# Name:Viviyan Richards W

Roll no:205229133

## Lab2. Computing Bigram Frequencies

### EXERCISE-1: Process simple bigram data file

### STEP 1: OPEN the file, count_2w.txt

In [1]:

```
import io
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:

```
with io.open('count_2w.txt','r',encoding='utf8') as f:
    text = f.readlines()
```

### STEP 2: build goog2w_list

In [3]:

```
mini = text[:10]
```

In [4]:

```
nimi = text[:]
```

In [5]:

```
mini[0].split()
```

Out[5]:

```
['0Uplink', 'verified', '523545']
```

In [6]:

```
mini_list = []
for m in mini:
    (w1, w2, count) = m.split()
    count = int(count)
    mini_list.append(((w1, w2), count))
mini_list
```

Out[6]:

```
[(('0Uplink', 'verified'), 523545),
 (('0km', 'to'), 116103),
 (('1000s', 'of'), 939476),
 (('100s', 'of'), 539389),
 (('100th', 'anniversary'), 158621),
 (('10am', 'to'), 376141),
 (('10th', 'and'), 183715),
 (('10th', 'anniversary'), 242830),
 (('10th', 'century'), 117755),
 (('10th', 'grade'), 174046)]
```

In [7]:
```
mini_list[0]
```

Out[7]:
```
(('0Uplink', 'verified'), 523545)
```

In [8]:
```
goog2w_list = []
for m in nimi:
    (w1, w2, count) = m.split()
    count = int(count)
    goog2w_list.append(((w1, w2), count))
goog2w_list
```

Out[8]:
```
[(('0Uplink', 'verified'), 523545),
 (('0km', 'to'), 116103),
 (('1000s', 'of'), 939476),
 (('100s', 'of'), 539389),
 (('100th', 'anniversary'), 158621),
 (('10am', 'to'), 376141),
 (('10th', 'and'), 183715),
 (('10th', 'anniversary'), 242830),
 (('10th', 'century'), 117755),
 (('10th', 'grade'), 174046),
 (('10th', 'in'), 107194),
 (('10th', 'of'), 277970),
 (('11am', 'to'), 127624),
 (('11th', 'and'), 178884),
 (('11th', 'century'), 168601),
 (('11th', 'grade'), 126301),
 (('11th', 'of'), 189501),
 (('125Mbps', 'w'), 108645),
 (('12th', 'and'), 136706),
```

In [9]:
```
goog2w_list[0]
```

Out[9]:
```
(('0Uplink', 'verified'), 523545)
```

### STEP 3: build goog2w_fd

In [10]:
```
!pip install nltk
```

```
Requirement already satisfied: nltk in c:\programdata\anaconda3\lib\site-packag
es (3.5)
Requirement already satisfied: joblib in c:\programdata\anaconda3\lib\site-pack
ages (from nltk) (0.17.0)
Requirement already satisfied: tqdm in c:\programdata\anaconda3\lib\site-packag
es (from nltk) (4.50.2)
Requirement already satisfied: click in c:\programdata\anaconda3\lib\site-packa
ges (from nltk) (7.1.2)
Requirement already satisfied: regex in c:\programdata\anaconda3\lib\site-packa
ges (from nltk) (2020.10.15)
```

```
In [11]: import nltk
         goog2w_fd = nltk.FreqDist()
         goog2w_fd

Out[11]: FreqDist({})
```

```
In [12]: for m in text:
             w1, w2, count = m.split()
             goog2w_fd[(w1, w2)] = count
```

```
In [13]: goog2w_fd[('of', 'the')]

Out[13]: '2766332391'
```

```
In [14]: goog2w_fd[('so', 'beautiful')]

Out[14]: '612472'
```

**STEP 4: explore**

**1. What are the top-10 bigrams?**

```
In [15]: goog2w_fd.most_common(10)

Out[15]: [(('You', 'think'), '999988'),
          (('a', 'middle'), '999987'),
          (('his', 'wife'), '9999448'),
          (('traditional', 'and'), '999927'),
          (('Ask', 'your'), '999907'),
          (('towards', 'the'), '9998989'),
          (('<S>', 'central'), '999848'),
          (('no', 'man'), '999833'),
          (('committee', 'members'), '999819'),
          (('each', 'country'), '999818')]
```

**STEP 5: pickle the data**

```
In [16]: import pickle as pkl
```

```
In [17]: with open('goog2w_list.pkl', 'ab') as handle:
             pkl.dump(goog2w_list,handle)
```

```
In [18]: with open('goog2w_fd.pkl', 'ab') as handle:
             pkl.dump(goog2w_fd,handle)
```

## EXERCISE - 2 Frequency distribution from Jane Austen Novels

### A. opens (and later closes) the text file, reads in the string content,

```
In [19]: with open('austen-emma.txt','r') as fl:
             cona=fl.read()
```

```
In [20]: with open('austen-persuasion.txt','r') as flp:
             conp=flp.read()
```

```
In [21]: with open('austen-sense.txt','r') as fls:
             cons=fls.read()
```

### B. builds a list of individual sentences,

```
In [22]: from nltk.tokenize import sent_tokenize as st
```

```
In [23]: st(cona)
```

```
Out[23]: ['[Emma by Jane Austen 1816]\n\nVOLUME I\n\nCHAPTER I\n\n\nEmma Woodhouse, ha
         ndsome, clever, and rich, with a comfortable home\nand happy disposition, see
         med to unite some of the best blessings\nof existence; and had lived nearly t
         wenty-one years in the world\nwith very little to distress or vex her.',
          "She was the youngest of the two daughters of a most affectionate,\nindulgen
         t father; and had, in consequence of her sister's marriage,\nbeen mistress of
         his house from a very early period.",
          'Her mother\nhad died too long ago for her to have more than an indistinct\n
         remembrance of her caresses; and her place had been supplied\nby an excellent
         woman as governess, who had fallen little short\nof a mother in affection.',
          "Sixteen years had Miss Taylor been in Mr. Woodhouse's family,\nless as a go
         verness than a friend, very fond of both daughters,\nbut particularly of Emm
         a.",
          'Between _them_ it was more the intimacy\nof sisters.',
          "Even before Miss Taylor had ceased to hold the nominal\noffice of governes
         s, the mildness of her temper had hardly allowed\nher to impose any restrain
         t; and the shadow of authority being\nnow long passed away, they had been liv
         ing together as friend and\nfriend very mutually attached, and Emma doing jus
         t what she liked;\nhighly esteeming Miss Taylor's judgment, but directed chie
```

In [24]:
```
st(conp)
```

Out[24]:
```
['[Persuasion by Jane Austen 1818]\n\n\nChapter 1\n\n\nSir Walter Elliot, of
Kellynch Hall, in Somersetshire, was a man who,\nfor his own amusement, never
took up any book but the Baronetage;\nthere he found occupation for an idle h
our, and consolation in a\ndistressed one; there his faculties were roused in
to admiration and\nrespect, by contemplating the limited remnant of the earli
est patents;\nthere any unwelcome sensations, arising from domestic affairs\n
changed naturally into pity and contempt as he turned over\nthe almost endles
s creations of the last century; and there,\nif every other leaf were powerle
ss, he could read his own history\nwith an interest which never failed.',
 'This was the page at which\nthe favourite volume always opened:\n\n
"ELLIOT OF KELLYNCH HALL.',
 '"Walter Elliot, born March 1, 1760, married, July 15, 1784, Elizabeth,\ndau
ghter of James Stevenson, Esq.',
 'of South Park, in the county of\nGloucester, by which lady (who died 1800)
he has issue Elizabeth,\nborn June 1, 1785; Anne, born August 9, 1787; a stil
l-born son,\nNovember 5, 1789; Mary, born November 20, 1791."',
 'Precisely such had the paragraph originally stood from the printer\'s hand
s;\nbut Sir Walter had improved it by adding, for the information of\nhimself
and his family, these words, after the date of Mary\'s birth--\n"Married, Dec
```

In [25]:
```
st(cons)
```

Out[25]:
```
['[Sense and Sensibility by Jane Austen 1811]\n\nCHAPTER 1\n\n\nThe family of
Dashwood had long been settled in Sussex.',
 'Their estate was large, and their residence was at Norland Park,\nin the ce
ntre of their property, where, for many generations,\nthey had lived in so re
spectable a manner as to engage\nthe general good opinion of their surroundin
g acquaintance.',
 'The late owner of this estate was a single man, who lived\nto a very advanc
ed age, and who for many years of his life,\nhad a constant companion and hou
sekeeper in his sister.',
 'But her death, which happened ten years before his own,\nproduced a great a
lteration in his home; for to supply\nher loss, he invited and received into
his house the family\nof his nephew Mr. Henry Dashwood, the legal inheritor\n
of the Norland estate, and the person to whom he intended\nto bequeath it.',
 "In the society of his nephew and niece,\nand their children, the old Gentle
man's days were\ncomfortably spent.",
 'His attachment to them all increased.',
 'The constant attention of Mr. and Mrs. Henry Dashwood\nto his wishes, which
proceeded not merely from interest,\nbut from goodness of heart, gave him eve
ry degree of solid\ncomfort which his age could receive; and the cheerfulness
```

### C. prints out how many sentences there are,

In [26]:
```
print(len(st(cona)))
print(len(st(conp)))
print(len(st(cons)))
```

```
7493
3654
4833
```

### E. prints the token and the type counts of this corpus,

In [27]:
```
from nltk.tokenize import word_tokenize
```

In [28]:
```
t1=word_tokenize(cona)
print(t1)
```

```
['[', 'Emma', 'by', 'Jane', 'Austen', '1816', ']', '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', 'exi
stence', ';', 'and', 'had', 'lived', 'nearly', 'twenty-one', 'years', 'in',
'the', 'world', 'with', 'very', 'little', 'to', 'distress', 'or', 'vex', 'he
r', '.', 'She', 'was', 'the', 'youngest', 'of', 'the', 'two', 'daughters', 'o
f', 'a', 'most', 'affectionate', ',', 'indulgent', 'father', ';', 'and', 'ha
d', ',', 'in', 'consequence', 'of', 'her', 'sister', "'s", 'marriage', ',',
'been', 'mistress', 'of', 'his', 'house', 'from', 'a', 'very', 'early', 'peri
od', '.', 'Her', 'mother', 'had', 'died', 'too', 'long', 'ago', 'for', 'her',
'to', 'have', 'more', 'than', 'an', 'indistinct', 'remembrance', 'of', 'her',
'caresses', ';', 'and', 'her', 'place', 'had', 'been', 'supplied', 'by', 'a
n', 'excellent', 'woman', 'as', 'governess', ',', 'who', 'had', 'fallen', 'li
ttle', 'short', 'of', 'a', 'mother', 'in', 'affection', '.', 'Sixteen', 'year
s', 'had', 'Miss', 'Taylor', 'been', 'in', 'Mr.', 'Woodhouse', "'s", 'famil
y', ',', 'less', 'as', 'a', 'governess', 'than', 'a', 'friend', ',', 'very',
'fond', 'of', 'both', 'daughters', ',', 'but', 'particularly', 'of', 'Emma',
'.', 'Between', '_them_', 'it', 'was', 'more', 'the', 'intimacy', 'of', 'sist
```

In [29]:
```
t2=word_tokenize(conp)
print(t2)
```

```
['[', 'Persuasion', 'by', 'Jane', 'Austen', '1818', ']', 'Chapter', '1', 'Si
r', 'Walter', 'Elliot', ',', 'of', 'Kellynch', 'Hall', ',', 'in', 'Somersetsh
ire', ',', 'was', 'a', 'man', 'who', ',', 'for', 'his', 'own', 'amusement',
',', 'never', 'took', 'up', 'any', 'book', 'but', 'the', 'Baronetage', ';',
'there', 'he', 'found', 'occupation', 'for', 'an', 'idle', 'hour', ',', 'an
d', 'consolation', 'in', 'a', 'distressed', 'one', ';', 'there', 'his', 'facu
lties', 'were', 'roused', 'into', 'admiration', 'and', 'respect', ',', 'by',
'contemplating', 'the', 'limited', 'remnant', 'of', 'the', 'earliest', 'paten
ts', ';', 'there', 'any', 'unwelcome', 'sensations', ',', 'arising', 'from',
'domestic', 'affairs', 'changed', 'naturally', 'into', 'pity', 'and', 'contem
pt', 'as', 'he', 'turned', 'over', 'the', 'almost', 'endless', 'creations',
'of', 'the', 'last', 'century', ';', 'and', 'there', ',', 'if', 'every', 'oth
er', 'leaf', 'were', 'powerless', ',', 'he', 'could', 'read', 'his', 'own',
'history', 'with', 'an', 'interest', 'which', 'never', 'failed', '.', 'This',
'was', 'the', 'page', 'at', 'which', 'the', 'favourite', 'volume', 'always',
'opened', ':', '``', 'ELLIOT', 'OF', 'KELLYNCH', 'HALL', '.', '``', 'Walter',
'Elliot', ',', 'born', 'March', '1', ',', '1760', ',', 'married', ',', 'Jul
y', '15', ',', '1784', ',', 'Elizabeth', ',', 'daughter', 'of', 'James', 'Ste
venson', ',', 'Esq', '.', 'of', 'South', 'Park', ',', 'in', 'the', 'county',
```

In [30]:
```
t3 = word_tokenize(cons)
print(t3)
```

```
['[', 'Sense', 'and', 'Sensibility', 'by', 'Jane', 'Austen', '1811', ']', 'CH
APTER', '1', 'The', 'family', 'of', 'Dashwood', 'had', 'long', 'been', 'settl
ed', 'in', 'Sussex', '.', 'Their', 'estate', 'was', 'large', ',', 'and', 'the
ir', 'residence', 'was', 'at', 'Norland', 'Park', ',', 'in', 'the', 'centre',
'of', 'their', 'property', ',', 'where', ',', 'for', 'many', 'generations',
',', 'they', 'had', 'lived', 'in', 'so', 'respectable', 'a', 'manner', 'as',
'to', 'engage', 'the', 'general', 'good', 'opinion', 'of', 'their', 'surround
ing', 'acquaintance', '.', 'The', 'late', 'owner', 'of', 'this', 'estate', 'w
as', 'a', 'single', 'man', ',', 'who', 'lived', 'to', 'a', 'very', 'advance
d', 'age', ',', 'and', 'who', 'for', 'many', 'years', 'of', 'his', 'life',
',', 'had', 'a', 'constant', 'companion', 'and', 'housekeeper', 'in', 'his',
'sister', '.', 'But', 'her', 'death', ',', 'which', 'happened', 'ten', 'year
s', 'before', 'his', 'own', ',', 'produced', 'a', 'great', 'alteration', 'i
n', 'his', 'home', ';', 'for', 'to', 'supply', 'her', 'loss', ',', 'he', 'inv
ited', 'and', 'received', 'into', 'his', 'house', 'the', 'family', 'of', 'hi
s', 'nephew', 'Mr.', 'Henry', 'Dashwood', ',', 'the', 'legal', 'inheritor',
'of', 'the', 'Norland', 'estate', ',', 'and', 'the', 'person', 'to', 'whom',
'he', 'intended', 'to', 'bequeath', 'it', '.', 'In', 'the', 'society', 'of',
'his', 'nephew', 'and', 'niece', ',', 'and', 'their', 'children', ',', 'the',
```

**F. builds a frequency count dictionary of words,**

In [31]:
```
from nltk import *
```

In [32]:
```
da1 = FreqDist(t1)
da1
```

Out[32]:
```
FreqDist({',': 12016, '.': 6355, 'to': 5125, 'the': 4844, 'and': 4653, 'of': 42
72, 'I': 3177, '--': 3100, 'a': 3001, "'": 2452, ...})
```

In [33]:
```
da2 = FreqDist(t2)
da2
```

Out[33]:
```
FreqDist({',': 7024, 'the': 3119, '.': 3119, 'to': 2751, 'and': 2724, 'of': 256
2, 'a': 1528, 'in': 1340, 'was': 1330, ';': 1319, ...})
```

In [34]:
```
da3 = FreqDist(t3)
da3
```

Out[34]:
```
FreqDist({',': 9901, 'to': 4050, '.': 4023, 'the': 3860, 'of': 3564, 'and': 334
8, 'her': 2434, 'a': 2025, 'I': 2003, 'in': 1873, ...})
```

**G. prints the top 50 word types and their counts.**

In [35]:

```
da1.most_common(50)
```

Out[35]:

```
[(',', 12016),
 ('.', 6355),
 ('to', 5125),
 ('the', 4844),
 ('and', 4653),
 ('of', 4272),
 ('I', 3177),
 ('--', 3100),
 ('a', 3001),
 ("''", 2452),
 ('was', 2383),
 ('her', 2360),
 (';', 2353),
 ('not', 2242),
 ('in', 2103),
 ('it', 2103),
 ('be', 1965),
 ('she', 1774),
 ('``', 1735),
 ('that', 1729),
 ('you', 1664),
 ('had', 1605),
 ('as', 1387),
 ('he', 1365),
 ('for', 1320),
 ('have', 1301),
 ('is', 1221),
 ('with', 1185),
 ('very', 1151),
 ('but', 1148),
 ('Mr.', 1091),
 ('his', 1084),
 ('!', 1063),
 ('at', 996),
 ('so', 918),
 ("'s", 866),
 ('Emma', 855),
 ('all', 831),
 ('could', 824),
 ('would', 813),
 ('been', 755),
 ('him', 748),
 ('on', 674),
 ('Mrs.', 668),
 ('any', 651),
 ('?', 621),
 ('my', 619),
 ('no', 616),
 ('Miss', 592),
 ('were', 590)]
```

```
In [36]: da2.most_common(50)
```

```
Out[36]: [(',', 7024),
          ('the', 3119),
          ('.', 3119),
          ('to', 2751),
          ('and', 2724),
          ('of', 2562),
          ('a', 1528),
          ('in', 1340),
          ('was', 1330),
          (';', 1319),
          ('had', 1177),
          ('her', 1158),
          ('I', 1123),
          ('not', 968),
          ('be', 949),
          ("''", 912),
          ('it', 857),
          ('that', 853),
          ('she', 819),
          ('as', 787),
          ('he', 736),
          ('for', 695),
          ('``', 652),
          ('with', 643),
          ('his', 625),
          ('have', 583),
          ('but', 553),
          ('you', 548),
          ('at', 519),
          ('all', 517),
          ('Anne', 496),
          ('been', 496),
          ('him', 467),
          ("'s", 464),
          ('could', 444),
          ('were', 426),
          ('very', 425),
          ('which', 415),
          ('by', 409),
          ('is', 393),
          ('on', 386),
          ('would', 351),
          ('so', 338),
          ('She', 327),
          ('they', 323),
          ('!', 318),
          ('no', 309),
          ('Captain', 297),
          ('Mrs', 291),
          ('from', 290)]
```

```
In [37]: da3.most_common(50)
```

```
Out[37]: [(',', 9901),
          ('to', 4050),
          ('.', 4023),
          ('the', 3860),
          ('of', 3564),
          ('and', 3348),
          ('her', 2434),
          ('a', 2025),
          ('I', 2003),
          ('in', 1873),
          ('was', 1846),
          ("''", 1807),
          (';', 1572),
          ('it', 1561),
          ('she', 1333),
          ('be', 1304),
          ('not', 1301),
          ('that', 1296),
          ('``', 1277),
          ('for', 1231),
          ('as', 1179),
          ('--', 1178),
          ('you', 1034),
          ('with', 971),
          ('had', 969),
          ('his', 941),
          ('he', 894),
          ('have', 806),
          ('at', 805),
          ('by', 734),
          ('is', 732),
          ('Elinor', 680),
          ('on', 675),
          ("'s", 644),
          ('all', 640),
          ('him', 632),
          ('so', 616),
          ('but', 597),
          ('which', 592),
          ('could', 568),
          ('!', 560),
          ('Marianne', 558),
          ('my', 550),
          ('from', 527),
          ('Mrs.', 523),
          ('would', 507),
          ('very', 492),
          ('no', 488),
          ('their', 463),
          ('them', 460)]
```

## EXCERCISE 3

**A. imports necessary modules,**

**B. opens the text files and reads in the content as text strings,**

In [38]:

```
with open("jane_austen.txt") as fn:
    nov=fn.read()
print(nov)
```

```
[Emma by Jane Austen 1816]

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 caresses; and her place had been supplied
by an excellent woman as governess, who had fallen little short
of a mother in affection.
```

In [39]:

```
tokenizer = nltk.tokenize.WhitespaceTokenizer()
tok = tokenizer.tokenize(nov)
tok
```

Out[39]:

```
['[Emma',
 'by',
 'Jane',
 'Austen',
 '1816]',
 'VOLUME',
 'I',
 'CHAPTER',
 'I',
 'Emma',
 'Woodhouse,',
 'handsome,',
 'clever,',
 'and',
 'rich,',
 'with',
 'a',
 'comfortable',
 'home',
In [40]: b2 = list(nltk.bigrams(tok))
b2fd = nltk.FreqDist(b2)
b2fd
```

```
Out[40]: FreqDist({('of', 'the'): 1409, ('to', 'be'): 1333, ('in', 'the'): 1086, ('had',
         'been'): 668, ('to', 'the'): 645, ('of', 'her'): 601, ('could', 'not'): 573,
         ('I', 'am'): 569, ('she', 'had'): 548, ('it', 'was'): 546, ...})
```

```
In [41]: import re
         from collections import Counter
```

```
In [42]: words = re.findall(r'so+ \w+',open('jane_austen.txt').read())
         ab = Counter(zip(words))
         print(ab)
```

```
Counter({('so much',): 201, ('so very',): 102, ('so well',): 59, ('so man
y',): 54, ('so long',): 50, ('so little',): 44, ('so far',): 40, ('so I',): 2
9, ('so soon',): 23, ('so good',): 20, ('so often',): 16, ('so kind',): 14,
('so great',): 14, ('so it',): 14, ('so entirely',): 11, ('so happy',): 11,
('so you',): 11, ('so near',): 11, ('so to',): 10, ('so anxious',): 10, ('so
easily',): 9, ('so she',): 9, ('so glad',): 9, ('so fond',): 8, ('so ill',):
8, ('so strong',): 8, ('so bad',): 7, ('so as',): 7, ('so lately',): 7, ('so
miserable',): 7, ('so young',): 7, ('so totally',): 6, ('so truly',): 6, ('so
short',): 6, ('so few',): 6, ('so that',): 6, ('so particularly',): 6, ('so f
ull',): 6, ('so large',): 6, ('so extremely',): 6, ('so cheerful',): 6, ('so
pleasantly',): 5, ('so interesting',): 5, ('so completely',): 5, ('so fas
t',): 5, ('so obliging',): 5, ('so lovely',): 5, ('so at',): 5, ('so suddenl
y',): 5, ('so agreeable',): 5, ('so dear',): 4, ('so proper',): 4, ('so bus
y',): 4, ('so forth',): 4, ('so warmly',): 4, ('so charming',): 4, ('so wit
h',): 4, ('so deceived',): 4, ('so odd',): 4, ('so pleased',): 4, ('so deligh
ted',): 4, ('so happened',): 4, ('so thoroughly',): 4, ('so sudden',): 4, ('s
o on',): 4, ('so liberal',): 4, ('so attentive',): 4, ('so he',): 4, ('so sor
ry',): 4, ('so shocked',): 4, ('so wretched',): 4, ('so highly',): 4, ('so de
termined',): 4, ('so does',): 4, ('so unfeeling',): 4, ('so steady',): 4, ('s
```

### C. builds the following objects, a_ for Austen:

### 1. a_toks: word tokens, all in lowercase

In [43]:
```
tokenizer = nltk.tokenize.WhitespaceTokenizer()
a_toks = tokenizer.tokenize(nov.lower())
a_toks
```

Out[43]:
```
['[emma',
 'by',
 'jane',
 'austen',
 '1816]',
 'volume',
 'i',
 'chapter',
 'i',
 'emma',
 'woodhouse,',
 'handsome,',
 'clever,',
 'and',
 'rich,',
 'with',
 'a',
 'comfortable',
 'home',
```

**2. a_tokfd: word frequency distribution**

In [44]:
```
a_tokfd = FreqDist(a_toks)
a_tokfd
```

Out[44]:
```
FreqDist({'the': 12497, 'to': 11875, 'and': 10444, 'of': 10264, 'a': 6664, 'wa
s': 5363, 'in': 5343, 'i': 5261, 'her': 5238, 'she': 4787, ...})
```

**3. a_bigrams: word bigrams, cast as a list**

In [45]:
```
a_bigrams = list(nltk.bigrams(a_toks))
a_bigrams
```

Out[45]:
```
[('[emma', 'by'),
 ('by', 'jane'),
 ('jane', 'austen'),
 ('austen', '1816]'),
 ('1816]', 'volume'),
 ('volume', 'i'),
 ('i', 'chapter'),
 ('chapter', 'i'),
 ('i', 'emma'),
 ('emma', 'woodhouse,'),
 ('woodhouse,', 'handsome,'),
 ('handsome,', 'clever,'),
 ('clever,', 'and'),
 ('and', 'rich,'),
 ('rich,', 'with'),
 ('with', 'a'),
 ('a', 'comfortable'),
 ('comfortable', 'home'),
 ('home', 'and'),
```

### 4. a_bigramfd: bigram frequency distribution

In [46]:
```
a_bigramfd = nltk.FreqDist(a_bigrams)
a_bigramfd
```

Out[46]:
```
FreqDist({('of', 'the'): 1411, ('to', 'be'): 1342, ('in', 'the'): 1115, ('it',
'was'): 826, ('she', 'had'): 715, ('had', 'been'): 669, ('to', 'the'): 650, ('s
he', 'was'): 648, ('of', 'her'): 601, ('could', 'not'): 576, ...})
```

### 5. a_bigramcfd: bigram (w1, w2) conditional frequency distribution ("CFD"),where w1 is construed as the condition and w2 the outcome

In [47]:
```
from nltk.probability import ConditionalFreqDist
from nltk.tokenize import word_tokenize
```

In [48]:
```
a_bigramcfd = ConditionalFreqDist()
```

In [49]:
```
for word in a_toks:
    condition = len(word)
    a_bigramcfd[condition][word] += 1
```

In [50]:
```
a_bigramcfd
```

Out[50]:
```
<ConditionalFreqDist with 30 conditions>
```

### D. pickles the bigram CFDs (conditional frequency distributions) using the highest binary protocol: name the file as austen_bigramcfd.pkl.

In [51]:
```
with open('austen_bigramcfd.pkl', 'ab') as handle:
    pkl.dump(a_bigramcfd,handle)
```

### E. answers the following questions by exploring the objects

### 1. How many word tokens and types are there? what is its size

In [52]:
```
len(a_toks)
```

Out[52]:
```
360148
```

### 2. What are the top 20 most frequent words and their counts?. Draw chart using Matplotlib's plot() method.

In [53]:
```
ws=a_tokfd.most_common(20)
n = dict(ws)
n
```

Out[53]:
```
{'the': 12497,
 'to': 11875,
 'and': 10444,
 'of': 10264,
 'a': 6664,
 'was': 5363,
 'in': 5343,
 'i': 5261,
 'her': 5238,
 'she': 4787,
 'not': 4107,
 'be': 4035,
 'it': 3941,
 'had': 3729,
 'that': 3715,
 'he': 3544,
 'as': 3407,
 'for': 3113,
 'you': 2896,
 'his': 2761}
```

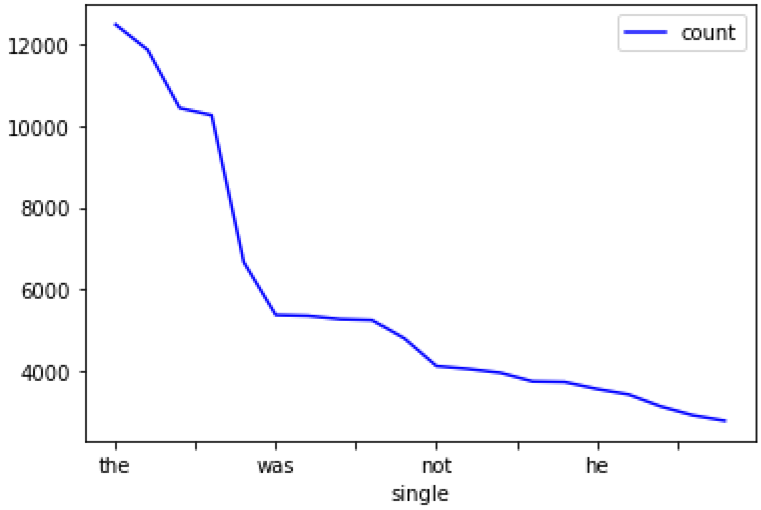
In [54]:
```
df = pd.DataFrame(list(n.items()))
df.columns = ['single','count']
df
```

Out[54]:

| | single | count |
|---|---|---|
| **0** | the | 12497 |
| **1** | to | 11875 |
| **2** | and | 10444 |
| **3** | of | 10264 |
| **4** | a | 6664 |
| **5** | was | 5363 |
| **6** | in | 5343 |
| **7** | i | 5261 |
| **8** | her | 5238 |
| **9** | she | 4787 |
| **10** | not | 4107 |
| **11** | be | 4035 |
| **12** | it | 3941 |
| **13** | had | 3729 |
| **14** | that | 3715 |
| **15** | he | 3544 |
| **16** | as | 3407 |
| **17** | for | 3113 |
| **18** | you | 2896 |
| **19** | his | 2761 |

In [55]:

```
df.plot(kind='line',x='single',y='count',color='blue')
plt.show()
```



### 4. What are the top 20 most frequent word bigrams and their counts, omitting bigrams that contain stopwords?

In [56]:

```
v=a_bigramfd.most_common(20)
m = dict(v)
m
```

Out[56]:

```
{('of', 'the'): 1411,
 ('to', 'be'): 1342,
 ('in', 'the'): 1115,
 ('it', 'was'): 826,
 ('she', 'had'): 715,
 ('had', 'been'): 669,
 ('to', 'the'): 650,
 ('she', 'was'): 648,
 ('of', 'her'): 601,
 ('could', 'not'): 576,
 ('i', 'am'): 570,
 ('he', 'had'): 513,
 ('have', 'been'): 495,
 ('of', 'his'): 493,
 ('and', 'the'): 474,
 ('i', 'have'): 474,
 ('he', 'was'): 442,
 ('it', 'is'): 419,
 ('in', 'a'): 408,
 ('for', 'the'): 406}
```

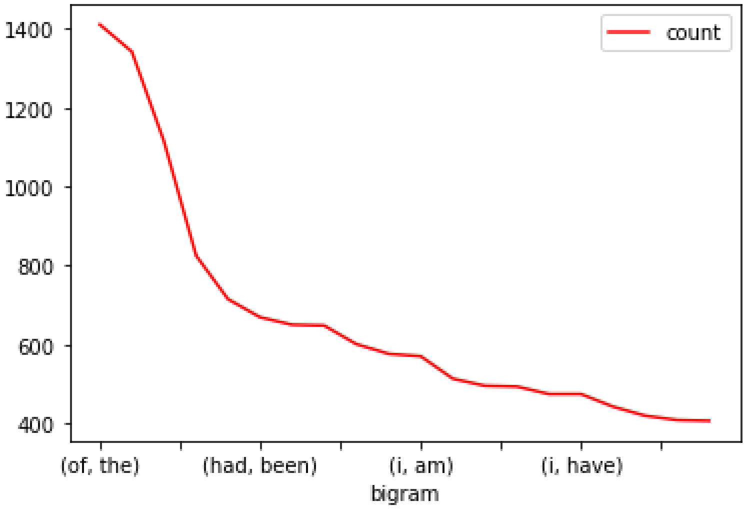
In [57]:
```
df2 = pd.DataFrame(list(m.items()))
df2.columns = ['bigram','count']
df2
```

Out[57]:

| | bigram | count |
|---|---|---|
| **0** | (of, the) | 1411 |
| **1** | (to, be) | 1342 |
| **2** | (in, the) | 1115 |
| **3** | (it, was) | 826 |
| **4** | (she, had) | 715 |
| **5** | (had, been) | 669 |
| **6** | (to, the) | 650 |
| **7** | (she, was) | 648 |
| **8** | (of, her) | 601 |
| **9** | (could, not) | 576 |
| **10** | (i, am) | 570 |
| **11** | (he, had) | 513 |
| **12** | (have, been) | 495 |
| **13** | (of, his) | 493 |
| **14** | (and, the) | 474 |
| **15** | (i, have) | 474 |
| **16** | (he, was) | 442 |
| **17** | (it, is) | 419 |
| **18** | (in, a) | 408 |
| **19** | (for, the) | 406 |

### 5. What are the top 20 most frequent word bigrams and their counts, omitting bigrams that contain stopwords?. Draw chart using Matplotlib’s plot() method.

In [58]:
```
df2.plot(kind='line',x='bigram',y='count',color='red')
plt.show()
```



**6. How many times does the word 'so' occur? What are their relative frequency against the corpus size (= total # of tokens)?**

In [59]:
```
so_count=a_tokfd['so']
print(so_count)

tot=len(a_tokfd)
print(tot)

rel_freq = so_count/tot
rel_freq
```

```
1746
26903
```

Out[59]:
```
0.06489982529829387
```

**7. What are the top 20 'so-initial' bigrams (bigrams that have the word “so” as the first word) and their counts?**

In [60]:
```
ab.most_common(20)
```

Out[60]:
```
[(('so much',), 201),
 (('so very',), 102),
 (('so well',), 59),
 (('so many',), 54),
 (('so long',), 50),
 (('so little',), 44),
 (('so far',), 40),
 (('so I',), 29),
 (('so soon',), 23),
 (('so good',), 20),
 (('so often',), 16),
 (('so kind',), 14),
 (('so great',), 14),
 (('so it',), 14),
 (('so entirely',), 11),
 (('so happy',), 11),
 (('so you',), 11),
 (('so near',), 11),
 (('so to',), 10),
 (('so anxious',), 10)]
```

### 8. Given the word 'so' as the current word, what is the probability of getting 'much' as the next word?

In [61]:
```
ab_dict = dict(ab)
ab_dict
```

Out[61]:
```
{('so unperceived',): 1,
 ('so far',): 40,
 ('so obliged',): 2,
 ('so mild',): 1,
 ('so much',): 201,
 ('so to',): 10,
 ('so well',): 59,
 ('so happily',): 3,
 ('so many',): 54,
 ('so long',): 50,
 ('so perfectly',): 3,
 ('so constantly',): 2,
 ('so entirely',): 11,
 ('so comfortably',): 1,
 ('so very',): 102,
 ('so kind',): 14,
 ('so avowed',): 1,
 ('so dear',): 4,
 ('so deservedly',): 1,
```

In [62]:
```
tot_occ=len(ab_dict)
tot_occ
```

Out[62]:
```
584
```

In [63]:

```
for i , j in ab_dict.items():
    if i == ('so much',):
        print(i,j)
        print(j/tot_occ)
```

```
('so much',) 201
0.3441780821917808
```

### 9. Given the word 'so' as the current word, what is the probability of getting 'will' as the next word?

In [64]:

```
for i , j in ab_dict.items():
    if i == ('so will',):
        print(i,j)
        print(j/tot_occ)
```

```
('so will',) 1
0.0017123287671232876
```