/pg56665.txt - Tales and Stories
- https://www.gutenberg.org/cache/epub/15238/pg15238.txt - Mathilda

Jane Austen:

- https://www.gutenberg.org/cache/epub/141/pg141.txt - Mansfield Park
- https://www.gutenberg.org/cache/epub/946/pg946.txt - Lady Susan
- https://www.gutenberg.org/cache/epub/21839/pg21839.txt - Sense and Sensibility

Herman Melville:

- https://www.gutenberg.org/cache/epub/11231/pg11231.txt - Bartleby, the Scrivener: A Story of Wall-Street
- https://www.gutenberg.org/cache/epub/15859/pg15859.txt - The Piazza Tales
- https://www.gutenberg.org/cache/epub/1900/pg1900.txt - Typee: A Romance of the South Seas