Brody_Vogel_NLTK_HW_2

September 16, 2018

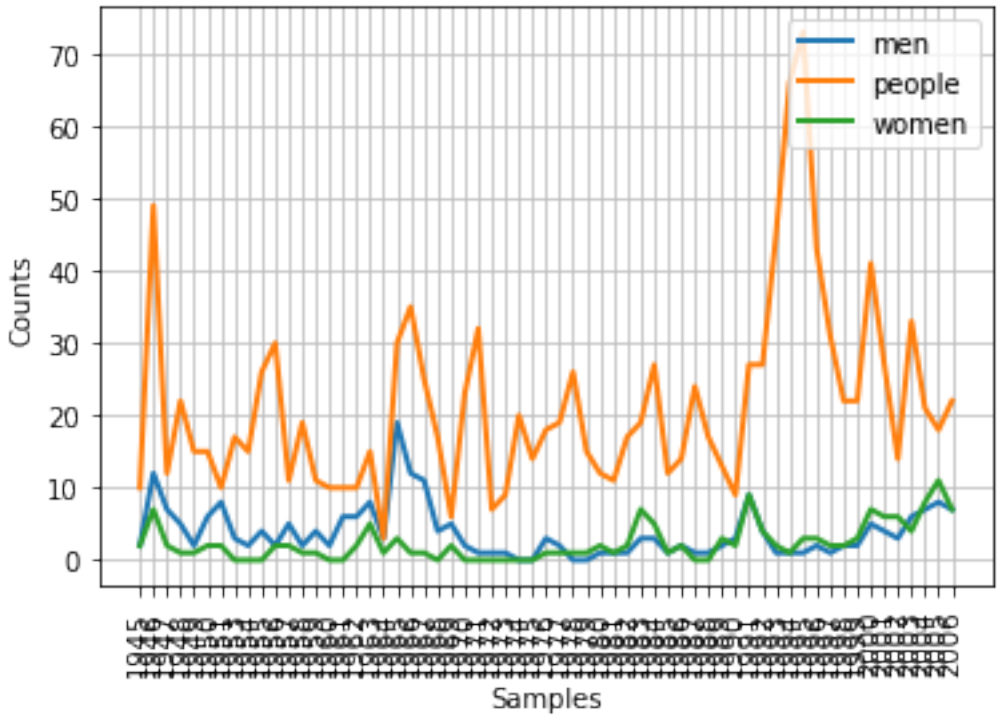
In [53]:
```
# Brody Vogel NLP Assignment 2 #
import nltk
from nltk.book import *
from nltk.corpus import state_union
import matplotlib.pyplot as plt
```

In [27]:
```
#4
# It looks like the use of 'people' has slightly
# increased with time. The word 'men' used to be used
# more frequently than 'women', but that appears to
# have changed in the 70s or 80s, although neither is
# used nearly as often as the word 'people'.

print('Name  ' + 'men ' + 'women ' + 'people')
for fileid in state_union.fileids():
    men = len([w for w in state_union.words(fileid) if w.lower() == 'men'])
    women = len([w for w in state_union.words(fileid) if w.lower() == 'women'])
    people = len([w for w in state_union.words(fileid) if w.lower() == 'people'])
    print(str(fileid[:4]) + ': ' + str(men) + ' ' + str(women) + ' ' + str(people))

cfd = nltk.ConditionalFreqDist(
            (word, fileid[:4])
            for fileid in state_union.fileids()
            for w in state_union.words(fileid)
            for word in ['men', 'women', 'people']
            if w.lower() == word)
cfd.plot()
```

```
Name  men women people
1945: 2 2 10
1946: 12 7 49
1947: 7 2 12
1948: 5 1 22
1949: 2 1 15
1950: 6 2 15
1951: 8 2 10
1953: 3 0 17
```

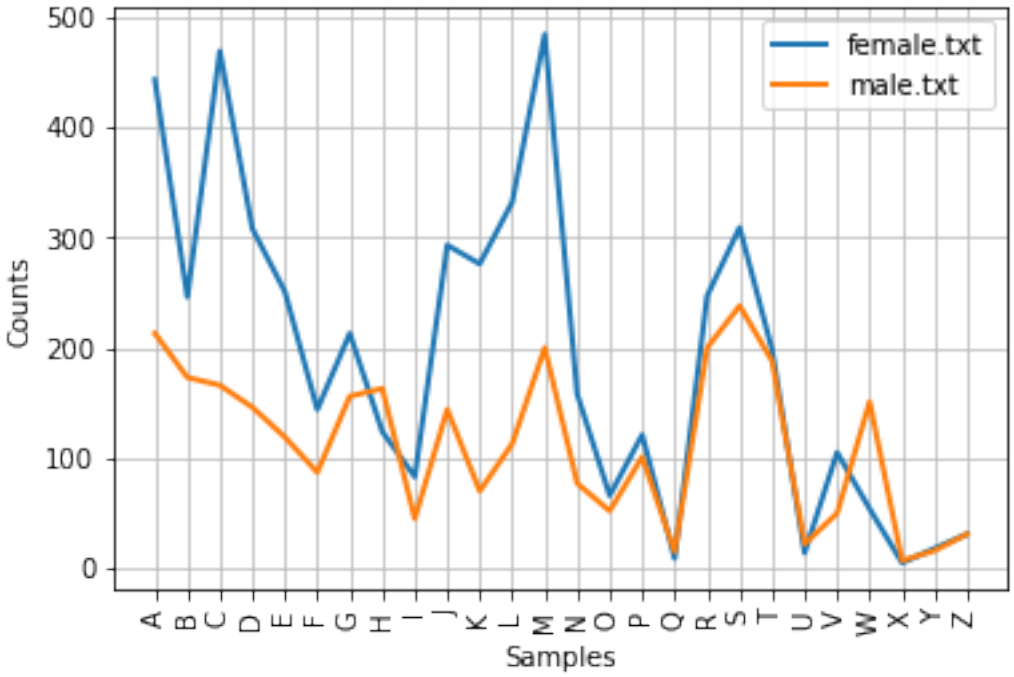
1954: 2 0 15
1955: 4 0 26
1956: 2 2 30
1957: 5 2 11
1958: 2 1 19
1959: 4 1 11
1960: 2 0 10
1961: 6 0 10
1962: 6 2 10
1963: 0 0 3
1963: 8 5 12
1964: 3 1 3
1965: 7 0 16
1965: 12 3 14
1966: 12 1 35
1967: 11 1 25
1968: 4 0 17
1969: 5 2 6
1970: 2 0 23
1971: 1 0 32
1972: 1 0 7
1973: 1 0 9
1974: 0 0 20
1975: 0 0 14
1976: 3 1 18
1977: 2 1 19
1978: 0 1 26
1979: 0 1 15
1980: 1 2 12
1981: 1 1 11
1982: 1 2 17
1983: 3 7 19
1984: 3 5 27
1985: 1 1 12
1986: 2 2 14
1987: 1 0 24
1988: 1 0 17
1989: 2 3 13
1990: 3 2 9
1991: 2 2 14
1991: 7 7 13
1992: 4 4 27
1993: 1 2 45
1994: 1 1 66
1995: 1 3 73
1996: 2 3 43
1997: 1 2 31
1998: 2 2 22

```
1999: 2 3 22
2000: 5 7 41
2001: 3 3 15
2001: 1 3 12
2002: 3 6 14
2003: 6 4 33
2004: 7 8 21
2005: 8 11 18
2006: 7 7 22
```



In [29]:
```
#8
# It looks like, in general, there are
# more female than male names in the corpus.
# That said, it looks like
# names that begin with letters at the end
# of the alphabet tend to be male.

names = nltk.corpus.names

cfd = nltk.ConditionalFreqDist(
            (fileid, name[0])
            for fileid in names.fileids()
```

```
                for name in names.words(fileid))
cfd.plot()
```



In [33]:
```
#16
# It seems the lowest diversity score belongs
# to learned. This is not what I would've
# expected. I can
# only guess that the learned texts are textbooks,
# which are very long and therefore must reuse words
# frequently.

from nltk.corpus import brown

print('Genre: Lexical Diversity Score')
for genre in brown.categories():
    print(str(genre) + ': ' + str(len(set(brown.words(categories = genre)))/len(brown
```

```
Genre: Lexical Diversity Score
adventure: 0.1279743878169075
belles_lettres: 0.10642071451679992
editorial: 0.16054152327770924
fiction: 0.1358194136199042
government: 0.11667641228232811
hobbies: 0.14493897625842492
humor: 0.23125144042406084
```

```
learned: 0.09268890745953554
lore: 0.13148804612915801
mystery: 0.12212912592488936
news: 0.14314696580941583
religion: 0.1617553745018909
reviews: 0.21192020440251572
romance: 0.12070492131044529
science_fiction: 0.22342778161713892
```

```
In [42]: #17
         # ( I also threw out punctuation )
         # below are the 50 most common words
         # - without stop words - from the
         # Book of Genesis

         from nltk.corpus import stopwords

         def no_stop_words_and_common(text):
             cleaned = [w.lower() for w in text if w.lower() not in stopwords.words('english')]
             fdist = nltk.FreqDist(cleaned)
             return(fdist.most_common(50))

         no_stop_words_and_common(text3)
```

```
Out[42]: [('unto', 598),
          ('said', 477),
          ('thou', 284),
          ('thy', 278),
          ('shall', 259),
          ('thee', 257),
          ('god', 231),
          ('lord', 207),
          ('father', 198),
          ('land', 184),
          ('jacob', 179),
          ('came', 177),
          ('joseph', 157),
          ('son', 152),
          ('sons', 142),
          ('upon', 140),
          ('abraham', 129),
          ('behold', 118),
          ('man', 115),
          ('earth', 112),
          ('went', 110),
          ('wife', 104),
          ('years', 102),
```

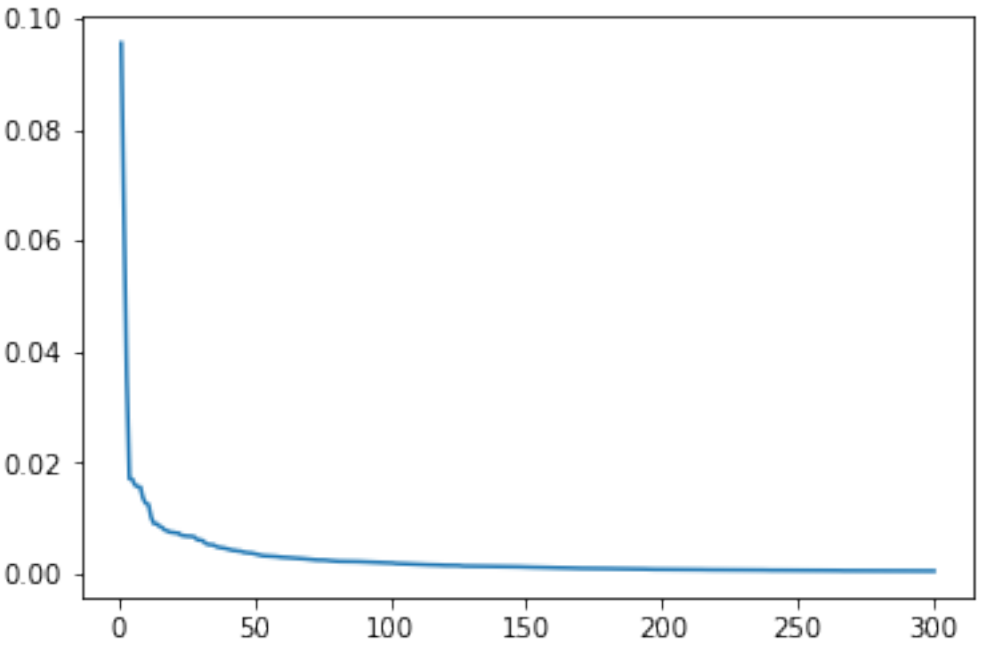
```
 ('name', 100),
 ('called', 98),
 ('ye', 96),
 ('let', 93),
 ('us', 93),
 ('every', 91),
 ('brother', 91),
 ('pharaoh', 90),
 ('also', 83),
 ('hand', 82),
 ('pass', 82),
 ('house', 82),
 ('took', 81),
 ('hath', 80),
 ('brethren', 80),
 ('saying', 79),
 ('go', 78),
 ('isaac', 77),
 ('come', 75),
 ('shalt', 74),
 ('egypt', 74),
 ('esau', 74),
 ('day', 72),
 ('made', 72),
 ('one', 70),
 ('give', 67),
 ('begat', 67)]
```

In [51]:
```
#18
# ( I also threw out punctuation )
# below are the 50 most common bigrams
# - without stop words - from
# the Book of Genesis

from nltk.corpus import stopwords

def no_stop_words_and_frequent_bigrams(text):
    bigs = bigrams(text)
    bigs = [bigram for bigram in bigs if bigram[0] not in stopwords.words('english')
                                                        and bigram[0].isalpha() == Tru
                                                        and bigram[1] not in stopwords
                                                        and bigram[1].isalpha() == Tru

    fdist = nltk.FreqDist(bigs)
    return(fdist.most_common(50))

no_stop_words_and_frequent_bigrams(text3)
```

Out[51]:
```
[(('said', 'unto'), 178),
 (('And', 'Jacob'), 56),
```

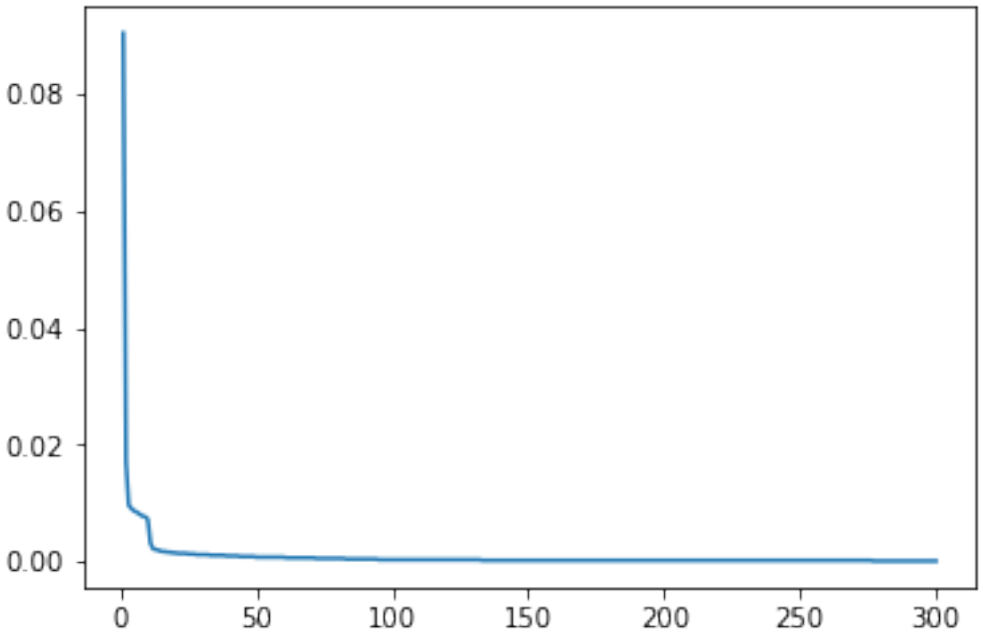
```
(('And', 'God'), 51),
(('And', 'Joseph'), 51),
(('I', 'pray'), 45),
(('thou', 'shalt'), 43),
(('thou', 'hast'), 39),
(('pray', 'thee'), 38),
(('And', 'Abraham'), 36),
(('thy', 'seed'), 35),
(('LORD', 'God'), 29),
(('unto', 'thee'), 29),
(('spake', 'unto'), 28),
(('God', 'said'), 26),
(('ye', 'shall'), 23),
(('And', 'Isaac'), 23),
(('And', 'I'), 22),
(('God', 'hath'), 21),
(('thou', 'art'), 21),
(('years', 'old'), 21),
(('unto', 'Joseph'), 21),
(('shalt', 'thou'), 19),
(('thy', 'father'), 19),
(('LORD', 'said'), 18),
(('And', 'Pharaoh'), 18),
(('And', 'Laban'), 18),
(('begat', 'sons'), 17),
(('unto', 'Jacob'), 17),
(('shall', 'come'), 16),
(('let', 'us'), 16),
(('unto', 'Abraham'), 16),
(('Jacob', 'said'), 16),
(('Joseph', 'said'), 16),
(('seven', 'years'), 15),
(('I', 'may'), 15),
(('thy', 'brother'), 14),
(('LORD', 'hath'), 14),
(('unto', 'us'), 14),
(('thy', 'servant'), 14),
(('every', 'man'), 13),
(('And', 'Abram'), 13),
(('unto', 'Pharaoh'), 13),
(('every', 'living'), 12),
(('thy', 'wife'), 12),
(('I', 'know'), 12),
(('shall', 'I'), 12),
(('Now', 'therefore'), 12),
(('unto', 'thy'), 12),
(('And', 'Esau'), 12),
(('And', 'Israel'), 12)]
```

In [76]:
```
#23a
# ( Truncated to the 300 most common
# words ; log scale didn't change much )
# At the left extreme end of the line,
# it looks like we have stopwords -
# words that're used very frequently.
# At the right extreme end, it looks
# like we have words that're used once.
# Zipf's Law seems to roughly hold
# in my plot for The Book of Genesis;
# looking at the 50th and 150th most
# common words, it does look like the
# former has about 3 times as many
# occurrences as the latter.

import pylab
import math

def freq_vs_rank(text):
    lowered = [w.lower() for w in text if w.isalpha() == True]
    fdist = nltk.FreqDist(lowered)
    pylab.plot([z + 1 for z in range(0, 300)], sorted([fdist.freq(x) for x in fdist],
    #pylab.plot([z + 1 for z in range(0, 300)], sorted([math.log(fdist.freq(x)) for x

freq_vs_rank(text3)
```



In [80]:
```
#23b
# ( Again truncated to the 300 most
# common words )
# The plots look very similar,
# even with these nonsensical words.
# Again, it looks like the 50th
# most common words occurs about three
# times as often as the 150th. Thus, it
# would appear that
# Zipf's Law holds in both of these experiments.

import random

brody_string = ''
for x in range(0, 100000):
    brody_string = brody_string + random.choice('brodyvogel ')

brody_tokens = brody_string.split(' ')

fdist = nltk.FreqDist(brody_tokens)

pylab.plot([z + 1 for z in range(0, 300)], sorted([fdist.freq(x) for x in fdist], reve
```

Out[80]:
```
[<matplotlib.lines.Line2D at 0x1180ae278>]
```



In [84]:
```
#27
# It looks like verbs have the most average
```

```
        # senses. I wouldn't have guessed that,
        # but intuitively it does seem
        # like most verbs can be used as other
        # parts of speech.

        from nltk.corpus import wordnet as wn

        def avg_poly(speech_part):
            total_senses = 0
            for lemma in wn.all_lemma_names(pos = speech_part):
                total_senses += len(wn.synsets(lemma, speech_part))
            print(speech_part + ': ' + str(total_senses/len(list(wn.all_lemma_names(pos = spe

        for part in ['n', 'v', 'a', 'r']:
            avg_poly(part)
```

```
n: 1.2610825311125826
v: 2.1865729898516784
a: 1.406536617160948
r: 1.2532916759651864
```