# Week8

March 23, 2024

#### 0.0.1 Week 8 - High Frequency Words

#### 0.0.2 Shamecca Marshall

#### 0.0.3 Project Overview

Please answer the following questions in an IPython Notebook, posted to GitHub.

1. Choose a corpus of interest.
2. How many total unique words are in the corpus? (Please feel free to define unique words in any interesting, defensible way).
3. Taking the most common words, how many unique words represent half of the total words in the corpus?
4. Identify the 200 highest frequency words in this corpus.
5. Create a graph that shows the relative frequency of these 200 words.
6. Does the observed relative frequency of these words follow Zipf's law? Explain.
7. In what ways do you think the frequency of the words in this corpus differ from "all words in all corpora."

#### 0.0.4 1. Choosing a corpus of interest.

I selected one of the corpus from the freely available Gutenburg library that can be downloaded from the NLTK package. Our corpus is Emma written by Jane Austen.

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import nltk
nltk.download('gutenberg')
```

```
[nltk_data] Downloading package gutenberg to /Users/MECCA/nltk_data…
[nltk_data]   Unzipping corpora/gutenberg.zip.
```

[1]:
```
True
```

[2]:
```
nltk.corpus.gutenberg.fileids()
```

[2]:
```
['austen-emma.txt',
 'austen-persuasion.txt',
 'austen-sense.txt',
 'bible-kjv.txt',
```

```
 'blake-poems.txt',
 'bryant-stories.txt',
 'burgess-busterbrown.txt',
 'carroll-alice.txt',
 'chesterton-ball.txt',
 'chesterton-brown.txt',
 'chesterton-thursday.txt',
 'edgeworth-parents.txt',
 'melville-moby_dick.txt',
 'milton-paradise.txt',
 'shakespeare-caesar.txt',
 'shakespeare-hamlet.txt',
 'shakespeare-macbeth.txt',
 'whitman-leaves.txt']
```

[4]:
```
austen = nltk.Text(nltk.corpus.gutenberg.words('austen-emma.txt'))
```

### 0.0.5 2. Total Unique Words.

How many total unique words are in the corpus?

I can count all 'words' by taking the lenth of our corpus…

[6]:
```
AW=len(austen)
AW
```

[6]:
```
192427
```

However, this tally includes every word, even duplicates. We can eliminate duplicates by encapsulating our corpus within the Python set function. Let's proceed with that approach and then arrange the outcomes in ascending order to examine the results.

[7]:
```
print(*sorted(set(austen))[:100], sep = "   ")
```

```
!   !"   !"--   !'   !'--   !)--   !--   !--"   !--(   !--`   "   "'   "--   "`
&   '   '--   ';   (   )   ),   )--   ).   ).--   );--   ,   ,"   ,"--   ,'
,'"   ,)   ,--   ,--"   -   --   --"   --(   --,   ----   ----------,
--------.'   --.   --."   --.'   --:   --`   .   ."   ."--   .'   .'"   .'--
.'--`   .)   .,   .,"   .,'   .--   .--"   .--`   .]   000   10   1816   23rd
24th   26th   28th   7th   8th   :   :"   :"--   :'   :'--   :--   :--"   ;   ;"
;"--   ;'   ;'--   ;--   ;--"   ?   ?"   ?"--   ?"--"   ?'   ?'"   ?)--   ?--
?--"   ?--(   A   Abbey   Abbots   Abdy   Abominable   About
```

The initial nearly 100 entries comprise non-word elements.

'Unique' Words Upon reviewing the provided word sample, it becomes apparent that it encompasses punctuation marks, numbers, and words. Additionally, considering that Python distinguishes between uppercase and lowercase letters, it becomes imperative to convert all words to lowercase and eliminate punctuation and numbers to obtain solely the unique words.

We have adapted code from the textbook to exclude punctuation and numbers while also converting all letters to lowercase.

[8]:
```
# create list of all words including duplicates, but excluding punctuation,␣
↪numbers and capitalization
AWwoNP = [word.lower() for word in austen if word.isalpha()]

# print
print(*AWwoNP[:100], sep = "   ")
```

```
emma   by   jane   austen   volume   i   chapter   i   emma   woodhouse
handsome   clever   and   rich   with   a   comfortable   home   and   happy
disposition   seemed   to   unite   some   of   the   best   blessings   of
existence   and   had   lived   nearly   twenty   one   years   in   the   world
with   very   little   to   distress   or   vex   her   she   was   the
youngest   of   the   two   daughters   of   a   most   affectionate   indulgent
father   and   had   in   consequence   of   her   sister   s   marriage   been
mistress   of   his   house   from   a   very   early   period   her   mother
had   died   too   long   ago   for   her   to   have   more   than   an
indistinct   remembrance   of   her
```

[9]:
```
# take the length of the set of those words to find the number of unique words
len(set(AWwoNP))
```

[9]: 7079

By taking the length of the set of words after we removed punctuation, numbers and capitalization we find that we have 7,079 'unique' words in our corpus.

### 0.0.6 3. How many unique words represent half of all words.

How many distinct words, derived from the most frequently occurring ones, are required to account for half of the total words in the corpus? To achieve this, we generate a frequency distribution of all words, excluding numerals, punctuation, and capitalization. Subsequently, we devise a function to iteratively accumulate the frequency counts of the most prevalent words, ordered from highest to lowest frequency, until their cumulative count reaches or exceeds half of the total word count in the corpus. Simultaneously, we keep track of the number of word frequencies combined to ascertain the quantity of words representing half of the total words.

[10]:
```
fdist = nltk.FreqDist(AWwoNP)
fdist
```

[10]:
```
FreqDist({'to': 5239, 'the': 5201, 'and': 4896, 'of': 4291, 'i': 3178, 'a':
3129, 'it': 2528, 'her': 2469, 'was': 2398, 'she': 2340, …})
```

[11]:
```
tw=len(AWwoNP)
tcount=0
wcount=0
```

```
for word, count in fdist.most_common():
    tcount=tcount+count
    wcount=wcount+1
    if tcount>(tw/2):
        print(wcount)
        break
```

56

The following 56 unique words represent half of the total words in the corpus.

[13]:
```
print(*[w for w,n in fdist.most_common()[:56]], sep = ", ")
```

```
to, the, and, of, i, a, it, her, was, she, in, not, you, be, that, he, had, but,
as, for, have, is, with, very, mr, his, at, so, s, emma, all, could, would,
been, him, no, my, mrs, on, any, do, were, miss, me, by, will, must, which,
there, from, they, what, this, harriet, or, such
```

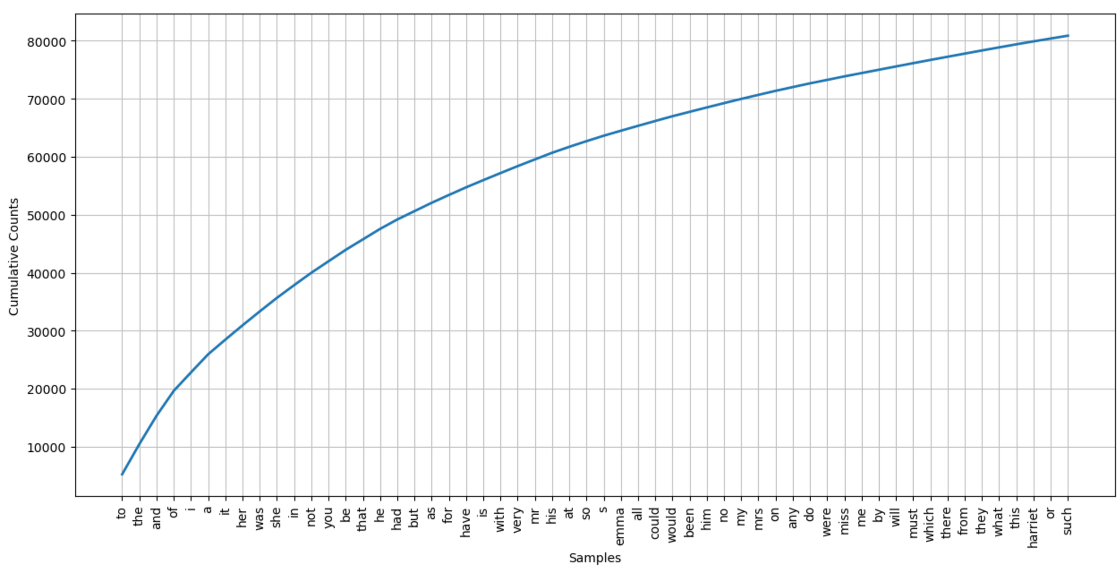
We can double check this by plotting a cumulative frequency plot for the first 56 words and comparing the cumulative count to half the total word count tw.

[15]:
```
tw/2
```

[15]: 80800.0

The cumulative word count for the first 56 words in the plot below matches pretty well to the expected value of 80,800.

[16]:
```
plt.figure(figsize=(15,7))
fdist.plot(56, cumulative = True)
```



[16]:
```
<Axes: xlabel='Samples', ylabel='Cumulative Counts'>
```

**4. The 200 most frequent words.** Identify the 200 highest frequency words in this corpus

[17]:
```python
wlist = []
for i in range(0, 200, 25):
    df = pd.DataFrame(fdist.most_common()[i:(i+25)])
    df.columns=['word', 'count']
    wlist.append(df)

pd.concat(wlist, axis=1)
```

[17]:

| | word | count | word | count | word | count | word | count | word \ |
|---|---|---|---|---|---|---|---|---|---|
| 0 | to | 5239 | his | 1145 | they | 540 | your | 364 | too |
| 1 | the | 5201 | at | 1031 | what | 536 | when | 363 | before |
| 2 | and | 4896 | so | 974 | this | 526 | little | 359 | has |
| 3 | of | 4291 | s | 935 | harriet | 506 | being | 358 | about |
| 4 | i | 3178 | emma | 865 | or | 494 | never | 358 | most |
| 5 | a | 3129 | all | 845 | such | 489 | good | 358 | dear |
| 6 | it | 2528 | could | 837 | much | 486 | did | 352 | fairfax |
| 7 | her | 2469 | would | 820 | if | 485 | we | 349 | always |
| 8 | was | 2398 | been | 759 | said | 484 | only | 341 | man |
| 9 | she | 2340 | him | 759 | more | 467 | know | 337 | thought |
| 10 | in | 2188 | no | 742 | an | 464 | might | 326 | soon |
| 11 | not | 2140 | my | 728 | are | 455 | woodhouse | 313 | churchill |
| 12 | you | 1980 | mrs | 699 | one | 452 | say | 310 | see |
| 13 | be | 1975 | on | 692 | weston | 440 | now | 309 | other |
| 14 | that | 1806 | any | 654 | every | 435 | their | 306 | may |
| 15 | he | 1806 | do | 640 | them | 432 | jane | 301 | again |
| 16 | had | 1624 | were | 600 | am | 425 | own | 301 | shall |
| 17 | but | 1441 | miss | 599 | than | 415 | who | 294 | without |
| 18 | as | 1436 | me | 573 | well | 401 | can | 284 | out |
| 19 | for | 1347 | by | 571 | thing | 398 | quite | 282 | first |
| 20 | have | 1320 | will | 570 | knightley | 389 | herself | 279 | frank |
| 21 | is | 1240 | must | 567 | elton | 385 | time | 279 | father |
| 22 | with | 1217 | which | 556 | think | 383 | great | 264 | sure |
| 23 | very | 1202 | there | 549 | how | 371 | some | 262 | indeed |
| 24 | mr | 1153 | from | 546 | should | 369 | nothing | 256 | like |

| | count | word | count | word | count | word | count |
|---|---|---|---|---|---|---|---|
| 0 | 254 | made | 199 | long | 146 | its | 122 |
| 1 | 250 | body | 193 | rather | 146 | look | 121 |
| 2 | 250 | ever | 193 | himself | 146 | going | 120 |
| 3 | 249 | oh | 193 | us | 145 | heard | 120 |
| 4 | 248 | day | 192 | hope | 143 | moment | 120 |
| 5 | 241 | young | 192 | done | 142 | came | 119 |
| 6 | 241 | up | 190 | cannot | 142 | last | 119 |

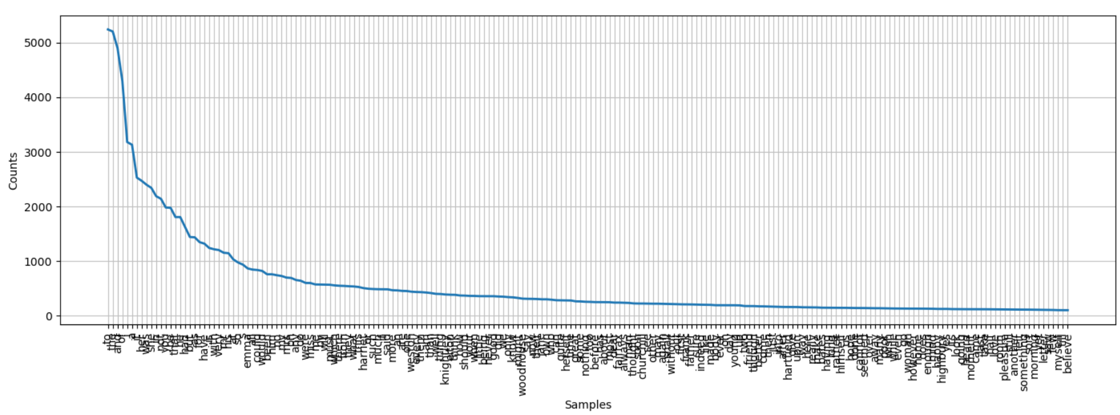
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 7 | 238 | two | 178 | seemed | 141 | take | 119 |
| 8 | 235 | friend | 177 | over | 139 | half | 118 |
| 9 | 226 | though | 177 | away | 138 | love | 117 |
| 10 | 224 | better | 173 | many | 138 | room | 117 |
| 11 | 224 | come | 172 | poor | 136 | pleasure | 115 |
| 12 | 222 | then | 169 | wish | 135 | still | 115 |
| 13 | 221 | just | 165 | while | 133 | another | 114 |
| 14 | 221 | into | 163 | even | 132 | felt | 113 |
| 15 | 219 | after | 161 | go | 132 | something | 113 |
| 16 | 217 | hartfield | 160 | woman | 131 | sort | 112 |
| 17 | 214 | give | 159 | however | 131 | morning | 111 |
| 18 | 212 | upon | 159 | home | 130 | yet | 109 |
| 19 | 209 | way | 155 | does | 130 | letter | 109 |
| 20 | 208 | here | 154 | enough | 129 | saw | 108 |
| 21 | 207 | really | 153 | mind | 128 | few | 106 |
| 22 | 204 | make | 152 | happy | 125 | myself | 103 |
| 23 | 202 | bates | 148 | highbury | 125 | till | 102 |
| 24 | 200 | having | 147 | yes | 125 | believe | 102 |

### 0.0.7 5. Relative frequency of these 200 words.

Create a graph that shows the relative frequency of these 200 words.

Unfortunately, the default NLTK plot for frequency distributions does not actually plot the relative frequency but plots the actual counts…

[18]:
```
plt.figure(figsize=(17,5))
fdist.plot(200, )
```



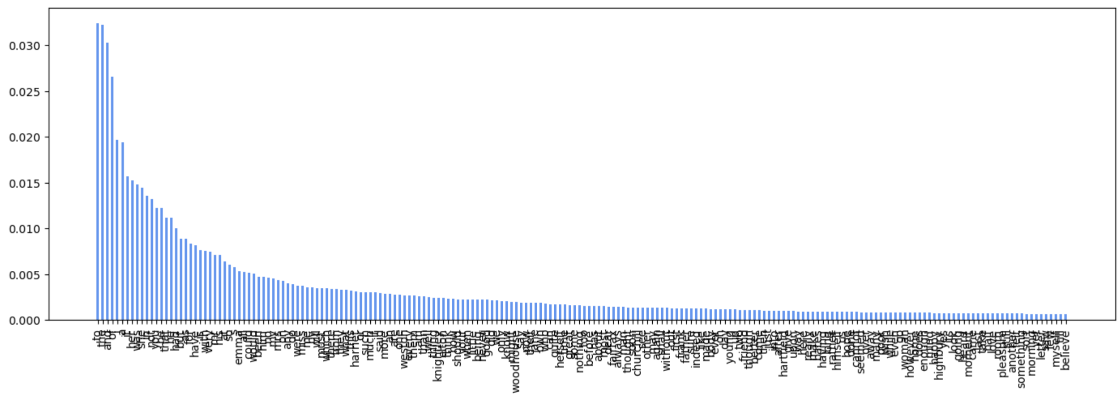

[18]:
```
<Axes: xlabel='Samples', ylabel='Counts'>
```

[ ]:
```
```

my_dict = {} wcount=0 for word, count in fdist.most_common(): wcount=wcount+1 my_dict[word]=count/tw if wcount>199: break

[20]:
```python
plt.figure(figsize=(17,5))
plt.xticks(rotation=90)
plt.rc('xtick',labelsize=4)
plt.bar(my_dict.keys(), my_dict.values(), width=.5, color='cornflowerblue')
```

[20]:
```
<BarContainer object of 200 artists>
```




### 0.0.8 6. Zipf's law

Does the observed relative frequency of these words follow Zipf's law? Explain.

Zipf's Law is a statistical distribution in certain data sets, such as words in a linguistic corpus, in which the frequencies of certain words are inversely proportional to their ranks. Named for linguist George Kingsley Zipf, who around 1935 was the first to draw attention to this phenomenon, the law examines the frequency of words in natural language and how the most common word occurs twice as often as the second most frequent word, three times as often as the subsequent word and so on until the least frequent word. The word in the position n appears 1/n times as often as the most frequent one.

https://www.techtarget.com/whatis/definition/Zipfs-Law

We'll examine whether the observed counts of each word match the anticipated counts as per Zipf's Law. Initially, we'll create a function to compute the expected counts. Subsequently, we'll merge this data into a dataframe alongside the actual counts, and calculate the variance and percentage variance for each word.

[23]:
```python
mostf = fdist.most_common()[0][1]
expected_counts = []
rank = 0
for i in range(len(fdist)):
    rank += 1
    expected_counts.append(round(mostf * (1/rank)))
```

```
expected_counts

zipfs_df = pd.DataFrame(fdist.most_common())
zipfs_df.columns=['Word', 'Actual count']
#pd.concat(zipfs_df, axis=1)
zipfs_df["Zipf's Expected Count"] = expected_counts
zipfs_df['Difference'] = zipfs_df['Actual count'] - zipfs_df["Zipf's Expected␣
↪Count"]
zipfs_df['Percent Difference'] = round(((zipfs_df['Actual count'] /␣
↪zipfs_df["Zipf's Expected Count"])
                                        - 1) *100).astype(int)
zipfs_df.head(30)
```

[23]:

| | Word | Actual count | Zipf's Expected Count | Difference | Percent Difference |
|---|---|---|---|---|---|
| 0 | to | 5239 | 5239 | 0 | 0 |
| 1 | the | 5201 | 2620 | 2581 | 99 |
| 2 | and | 4896 | 1746 | 3150 | 180 |
| 3 | of | 4291 | 1310 | 2981 | 228 |
| 4 | i | 3178 | 1048 | 2130 | 203 |
| 5 | a | 3129 | 873 | 2256 | 258 |
| 6 | it | 2528 | 748 | 1780 | 238 |
| 7 | her | 2469 | 655 | 1814 | 277 |
| 8 | was | 2398 | 582 | 1816 | 312 |
| 9 | she | 2340 | 524 | 1816 | 347 |
| 10 | in | 2188 | 476 | 1712 | 360 |
| 11 | not | 2140 | 437 | 1703 | 390 |
| 12 | you | 1980 | 403 | 1577 | 391 |
| 13 | be | 1975 | 374 | 1601 | 428 |
| 14 | that | 1806 | 349 | 1457 | 417 |
| 15 | he | 1806 | 327 | 1479 | 452 |
| 16 | had | 1624 | 308 | 1316 | 427 |
| 17 | but | 1441 | 291 | 1150 | 395 |
| 18 | as | 1436 | 276 | 1160 | 420 |
| 19 | for | 1347 | 262 | 1085 | 414 |
| 20 | have | 1320 | 249 | 1071 | 430 |
| 21 | is | 1240 | 238 | 1002 | 421 |
| 22 | with | 1217 | 228 | 989 | 434 |
| 23 | very | 1202 | 218 | 984 | 451 |
| 24 | mr | 1153 | 210 | 943 | 449 |
| 25 | his | 1145 | 202 | 943 | 467 |
| 26 | at | 1031 | 194 | 837 | 431 |
| 27 | so | 974 | 187 | 787 | 421 |
| 28 | s | 935 | 181 | 754 | 417 |
| 29 | emma | 865 | 175 | 690 | 394 |

We can see in the data above that the actual counts are consistenetly far above the expected counts and by the 13th word the actual counts rise to about 428% above the expected count and remain consistently there for at least the first 30 words. Let's jump to the 500th word to see if the counts are still consistently higher.

At around 500 words into the frequency distribution the actual counts are still around 270% higher than would be expected with Zipf's Law.

[24]:
```
zipfs_df[505:515]
```

[24]:
```
          Word  Actual count  Zipf's Expected Count  Difference  \
505      added            37                     10          27
506     temper            36                     10          26
507  necessary            36                     10          26
508        got            36                     10          26
509   received            36                     10          26
510   enscombe            36                     10          26
511    forward            36                     10          26
512        new            36                     10          26
513 determined            36                     10          26
514    meaning            36                     10          26

     Percent Difference
505                 270
506                 260
507                 260
508                 260
509                 260
510                 260
511                 260
512                 260
513                 260
514                 260
```

At about half way through the frequency distribution the counts are still about 100% higher than would be expected.

[25]:
```
zipfs_df[2500:2505]
```

[25]:
```
            Word  Actual count  Zipf's Expected Count  Difference  \
2500  collecting             4                      2           2
2501    charades             4                      2           2
2502    intently             4                      2           2
2503    security             4                      2           2
2504   addressed             4                      2           2

      Percent Difference
2500                 100
2501                 100
```

```
2502                    100
2503                    100
2504                    100
```

Not until the tail of the distribution do the actual and expected counts match.

[26]:
```
zipfs_df.tail(5)
```

[26]:
```
              Word  Actual count  Zipf's Expected Count  Difference  \
7074         stare             1                      1           0
7075  deficiencies             1                      1           0
7076   predictions             1                      1           0
7077          band             1                      1           0
7078         finis             1                      1           0

      Percent Difference
7074                   0
7075                   0
7076                   0
7077                   0
7078                   0
```

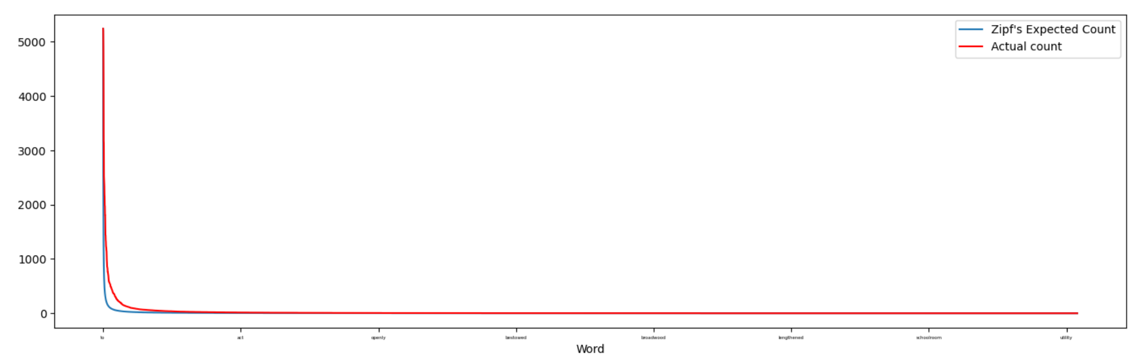
Let's plot the expected counts and actual counts to visualize the entire distribution.

[28]:
```
plt.figure(figsize=(17,5))

# gca stands for 'get current axis'
ax = plt.gca()

zipfs_df.plot(kind='line',y="Zipf's Expected Count",x='Word',ax=ax)
zipfs_df.plot(kind='line',y='Actual count',x='Word', color='red', ax=ax)

plt.show()
```



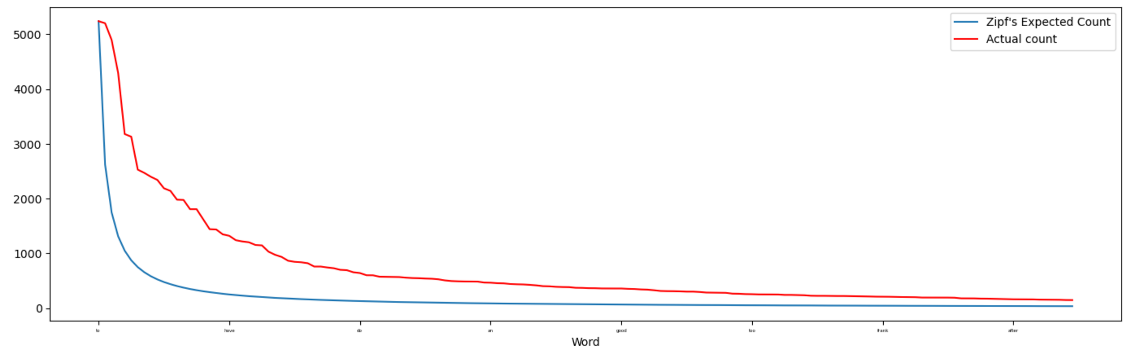

Examining the plot above, it appears that the relative frequency observed for these words broadly adheres to Zipf's law, as indicated by the similarity in the plot shapes. However, upon closer inspection of the first 150 words, as depicted in the second plot below, it becomes evident that there are notable discrepancies in the relative frequencies.

[29]:
```
plt.figure(figsize=(17,5))

# gca stands for 'get current axis'
ax = plt.gca()

zipfs_df[:150].plot(kind='line',y="Zipf's Expected Count",x='Word',ax=ax)
zipfs_df[:150].plot(kind='line',y='Actual count',x='Word', color='red', ax=ax)

plt.show()
```




### 0.0.9 7. Compare with 'all words in all corpora'

How do you perceive the disparities in word frequency within this corpus compared to "all words in all corpora"? For our reference point of "all words in all corpora," we'll utilize Wikipedia's compilation of the 100 most commonly used English words. We'll import the list containing the first 100 most frequently employed English words from Wikipedia.

[31]:
```
mc_english_words = pd.read_html('https://en.wikipedia.org/wiki/
↪Most_common_words_in_English#100_most_common_words',
                  header=0, index_col=None)
mc_english_words = mc_english_words[0]
mc_english_words['Word'] = mc_english_words['Word'].str.lower()
mc_english_words
```

[31]:
```
     Word Parts of speech OEC rank COCA rank[9]           Dolch level  \
0     the         Article        1            1            Pre-primer
1      be            Verb        2            2                Primer
2      to     Preposition        3         7, 9            Pre-primer
3      of     Preposition        4            4               Grade 1
4     and     Coordinator        5            3            Pre-primer
..    …             …       …          …                        …
```

```
95   these      Pronoun     96        82              Grade 2
96    give         Verb     97        98              Grade 1
97     day         Noun     98        90  Dolch list of 95 nouns
98    most       Adverb     99  144, 187                  NaN
99      us      Pronoun    100       113              Grade 2

    Polysemy
0         12
1         21
2         17
3         12
4         16
..       …
95         2
96        19
97         9
98        12
99         6

[100 rows x 6 columns]
```

Subsequently, we can construct a dataframe comprising solely the first 100 most commonly utilized words in the Jane Austen text, facilitating a comparison between the two lists. The Wikipedia list is already furnished with a rank in the OEC rank column, and we can leverage the index incremented by one to generate a rank column for the Jane Austen data.

[32]:
```
austen_100 = pd.DataFrame(fdist.most_common()[:100], columns=['Austen_Word',␣
↪'Austen_Frequency'])
austen_100['Austen_Rank'] = austen_100.index +1
austen_100
```

[32]:
```
   Austen_Word  Austen_Frequency  Austen_Rank
0           to              5239            1
1          the              5201            2
2          and              4896            3
3           of              4291            4
4            i              3178            5
..         …                 …          …
95     herself               279           96
96        time               279           97
97       great               264           98
98        some               262           99
99     nothing               256          100

[100 rows x 3 columns]
```

[ ]:

[33]:
```
pd.set_option('display.max_rows', 500)
words_merged = pd.
 ↪merge(austen_100,mc_english_words,left_on='Austen_Word',right_on='Word',how='outer')
top_100 = words_merged.fillna('')
top_100
```

[33]:

| | Austen_Word | Austen_Frequency | Austen_Rank | Word | \ |
|---|---|---|---|---|---|
| 0 | to | 5239.0 | 1.0 | to | |
| 1 | the | 5201.0 | 2.0 | the | |
| 2 | and | 4896.0 | 3.0 | and | |
| 3 | of | 4291.0 | 4.0 | of | |
| 4 | i | 3178.0 | 5.0 | i | |
| 5 | a | 3129.0 | 6.0 | a | |
| 6 | it | 2528.0 | 7.0 | it | |
| 7 | her | 2469.0 | 8.0 | her | |
| 8 | was | 2398.0 | 9.0 | | |
| 9 | she | 2340.0 | 10.0 | she | |
| 10 | in | 2188.0 | 11.0 | in | |
| 11 | not | 2140.0 | 12.0 | not | |
| 12 | you | 1980.0 | 13.0 | you | |
| 13 | be | 1975.0 | 14.0 | be | |
| 14 | that | 1806.0 | 15.0 | that | |
| 15 | he | 1806.0 | 16.0 | he | |
| 16 | had | 1624.0 | 17.0 | | |
| 17 | but | 1441.0 | 18.0 | but | |
| 18 | as | 1436.0 | 19.0 | as | |
| 19 | for | 1347.0 | 20.0 | for | |
| 20 | have | 1320.0 | 21.0 | have | |
| 21 | is | 1240.0 | 22.0 | | |
| 22 | with | 1217.0 | 23.0 | with | |
| 23 | very | 1202.0 | 24.0 | | |
| 24 | mr | 1153.0 | 25.0 | | |
| 25 | his | 1145.0 | 26.0 | his | |
| 26 | at | 1031.0 | 27.0 | at | |
| 27 | so | 974.0 | 28.0 | so | |
| 28 | s | 935.0 | 29.0 | | |
| 29 | emma | 865.0 | 30.0 | | |
| 30 | all | 845.0 | 31.0 | all | |
| 31 | could | 837.0 | 32.0 | could | |
| 32 | would | 820.0 | 33.0 | would | |
| 33 | been | 759.0 | 34.0 | | |
| 34 | him | 759.0 | 35.0 | him | |
| 35 | no | 742.0 | 36.0 | no | |
| 36 | my | 728.0 | 37.0 | my | |
| 37 | mrs | 699.0 | 38.0 | | |
| 38 | on | 692.0 | 39.0 | on | |
| 39 | any | 654.0 | 40.0 | any | |

| | | | | |
|---|---|---|---|---|
| 40 | do | 640.0 | 41.0 | do |
| 41 | were | 600.0 | 42.0 | |
| 42 | miss | 599.0 | 43.0 | |
| 43 | me | 573.0 | 44.0 | me |
| 44 | by | 571.0 | 45.0 | by |
| 45 | will | 570.0 | 46.0 | will |
| 46 | must | 567.0 | 47.0 | |
| 47 | which | 556.0 | 48.0 | which |
| 48 | there | 549.0 | 49.0 | there |
| 49 | from | 546.0 | 50.0 | from |
| 50 | they | 540.0 | 51.0 | they |
| 51 | what | 536.0 | 52.0 | what |
| 52 | this | 526.0 | 53.0 | this |
| 53 | harriet | 506.0 | 54.0 | |
| 54 | or | 494.0 | 55.0 | or |
| 55 | such | 489.0 | 56.0 | |
| 56 | much | 486.0 | 57.0 | |
| 57 | if | 485.0 | 58.0 | if |
| 58 | said | 484.0 | 59.0 | |
| 59 | more | 467.0 | 60.0 | |
| 60 | an | 464.0 | 61.0 | an |
| 61 | are | 455.0 | 62.0 | |
| 62 | one | 452.0 | 63.0 | one |
| 63 | weston | 440.0 | 64.0 | |
| 64 | every | 435.0 | 65.0 | |
| 65 | them | 432.0 | 66.0 | them |
| 66 | am | 425.0 | 67.0 | |
| 67 | than | 415.0 | 68.0 | than |
| 68 | well | 401.0 | 69.0 | well |
| 69 | thing | 398.0 | 70.0 | |
| 70 | knightley | 389.0 | 71.0 | |
| 71 | elton | 385.0 | 72.0 | |
| 72 | think | 383.0 | 73.0 | think |
| 73 | how | 371.0 | 74.0 | how |
| 74 | should | 369.0 | 75.0 | |
| 75 | your | 364.0 | 76.0 | your |
| 76 | when | 363.0 | 77.0 | when |
| 77 | little | 359.0 | 78.0 | |
| 78 | being | 358.0 | 79.0 | |
| 79 | never | 358.0 | 80.0 | |
| 80 | good | 358.0 | 81.0 | good |
| 81 | did | 352.0 | 82.0 | |
| 82 | we | 349.0 | 83.0 | we |
| 83 | only | 341.0 | 84.0 | only |
| 84 | know | 337.0 | 85.0 | know |
| 85 | might | 326.0 | 86.0 | |
| 86 | woodhouse | 313.0 | 87.0 | |

| | | | | |
|---|---|---|---|---|
| 87 | say | 310.0 | 88.0 | say |
| 88 | now | 309.0 | 89.0 | now |
| 89 | their | 306.0 | 90.0 | their |
| 90 | jane | 301.0 | 91.0 | |
| 91 | own | 301.0 | 92.0 | |
| 92 | who | 294.0 | 93.0 | who |
| 93 | can | 284.0 | 94.0 | can |
| 94 | quite | 282.0 | 95.0 | |
| 95 | herself | 279.0 | 96.0 | |
| 96 | time | 279.0 | 97.0 | time |
| 97 | great | 264.0 | 98.0 | |
| 98 | some | 262.0 | 99.0 | some |
| 99 | nothing | 256.0 | 100.0 | |
| 100 | | | | up |
| 101 | | | | out |
| 102 | | | | about |
| 103 | | | | get |
| 104 | | | | go |
| 105 | | | | make |
| 106 | | | | like |
| 107 | | | | just |
| 108 | | | | take |
| 109 | | | | people |
| 110 | | | | into |
| 111 | | | | year |
| 112 | | | | see |
| 113 | | | | other |
| 114 | | | | then |
| 115 | | | | look |
| 116 | | | | come |
| 117 | | | | its |
| 118 | | | | over |
| 119 | | | | also |
| 120 | | | | back |
| 121 | | | | after |
| 122 | | | | use |
| 123 | | | | two |
| 124 | | | | our |
| 125 | | | | work |
| 126 | | | | first |
| 127 | | | | way |
| 128 | | | | even |
| 129 | | | | new |
| 130 | | | | want |
| 131 | | | | because |
| 132 | | | | these |
| 133 | | | | give |

| | |
|---|---|
| 134 | day |
| 135 | most |
| 136 | us |

| | Parts of speech | OEC rank | COCA rank[9] | \ |
|---|---|---|---|---|
| 0 | Preposition | 3 | 7, 9 | |
| 1 | Article | 1 | 1 | |
| 2 | Coordinator | 5 | 3 | |
| 3 | Preposition | 4 | 4 | |
| 4 | Pronoun | 10 | 11 | |
| 5 | Article | 6 | 5 | |
| 6 | Pronoun | 11 | 10 | |
| 7 | Possessive pronoun | 29, 106 | 42 | |
| 8 | | | | |
| 9 | Pronoun | 30 | 31 | |
| 10 | Preposition | 7 | 6, 128, 3038 | |
| 11 | Adverb et al. | 13 | 28, 2929 | |
| 12 | Pronoun | 18 | 14 | |
| 13 | Verb | 2 | 2 | |
| 14 | Subordinator, determiner | 8 | 12, 27, 903 | |
| 15 | Pronoun | 16 | 15 | |
| 16 | | | | |
| 17 | Preposition, adverb, coordinator | 22 | 23, 1715 | |
| 18 | Adverb, preposition | 17 | 33, 49, 129 | |
| 19 | Preposition | 12 | 13, 2339 | |
| 20 | Verb | 9 | 8 | |
| 21 | | | | |
| 22 | Preposition | 15 | 16 | |
| 23 | | | | |
| 24 | | | | |
| 25 | Possessive pronoun | 23 | 25, 1887 | |
| 26 | Preposition | 20 | 22 | |
| 27 | Coordinator, adverb, et al. | 41 | 55, 196 | |
| 28 | | | | |
| 29 | | | | |
| 30 | Adjective | 36 | 43, 222 | |
| 31 | Verb | 67 | 71 | |
| 32 | Verb | 37 | 41 | |
| 33 | | | | |
| 34 | Pronoun | 58 | 68 | |
| 35 | Determiner, adverb | 56 | 93, 699, 916, 1111, 4555 | |
| 36 | Possessive pronoun | 34 | 44 | |
| 37 | | | | |
| 38 | Preposition | 14 | 17, 155 | |
| 39 | Pronoun | 95 | 109, 4720 | |
| 40 | Verb, noun | 19 | 18 | |
| 41 | | | | |

| | | | |
|---|---|---|---|
| 42 | | | |
| 43 | Pronoun | 50 | 61 |
| 44 | Preposition | 24 | 30, 1190 |
| 45 | Verb, noun | 33 | 48, 1506 |
| 46 | | | |
| 47 | Pronoun | 48 | 58 |
| 48 | Adverb, pronoun, et al. | 38 | 53, 116 |
| 49 | Preposition | 25 | 26 |
| 50 | Pronoun | 26 | 21 |
| 51 | Pronoun, adverb, et al. | 40 | 34 |
| 52 | Determiner, adverb, noun | 21 | 20, 4665 |
| 53 | | | |
| 54 | Coordinator | 31 | 32 |
| 55 | | | |
| 56 | | | |
| 57 | Preposition | 44 | 40 |
| 58 | | | |
| 59 | | | |
| 60 | Article | 32 | (a) |
| 61 | | | |
| 62 | Noun, adjective, et al. | 35 | 51, 104, 839 |
| 63 | | | |
| 64 | | | |
| 65 | Pronoun | 68 | 59 |
| 66 | | | |
| 67 | Preposition | 71 | 73, 712 |
| 68 | Adverb | 89 | 100, 644 |
| 69 | | | |
| 70 | | | |
| 71 | | | |
| 72 | Verb | 79 | 56 |
| 73 | Adverb | 85 | 76 |
| 74 | | | |
| 75 | Possessive pronoun | 64 | 69 |
| 76 | Adverb | 51 | 57, 136 |
| 77 | | | |
| 78 | | | |
| 79 | | | |
| 80 | Adjective | 65 | 110, 2280 |
| 81 | | | |
| 82 | Pronoun | 27 | 24 |
| 83 | Adverb | 75 | 101, 329 |
| 84 | Verb, noun | 59 | 47 |
| 85 | | | |
| 86 | | | |
| 87 | Verb et al. | 28 | 19 |
| 88 | Preposition | 73 | 72, 1906 |

| | | | |
|---|---|---|---|
| 89 | Possessive pronoun | 39 | 36 |
| 90 | | | |
| 91 | | | |
| 92 | Pronoun, noun | 46 | 38 |
| 93 | Verb, noun | 53 | 37, 2973 |
| 94 | | | |
| 95 | | | |
| 96 | Noun | 55 | 52 |
| 97 | | | |
| 98 | Determiner | 66 | 60 |
| 99 | | | |
| 100 | Adverb, preposition, et al. | 42 | 50, 456 |
| 101 | Preposition | 43 | 64, 149 |
| 102 | Preposition, adverb, et al. | 45 | 46, 179 |
| 103 | Verb | 47 | 39 |
| 104 | Verb, noun | 49 | 35 |
| 105 | Verb, noun | 52 | 45 |
| 106 | Preposition, verb | 54 | 74, 208, 1123, 1684, 2702 |
| 107 | Adjective | 57 | 66, 1823 |
| 108 | Verb, noun | 60 | 63 |
| 109 | Noun | 61 | 62 |
| 110 | Preposition | 62 | 65 |
| 111 | Noun | 63 | 54 |
| 112 | Verb | 69 | 67 |
| 113 | Adjective, pronoun | 70 | 75, 715, 2355 |
| 114 | Adverb | 72 | 77 |
| 115 | Verb | 74 | 85, 604 |
| 116 | Verb | 76 | 70 |
| 117 | Possessive pronoun | 77 | 78 |
| 118 | Preposition | 78 | 124, 182 |
| 119 | Adverb | 80 | 87 |
| 120 | Noun, adverb | 81 | 108, 323, 1877 |
| 121 | Preposition | 82 | 120, 260 |
| 122 | Verb, noun | 83 | 92, 429 |
| 123 | Noun | 84 | 80 |
| 124 | Possessive pronoun | 86 | 79 |
| 125 | Verb, noun | 87 | 117, 199 |
| 126 | Adjective | 88 | 86, 2064 |
| 127 | Noun, adverb | 90 | 84, 4090 |
| 128 | Adjective | 91 | 107, 484 |
| 129 | Adjective et al. | 92 | 88 |
| 130 | Verb | 93 | 83 |
| 131 | Preposition | 94 | 89, 509 |
| 132 | Pronoun | 96 | 82 |
| 133 | Verb | 97 | 98 |
| 134 | Noun | 98 | 90 |
| 135 | Adverb | 99 | 144, 187 |

| | | | |
|---|---|---|---|
| 136 | Pronoun | 100 | 113 |

| | Dolch level | Polysemy |
|---|---|---|
| 0 | Pre-primer | 17.0 |
| 1 | Pre-primer | 12.0 |
| 2 | Pre-primer | 16.0 |
| 3 | Grade 1 | 12.0 |
| 4 | Pre-primer | 7.0 |
| 5 | Pre-primer | 20.0 |
| 6 | Pre-primer | 18.0 |
| 7 | Grade 1 | 3.0 |
| 8 | | |
| 9 | Primer | 7.0 |
| 10 | Pre-primer | 23.0 |
| 11 | Pre-primer | 5.0 |
| 12 | Pre-primer | 9.0 |
| 13 | Primer | 21.0 |
| 14 | Primer | 17.0 |
| 15 | Primer | 7.0 |
| 16 | | |
| 17 | Primer | 17.0 |
| 18 | Grade 1 | 17.0 |
| 19 | Pre-primer | 19.0 |
| 20 | Primer | 25.0 |
| 21 | | |
| 22 | Primer | 11.0 |
| 23 | | |
| 24 | | |
| 25 | Grade 1 | 6.0 |
| 26 | Primer | 14.0 |
| 27 | Primer | 18.0 |
| 28 | | |
| 29 | | |
| 30 | Primer | 15.0 |
| 31 | Grade 1 | 6.0 |
| 32 | Grade 2 | 13.0 |
| 33 | | |
| 34 | Grade 1 | 5.0 |
| 35 | Primer | 10.0 |
| 36 | Pre-primer | 5.0 |
| 37 | | |
| 38 | Primer | 43.0 |
| 39 | Grade 1 | 4.0 |
| 40 | Primer | 38.0 |
| 41 | | |
| 42 | | |
| 43 | Pre-primer | 10.0 |

| | | |
|---|---|---|
| 44 | Grade 1 | 19.0 |
| 45 | Primer | 16.0 |
| 46 | | |
| 47 | Grade 2 | 7.0 |
| 48 | Primer | 14.0 |
| 49 | Grade 1 | 4.0 |
| 50 | Primer | 6.0 |
| 51 | Primer | 19.0 |
| 52 | Primer | 9.0 |
| 53 | | |
| 54 | Grade 2 | 11.0 |
| 55 | | |
| 56 | | |
| 57 | Grade 3 | 9.0 |
| 58 | | |
| 59 | | |
| 60 | Grade 1 | 6.0 |
| 61 | | |
| 62 | Pre-primer | 24.0 |
| 63 | | |
| 64 | | |
| 65 | Grade 1 | 3.0 |
| 66 | | |
| 67 | | 4.0 |
| 68 | Primer | 30.0 |
| 69 | | |
| 70 | | |
| 71 | | |
| 72 | Grade 1 | 10.0 |
| 73 | Grade 1 | 11.0 |
| 74 | | |
| 75 | Grade 2 | 4.0 |
| 76 | Grade 1 | 11.0 |
| 77 | | |
| 78 | | |
| 79 | | |
| 80 | Primer | 32.0 |
| 81 | | |
| 82 | Pre-primer | 6.0 |
| 83 | Grade 3 | 11.0 |
| 84 | Grade 1 | 13.0 |
| 85 | | |
| 86 | | |
| 87 | Primer | 17.0 |
| 88 | Primer | 13.0 |
| 89 | Grade 2 | 2.0 |
| 90 | | |

| | | |
|---|---|---|
| 91 | | |
| 92 | Primer | 5.0 |
| 93 | Pre-primer | 18.0 |
| 94 | | |
| 95 | | |
| 96 | Dolch list of 95 nouns | 14.0 |
| 97 | | |
| 98 | Grade 1 | 10.0 |
| 99 | | |
| 100 | Pre-primer | 50.0 |
| 101 | Primer | 38.0 |
| 102 | Grade 3 | 18.0 |
| 103 | Primer | 37.0 |
| 104 | Pre-primer | 54.0 |
| 105 | Grade 2 [as "made"] | 48.0 |
| 106 | Primer | 26.0 |
| 107 | Grade 1 | 14.0 |
| 108 | Grade 1 | 66.0 |
| 109 | | 9.0 |
| 110 | Primer | 10.0 |
| 111 | | 7.0 |
| 112 | | 25.0 |
| 113 | | 12.0 |
| 114 | Grade 1 | 10.0 |
| 115 | Pre-primer | 17.0 |
| 116 | Pre-primer | 20.0 |
| 117 | Grade 2 | 2.0 |
| 118 | Grade 1 | 19.0 |
| 119 | | 2.0 |
| 120 | Dolch list of 95 nouns | 36.0 |
| 121 | Grade 1 | 14.0 |
| 122 | Grade 2 | 17.0 |
| 123 | Pre-primer | 6.0 |
| 124 | Primer | 3.0 |
| 125 | Grade 2 | 28.0 |
| 126 | Grade 2 | 10.0 |
| 127 | Dolch list of 95 nouns | 16.0 |
| 128 | | 23.0 |
| 129 | Primer | 18.0 |
| 130 | Primer | 10.0 |
| 131 | Grade 2 | 7.0 |
| 132 | Grade 2 | 2.0 |
| 133 | Grade 1 | 19.0 |
| 134 | Dolch list of 95 nouns | 9.0 |
| 135 | | 12.0 |
| 136 | Grade 2 | 6.0 |

[34]:
```
len(top_100)
```

[34]: 137

## 0.1 Words that are in both the Austen 100 most frequently used words and the Wikipedia top 100 most frequently used English words list

[35]:
```
matched_words = words_merged[['Austen_Word','Austen_Rank','Word','OEC rank']].
    ↪dropna()
matched_words['Austen_Rank'] = matched_words['Austen_Rank'].astype(int)
matched_words.reset_index(inplace=True)
matched_words
```

[35]:

| | index | Austen_Word | Austen_Rank | Word | OEC rank |
|---|---|---|---|---|---|
| 0 | 0 | to | 1 | to | 3 |
| 1 | 1 | the | 2 | the | 1 |
| 2 | 2 | and | 3 | and | 5 |
| 3 | 3 | of | 4 | of | 4 |
| 4 | 4 | i | 5 | i | 10 |
| 5 | 5 | a | 6 | a | 6 |
| 6 | 6 | it | 7 | it | 11 |
| 7 | 7 | her | 8 | her | 29, 106 |
| 8 | 9 | she | 10 | she | 30 |
| 9 | 10 | in | 11 | in | 7 |
| 10 | 11 | not | 12 | not | 13 |
| 11 | 12 | you | 13 | you | 18 |
| 12 | 13 | be | 14 | be | 2 |
| 13 | 14 | that | 15 | that | 8 |
| 14 | 15 | he | 16 | he | 16 |
| 15 | 17 | but | 18 | but | 22 |
| 16 | 18 | as | 19 | as | 17 |
| 17 | 19 | for | 20 | for | 12 |
| 18 | 20 | have | 21 | have | 9 |
| 19 | 22 | with | 23 | with | 15 |
| 20 | 25 | his | 26 | his | 23 |
| 21 | 26 | at | 27 | at | 20 |
| 22 | 27 | so | 28 | so | 41 |
| 23 | 30 | all | 31 | all | 36 |
| 24 | 31 | could | 32 | could | 67 |
| 25 | 32 | would | 33 | would | 37 |
| 26 | 34 | him | 35 | him | 58 |
| 27 | 35 | no | 36 | no | 56 |
| 28 | 36 | my | 37 | my | 34 |
| 29 | 38 | on | 39 | on | 14 |
| 30 | 39 | any | 40 | any | 95 |
| 31 | 40 | do | 41 | do | 19 |
| 32 | 43 | me | 44 | me | 50 |

| | | | | | |
|---|---|---|---|---|---|
| 33 | 44 | by | 45 | by | 24 |
| 34 | 45 | will | 46 | will | 33 |
| 35 | 47 | which | 48 | which | 48 |
| 36 | 48 | there | 49 | there | 38 |
| 37 | 49 | from | 50 | from | 25 |
| 38 | 50 | they | 51 | they | 26 |
| 39 | 51 | what | 52 | what | 40 |
| 40 | 52 | this | 53 | this | 21 |
| 41 | 54 | or | 55 | or | 31 |
| 42 | 57 | if | 58 | if | 44 |
| 43 | 60 | an | 61 | an | 32 |
| 44 | 62 | one | 63 | one | 35 |
| 45 | 65 | them | 66 | them | 68 |
| 46 | 67 | than | 68 | than | 71 |
| 47 | 68 | well | 69 | well | 89 |
| 48 | 72 | think | 73 | think | 79 |
| 49 | 73 | how | 74 | how | 85 |
| 50 | 75 | your | 76 | your | 64 |
| 51 | 76 | when | 77 | when | 51 |
| 52 | 80 | good | 81 | good | 65 |
| 53 | 82 | we | 83 | we | 27 |
| 54 | 83 | only | 84 | only | 75 |
| 55 | 84 | know | 85 | know | 59 |
| 56 | 87 | say | 88 | say | 28 |
| 57 | 88 | now | 89 | now | 73 |
| 58 | 89 | their | 90 | their | 39 |
| 59 | 92 | who | 93 | who | 46 |
| 60 | 93 | can | 94 | can | 53 |
| 61 | 96 | time | 97 | time | 55 |
| 62 | 98 | some | 99 | some | 66 |

[36]:
```
len(matched_words)
```

[36]: 63

63 of the words most frequently used in the Austen text also appear in the top 100 most frequently used words in the English language. This is not suprising since the list includes many words that are not necessarily meaningful to the story, but are words like, 'the', 'and', 'to', 'of', 'a', and 'in'.

### 0.1.1 Words in either the Austen 100 most frequently used words or the Wikipedia top 100 most frequently used English words list but not both

[37]:
```
unmatched_words = top_100[['Austen_Word','Austen_Rank','Word','OEC rank']]
unmatched_words = unmatched_words[(unmatched_words['Austen_Word']=='') | (unmatched_words['Word']=='')]
unmatched_words.reset_index(inplace=True)
unmatched_words
```

[37]:

| | index | Austen_Word | Austen_Rank | Word | OEC rank |
|---|---|---|---|---|---|
| 0 | 8 | was | 9.0 | | |
| 1 | 16 | had | 17.0 | | |
| 2 | 21 | is | 22.0 | | |
| 3 | 23 | very | 24.0 | | |
| 4 | 24 | mr | 25.0 | | |
| 5 | 28 | s | 29.0 | | |
| 6 | 29 | emma | 30.0 | | |
| 7 | 33 | been | 34.0 | | |
| 8 | 37 | mrs | 38.0 | | |
| 9 | 41 | were | 42.0 | | |
| 10 | 42 | miss | 43.0 | | |
| 11 | 46 | must | 47.0 | | |
| 12 | 53 | harriet | 54.0 | | |
| 13 | 55 | such | 56.0 | | |
| 14 | 56 | much | 57.0 | | |
| 15 | 58 | said | 59.0 | | |
| 16 | 59 | more | 60.0 | | |
| 17 | 61 | are | 62.0 | | |
| 18 | 63 | weston | 64.0 | | |
| 19 | 64 | every | 65.0 | | |
| 20 | 66 | am | 67.0 | | |
| 21 | 69 | thing | 70.0 | | |
| 22 | 70 | knightley | 71.0 | | |
| 23 | 71 | elton | 72.0 | | |
| 24 | 74 | should | 75.0 | | |
| 25 | 77 | little | 78.0 | | |
| 26 | 78 | being | 79.0 | | |
| 27 | 79 | never | 80.0 | | |
| 28 | 81 | did | 82.0 | | |
| 29 | 85 | might | 86.0 | | |
| 30 | 86 | woodhouse | 87.0 | | |
| 31 | 90 | jane | 91.0 | | |
| 32 | 91 | own | 92.0 | | |
| 33 | 94 | quite | 95.0 | | |
| 34 | 95 | herself | 96.0 | | |
| 35 | 97 | great | 98.0 | | |
| 36 | 99 | nothing | 100.0 | | |
| 37 | 100 | | | up | 42 |
| 38 | 101 | | | out | 43 |
| 39 | 102 | | | about | 45 |
| 40 | 103 | | | get | 47 |
| 41 | 104 | | | go | 49 |
| 42 | 105 | | | make | 52 |
| 43 | 106 | | | like | 54 |
| 44 | 107 | | | just | 57 |
| 45 | 108 | | | take | 60 |

| | | | |
|---|---|---|---|
| 46 | 109 | people | 61 |
| 47 | 110 | into | 62 |
| 48 | 111 | year | 63 |
| 49 | 112 | see | 69 |
| 50 | 113 | other | 70 |
| 51 | 114 | then | 72 |
| 52 | 115 | look | 74 |
| 53 | 116 | come | 76 |
| 54 | 117 | its | 77 |
| 55 | 118 | over | 78 |
| 56 | 119 | also | 80 |
| 57 | 120 | back | 81 |
| 58 | 121 | after | 82 |
| 59 | 122 | use | 83 |
| 60 | 123 | two | 84 |
| 61 | 124 | our | 86 |
| 62 | 125 | work | 87 |
| 63 | 126 | first | 88 |
| 64 | 127 | way | 90 |
| 65 | 128 | even | 91 |
| 66 | 129 | new | 92 |
| 67 | 130 | want | 93 |
| 68 | 131 | because | 94 |
| 69 | 132 | these | 96 |
| 70 | 133 | give | 97 |
| 71 | 134 | day | 98 |
| 72 | 135 | most | 99 |
| 73 | 136 | us | 100 |