

Taller 9

Métodos Computacionales para Políticas Públicas - UROSARIO

Entrega: viernes 26-abr-2019 11:59 PM

****Francisco Monsalve****

francisco.monsalve@urosario.edu.co

Instrucciones:

- Guarde una copia de este *Jupyter Notebook* en su computador, idealmente en una carpeta destinada al material del curso.
- Modifique el nombre del archivo del *notebook*, agregando al final un guión inferior y su nombre y apellido, separados estos últimos por otro guión inferior. Por ejemplo, mi *notebook* se llamaría: mcpp_taller9_santiago_matallana
- Marque el *notebook* con su nombre y e-mail en el bloque verde arriba. Reemplace el texto "[Su nombre acá]" con su nombre y apellido. Similar para su e-mail.
- Desarrolle la totalidad del taller sobre este *notebook*, insertando las celdas que sea necesario debajo de cada pregunta. Haga buen uso de las celdas para código y de las celdas tipo *markdown* según el caso.
- Recuerde salvar periódicamente sus avances.
- Cuando termine el taller:
 1. Descárguelo en PDF. Si tiene algún problema con la conversión, descárguelo en HTML.
 2. Suba todos los archivos a su repositorio en GitHub, en una carpeta destinada exclusivamente para este taller, antes de la fecha y hora límites.

NLTK Book (<http://www.nltk.org/book/>), Exercises:

- Chapter 1: 22, 26, 28
- Chapter 2: 2, 4, 11

```
In [107]: %matplotlib inline
import matplotlib.pyplot as plt
plt.rcParams["figure.figsize"] = [18.0, 8.0]
```

```
In [2]: import nltk
```

```
In [3]: from nltk.book import *
```

```
*** Introductory Examples for the NLTK Book ***
Loading text1, ..., text9 and sent1, ..., sent9
Type the name of the text or sentence to view it.
Type: 'texts()' or 'sents()' to list the materials.
text1: Moby Dick by Herman Melville 1851
text2: Sense and Sensibility by Jane Austen 1811
text3: The Book of Genesis
text4: Inaugural Address Corpus
text5: Chat Corpus
text6: Monty Python and the Holy Grail
text7: Wall Street Journal
text8: Personals Corpus
text9: The Man Who Was Thursday by G . K . Chesterton 1908
```

22. Find all the four-letter words in the Chat Corpus (text5). With the help of a frequency distribution (FreqDist), show these words in decreasing order of frequency.

```
In [4]: four_letter = [w for w in text5 if len(w) == 4]
four_letter
```

```
Out[4]: ['left',
         'with',
         'this',
         'name',
         'PART',
         'well',
         'NICK',
         'name',
         'U121',
         'golf',
         'clap',
         'JOIN',
         'that',
         'nice',
         'JOIN',
         'PART',
         'golf',
         'clap',
         'fuck',
         'U121',
         'PART',
         'PART',
         'clap',
         'your',
         'PART',
         'dont',
         'even',
         'know',
         'what',
         'that',
         'that',
         'chat',
         'JOIN',
         'drew',
         'cast',
         'PART',
```

'sexy',
'U115',
'JOIN',
'PART',
'drew',
'girl',
'with',
'legs',
'hope',
'draw',
'PART',
'head',
'legs',
'JOIN',
'JOIN',
'good',
'JOIN',
'PART',
'take',
'have',
'docs',
'Slip',
'away',
'Fade',
'away',
'Days',
'away',
'feel',
'have',
'back',
'U115',
'U129',
'U115',
'chat',
'with',
'PART',
'JOIN',
'JOIN',
'fast',

'U116',
'bowl',
'bong',
'JOIN',
'well',
'glad',
'hard',
'from',
'here',
'back',
'PART',
'PART',
'JOIN',
'U121',
'name',
'hard',
'very',
'fire',
'from',
'here',
'JOIN',
'PART',
'itch',
'JOIN',
'U133',
'ogan',
'male',
'JOIN',
'JOIN',
'show',
'will',
'talk',
'PART',
'haha',
'opps',
'JOIN',
'PART',
'U115',
'nice',

'warm',
'guys',
'with',
'cams',
'play',
'sits',
'JOIN',
'JOIN',
'guyz',
'chat',
'U126',
'PART',
'chat',
'PART',
'gooo',
'sure',
'U126',
'JOIN',
'what',
'feel',
'like',
'room',
'yeee',
'JOIN',
'want',
'pics',
'look',
'U139',
'PART',
'PART',
'JOIN',
'here',
'JOIN',
'PART',
'JOIN',
'U139',
'PART',
'JOIN',
'U138',

'U139',
'make',
'U139',
'that',
'U126',
'late',
'lmao',
'ahah',
'PART',
'U121',
'U121',
'does',
'like',
'that',
'guys',
'male',
'JOIN',
'U139',
'well',
'what',
'yeah',
'know',
'U136',
'hell',
'with',
'U139',
'U101',
'like',
'when',
'plan',
'PART',
'JOIN',
'gold',
'jeep',
'make',
'sure',
'nice',
'ring',
'U115',

'isnt',
'that',
'U136',
'hell',
'have',
'have',
'doin',
'U139',
'U121',
'many',
'Just',
'fine',
'that',
'like',
'PART',
'hiya',
'room',
'lmao',
'doin',
'Deep',
'Show',
'that',
'love',
'that',
'Turn',
'take',
'Hand',
'just',
'even',
'look',
'hang',
'PART',
'that',
'such',
'word',
'U141',
'hear',
'!!!!',
'PART',

'JOIN',
'PART',
'deaf',
'here',
'dont',
'U115',
'U115',
'....',
'hugs',
'chat',
'with',
'baby',
'Only',
'U121',
'U121',
'PART',
'have',
'away',
'from',
'U121',
'what',
'read',
'here',
'with',
'JOIN',
'read',
'have',
'here',
'JOIN',
'want',
'chat',
'talk',
'U121',
'JOIN',
'U121',
'VBox',
'PART',
'take',
'that',

'JOIN',
'PART',
'hate',
'when',
'U121',
'U115',
'lmao',
'PART',
'your',
'know',
'what',
'your',
'what',
'JOIN',
'love',
'more',
'than',
'ELSE',
'serg',
'well',
'most',
'love',
'JOIN',
'know',
'that',
'what',
'lmao',
'well',
'have',
'eyes',
'lmao',
'know',
'JOIN',
'girl',
'jerk',
'kids',
'guys',
'type',
'much',

'shut',
'fuck',
'girl',
'nice',
'shut',
'fuck',
'PART',
'dont',
'want',
'JOIN',
'want',
'U115',
'what',
'miss',
'much',
'work',
'nice',
'U116',
'PART',
'PART',
'hey',
'U148',
'hate',
'boys',
'JOIN',
'U148',
'hate',
'what',
'PART',
'hate',
'U121',
'fuck',
'your',
'ugly',
'JOIN',
'bein',
'PART',
'What',
'U115',

'whys',
'that',
'deep',
'U121',
'what',
'JOIN',
'tape',
'Your',
'sexs',
'best',
'phil',
'said',
'ugly',
'PART',
'date',
'feel',
'your',
'they',
'form',
'PART',
'sits',
'JOIN',
'sits',
'with',
'hmp',
'hate',
'does',
'that',
'mean',
'want',
'room',
'this',
'been',
'PART',
'JOIN',
'U115',
'U116',
'your',
'here',

'talk',
'wait',
'that',
'perv',
'lets',
'hope',
'U121',
'PART',
'U115',
'!!!!',
'lets',
'chat',
'JOIN',
'rule',
'land',
'wont',
'then',
'find',
'need',
'this',
'HUGE',
'perv',
'that',
'deal',
'???'',
'JOIN',
'shit',
'hell',
'lmao',
'PART',
'hell',
'JOIN',
'here',
'guys',
'have',
'U121',
'JOIN',
'U155',
'only',

'Poor',
'U121',
'love',
'pick',
'much',
'that',
'PART',
'PART',
'sits',
'with',
'U121',
'nads',
'JOIN',
'from',
'pick',
'your',
'pick',
'your',
'nose',
'pick',
'your',
'nose',
'JOIN',
'face',
'with',
'PART',
'U115',
'owww',
'PART',
'JOIN',
'U116',
'PART',
'does',
'want',
'talk',
'head',
'gags',
'even',
'U121',

'neck',
'Meep',
'U115',
'LAst',
'time',
'that',
'wash',
'your',
'dude',
'gets',
'JOIN',
'U121',
'dang',
'just',
"pm's",
'that',
'1.99',
'....',
'yeah',
'nice',
'neck',
'U115',
'like',
'shut',
'free',
'JOIN',
'goes',
'wash',
'lmao',
'Lies',
'lmao',
'U115',
'lick',
'very',
'lmao',
'U115',
'ummm',
'U109',
'dont',

'dead',
'more',
'than',
'call',
'just',
'case',
'dead',
'good',
'neck',
'talk',
'what',
'ummm',
'else',
'wont',
'bite',
'U115',
'yeah',
'wait',
'yeah',
'PART',
'your',
'want',
'have',
'sexy',
'bite',
'lmao',
'call',
'have',
'free',
'call',
'mins',
'JOIN',
'nite',
'lool',
'know',
'that',
'kina',
'give',
'away',

'then',
'room',
'call',
'yeah',
'U155',
'PART',
'U115',
'more',
'U115',
'guys',
'baby',
'U109',
'fuck',
'case',
'know',
'were',
'girl',
'JOIN',
'baby',
'what',
'U109',
'guys',
'chat',
'have',
'have',
'sext',
'piff',
'dont',
'talk',
'read',
'dang',
'lazy',
'dont',
'read',
'PART',
'JOIN',
'mean',
'fine',
'....',

'busy',
'work',
'okay',
'dont',
'talk',
'calm',
'down',
'busy',
'busy',
'want',
'chat',
'arms',
'kids',
'name',
'PART',
'sits',
'down',
'eats',
'JOIN',
'hugs',
'want',
'U121',
'near',
'just',
'PART',
'JOIN',
'PART',
'hell',
'yeah',
'U115',
'near',
'near',
'good',
'smax',
'JOIN',
'haha',
'only',
'>:->',
'near',

'PART',
'piff',
'Vvil',
'JOIN',
'free',
'wont',
'cold',
'U121',
'cell',
'runs',
'thru',
'back',
'hair',
'eyes',
'neck',
'yeah',
'caps',
'PART',
'JOIN',
'PART',
'PART',
'U165',
'jump',
'U165',
'baby',
'here',
'over',
'your',
'good',
'PART',
'that',
'left',
'room',
'este',
'U115',
'will',
'PART',
'U121',
'U165',

'lmao',
'PART',
'PART',
'very',
'guys',
'wana',
'chat',
'chik',
'from',
'mean',
'chat',
'well',
'PART',
'that',
'dont',
'shit',
'U165',
'U165',
'left',
'room',
'with',
'your',
'JOIN',
'U115',
'what',
'list',
'wish',
'cmon',
'U128',
'nice',
'JOIN',
'list',
'PART',
'list',
'U115',
'good',
'lmao',
'U128',
'hehe',

'hows',
'bout',
'good',
'hear',
'U165',
'have',
'JOIN',
'good',
'PART',
'JOIN',
'wats',
'they',
'PART',
'piff',
'aint',
'know',
'shut',
'much',
'good',
'PART',
'JOIN',
'JOIN',
'yeah',
'lost',
'like',
'same',
'well',
'work',
'what',
'Boyz',
'rock',
'what',
'hehe',
'went',
'back',
'some',
'then',
'came',
'back',

'home',
'PART',
'what',
'they',
'coat',
'nice',
'read',
'many',
'nice',
'hehe',
'lmao',
'even',
'JOIN',
'well',
'talk',
'nite',
'what',
'very',
'time',
'What',
'kind',
'....',
'nite',
'PART',
'Eyes',
'Dawn',
'last',
'song',
'LIVE',
'cool',
'good',
'nite',
'mauh',
'nite',
'mike',
'keep',
'must',
'girl',
'seem',

'pick',
'else',
'....',
'take',
'your',
'your',
'JOIN',
'lmao',
'just',
'days',
'late',
'with',
'room',
'JOIN',
'PART',
'good',
'ques',
'lmao',
'JOIN',
'that',
'like',
'dont',
'quit',
'what',
'your',
'4.20',
'PART',
'like',
'mine',
'over',
'cali',
'good',
'this',
'year',
'NICK',
'whoa',
'have',
'have',
'have',

'what',
'boys',
'gosh',
'that',
'ruff',
'what',
'PART',
'hell',
'rock',
'roll',
'PART',
'with',
'like',
'that',
'nope',
'....',
'rest',
'rock',
'roll',
'....',
'....',
'sing',
'from',
'kids',
'....',
'mame',
'nada',
'cali',
'here',
'that',
'cool',
'kids',
'cool',
'JOIN',
'with',
'from',
'said',
'alot',
'JOIN',


```
'year',  
'band',  
'JOIN',  
'NICK',  
'cool',  
'nice',  
'here',  
'hair',  
'hard',  
'what',  
'type',  
'does',  
'your',  
'band',  
'play',  
'hair',  
'yeah',  
'what',  
'doin',  
'....',  
'with',  
'hand',  
'what',  
'JOIN',  
'room',  
'....',  
'sexy',  
'PART',  
'dumb',  
'they',  
'wont',  
'chat',  
'with',  
'lmao',  
'damn',  
'what',  
'orgy',  
'orgy',  
'lmao',
```

'what',
'word',
'some',
'easy',
'PART',
'JOIN',
'back',
'orgy',
'lmao',
'PART',
'lets',
'play',
'room',
'PART',
'JOIN',
'JOIN',
'were',
'damn',
'PART',
'good',
'call',
'what',
'like',
'just',
'late',
'date',
'know',
'push',
'PART',
'lose',
'name',
'shit',
'head',
'long',
'time',
'said',
'shit',
'lost',
'baby',

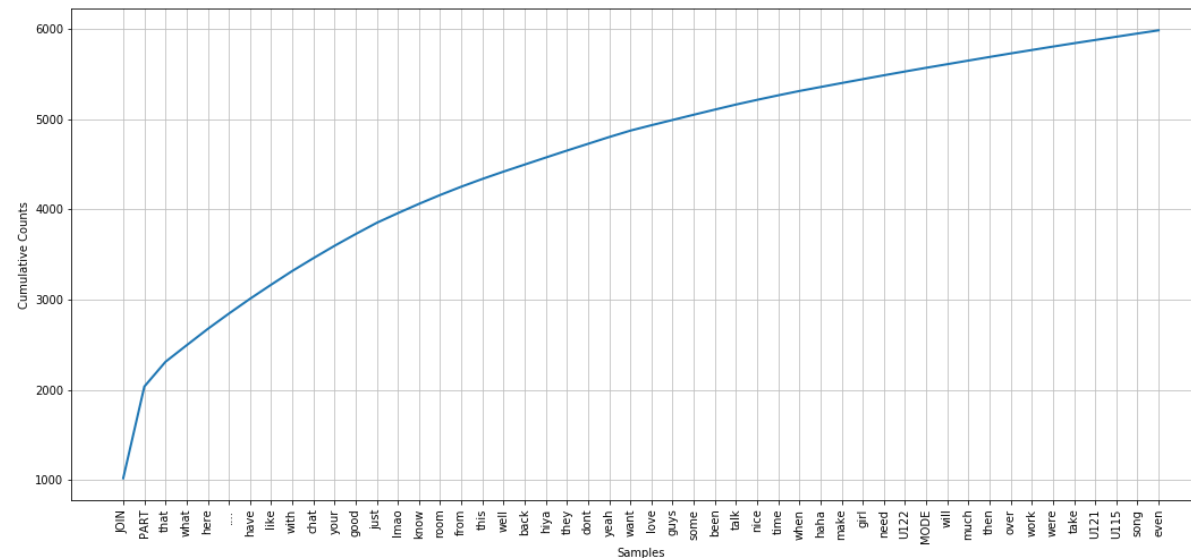
'then',
'JOIN',
'PART',
'JOIN',
'What',
'sure',
'JOIN',
'sure',
'lmao',
'this',
'room',
'....',
'....',
'your',
'....',
'lmao',
'that',
'yeah',
'have',
'many',
'lmao',
'sexy',
'stay',
'keep',
'lmao',
'with',
'hair',
'like',
'that',
'....',
'down',
'door',
'prob',
'....',
'hair',
'lmao',
'JOIN',
'sexy',
'just',

```
'this',  
'life',  
'just',  
'PART',  
'room',  
'with',  
'hair',  
'like',  
'what',  
'from',  
'here',  
'said',  
'wild',  
'....',  
'even',  
'from',  
'JOIN',  
'cool',  
'sexy',  
'JOIN',  
'sexy',  
'only',  
'none',  
'sexy',  
'whew',  
'sexy',  
'hell',  
'have',  
...]
```

```
In [8]: fdist5 = FreqDist(four_letter)  
fdist5
```

```
Out[8]: FreqDist({'JOIN': 1021, 'PART': 1016, 'that': 274, 'what': 183, 'here':  
181, '....': 170, 'have': 164, 'like': 156, 'with': 152, 'chat': 142,  
...})
```

```
In [108]: fdist5.plot(50, cumulative=True)
```



26.What does the following Python code do?

```
sum(len(w) for w in text1)
```

Can you use it to work out the average word length of a text?

El código está sumando el tamaño de cada palabra, por lo que al final va a dar el tamaño total de la suma de todas las palabras. Una vez ejecutado el código, se puede encontrar el tamaño promedio de la palabra de un texto si se divide por el número total de palabras.

Una aclaración importante, es que esta operación cuenta tokens, no únicamente palabras. Para tener el promedio de palabras únicamente, se debe utilizar la función set.

```
In [11]: total_length = sum(len(w) for w in text1)
total_length
```

```
Out[11]: 999044
```

```
In [22]: total_words = len(text1)
total_words
```

```
Out[22]: 260819
```

```
In [68]: avg_wrd_ln = (sum(len(w) for w in text1))/(len(text1))
avg_wrd_ln
```

```
Out[68]: 3.830411128023649
```

```
In [23]: def avg_word_lenght(text):
total_length = sum(len(w) for w in text)
total_words = len(text)
avg_word = (total_length)/(total_words)
return avg_word
```

```
In [24]: avg_word_lenght(text1)
```

```
Out[24]: 3.830411128023649
```

```
In [25]: avg_word_lenght(text2)
```

```
Out[25]: 3.881371136350794
```

En promedio, el tamaño de las palabras del texto1 son de alrededor de 4 caracteres (3.8)

28. Define a function percent(word, text) that calculates how often a given word occurs in a text, and expresses the result as a percentage.

```
In [72]: def percent(word, text):
fdist = FreqDist(text)
w = fdist[word]
l = len(text)
word_percentage = 100 * w / l
```

```
return word_percentage
```

```
In [73]: percent("whale", text1)
```

```
Out[73]: 0.3473673313677301
```

```
In [74]: percent("God", text3)
```

```
Out[74]: 0.5160396747386292
```

2. Use the corpus module to explore austen-persuasion.txt. How many word tokens does this book have? How many word types?

```
In [75]: import nltk  
nltk.corpus.gutenberg.fileids()
```

```
Out[75]: ['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']
```

```
In [77]: from nltk.corpus import gutenberg
```

```
gutenberg.fileids()
```

```
Out[77]: ['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']
```

```
In [80]: austen_p = gutenberg.words('austen-persuasion.txt')  
         #word tokens  
         len(austen_p)
```

```
Out[80]: 98171
```

```
In [81]: #word types  
         len(set(austen_p))
```

```
Out[81]: 6132
```

4. Read in the texts of the State of the Union addresses, using the `state_union` corpus reader. Count occurrences of men, women, and people in each document. What has happened to the usage of these words over time?


```
In [84]: from nltk.corpus import state_union
state_union.fileids()
```

```
Out[84]: ['1945-Truman.txt',
'1946-Truman.txt',
'1947-Truman.txt',
'1948-Truman.txt',
'1949-Truman.txt',
'1950-Truman.txt',
'1951-Truman.txt',
'1953-Eisenhower.txt',
'1954-Eisenhower.txt',
'1955-Eisenhower.txt',
'1956-Eisenhower.txt',
'1957-Eisenhower.txt',
'1958-Eisenhower.txt',
'1959-Eisenhower.txt',
'1960-Eisenhower.txt',
'1961-Kennedy.txt',
'1962-Kennedy.txt',
'1963-Johnson.txt',
'1963-Kennedy.txt',
'1964-Johnson.txt',
'1965-Johnson-1.txt',
'1965-Johnson-2.txt',
'1966-Johnson.txt',
'1967-Johnson.txt',
'1968-Johnson.txt',
'1969-Johnson.txt',
'1970-Nixon.txt',
'1971-Nixon.txt',
'1972-Nixon.txt',
'1973-Nixon.txt',
'1974-Nixon.txt',
'1975-Ford.txt',
'1976-Ford.txt',
'1977-Ford.txt',
'1978-Carter.txt',
'1979-Carter.txt',
'1980-Carter.txt',
```

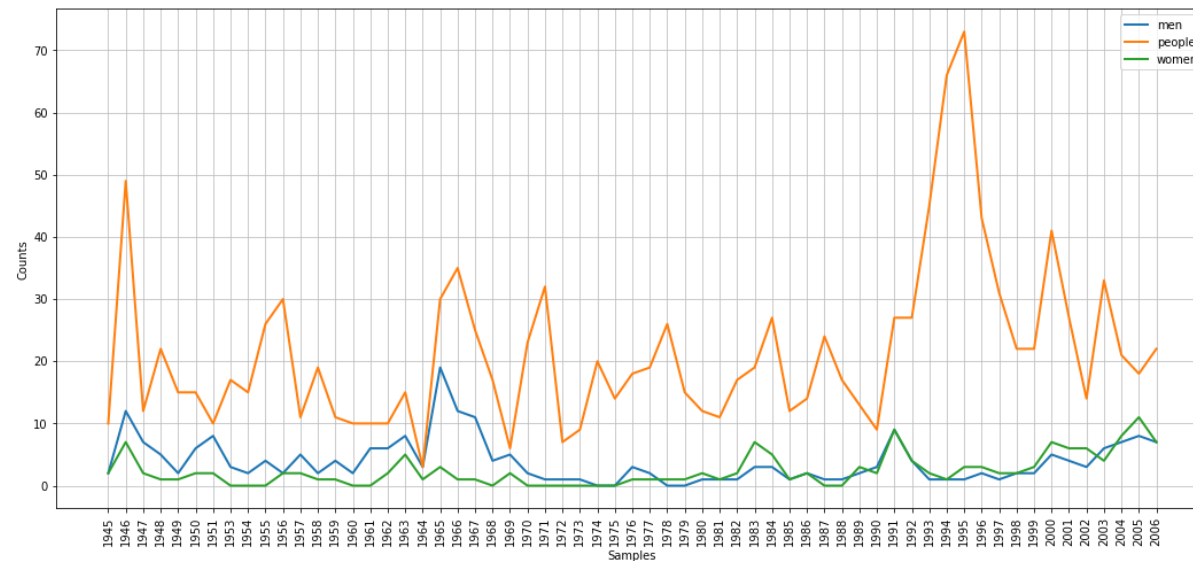
```
'1981-Reagan.txt',  
'1982-Reagan.txt',  
'1983-Reagan.txt',  
'1984-Reagan.txt',  
'1985-Reagan.txt',  
'1986-Reagan.txt',  
'1987-Reagan.txt',  
'1988-Reagan.txt',  
'1989-Bush.txt',  
'1990-Bush.txt',  
'1991-Bush-1.txt',  
'1991-Bush-2.txt',  
'1992-Bush.txt',  
'1993-Clinton.txt',  
'1994-Clinton.txt',  
'1995-Clinton.txt',  
'1996-Clinton.txt',  
'1997-Clinton.txt',  
'1998-Clinton.txt',  
'1999-Clinton.txt',  
'2000-Clinton.txt',  
'2001-GWBush-1.txt',  
'2001-GWBush-2.txt',  
'2002-GWBush.txt',  
'2003-GWBush.txt',  
'2004-GWBush.txt',  
'2005-GWBush.txt',  
'2006-GWBush.txt']
```

```
In [92]: len(state_union.fileids())
```

```
Out[92]: 65
```

```
In [109]: cfd = nltk.ConditionalFreqDist(  
            (interest, fileid[:4])  
            for fileid in state_union.fileids()  
            for w in state_union.words(fileid)  
            for interest in ['men', 'women', 'people'])
```

```
if w.lower() == interest)
cfd.plot()
```



11. Investigate the table of modal distributions and look for other patterns. Try to explain them in terms of your own impressionistic understanding of the different genres. Can you find other closed classes of words that exhibit significant differences across different genres?

```
In [101]: from nltk.corpus import brown
brown.categories()
```

```
Out[101]: ['adventure',
            'belles_lettres',
            'editorial',
            'fiction',
            'government',
            'hobbies',
            'humor',
            'learned',
```

```
'lore',
'mystery',
'news',
'religion',
'reviews',
'romance',
'science_fiction']
```

```
In [102]: cfd = nltk.ConditionalFreqDist(
            (genre, word)
            for genre in brown.categories()
            for word in brown.words(categories=genre))
genres = ['news', 'religion', 'hobbies', 'science_fiction', 'romance',
          'humor']
modals = ['can', 'could', 'may', 'might', 'must', 'will']
cfd.tabulate(conditions=genres, samples=modals)
```

	can	could	may	might	must	will
news	93	86	66	38	50	389
religion	82	59	78	12	54	71
hobbies	268	58	131	22	83	264
science_fiction	16	49	4	12	8	16
romance	74	193	11	51	45	43
humor	16	30	8	8	9	13

```
In [103]: cfd = nltk.ConditionalFreqDist(
            (genre, word)
            for genre in brown.categories()
            for word in brown.words(categories=genre))
genres = ['news', 'religion', 'hobbies', 'science_fiction', 'romance',
          'humor', 'editorial', 'belles_lettres', 'government']
pronouns = ['I', 'you', 'he', 'she', 'it', 'we', 'they']
cfd.tabulate(conditions=genres, samples=pronouns)
```

	I	you	he	she	it	we	they
news	179	55	451	42	363	77	205
religion	155	100	137	10	264	176	115
hobbies	154	383	155	21	476	100	177
science_fiction	98	81	139	36	129	30	53

romance	951	456	702	496	573	78	168
humor	239	131	146	58	162	32	70
editorial	201	83	268	41	386	167	148
belles_lettres	845	188	1174	178	1059	398	488
government	97	74	120	0	218	112	92

```
In [106]: cfd = nltk.ConditionalFreqDist(
          (genre, word)
          for genre in brown.categories()
          for word in brown.words(categories=genre))
genres = ['news', 'religion', 'hobbies', 'science_fiction', 'romance',
          'humor', 'editorial', 'belles_lettres', 'government']
others = ['love', 'family', 'entertainment', 'important', "society"]
cfd.tabulate(conditions=genres, samples=others)
```

	love	family	entertainment	important
society				
news	3	41	7	13
12				
religion	13	20	2	17
10				
hobbies	6	25	2	40
3				
science_fiction	3	1	0	2
0				
romance	32	23	0	4
1				
humor	4	6	1	7
2				
editorial	13	14	0	16
2				
belles_lettres	68	54	2	61
78				
government	1	10	7	44
3				

