Taller 9

Métodos Computacionales para Políticas Públicas - URosario

Entrega: viernes 24-abr-2020 11:59 PM

[Andrés Ramírez Vela]

[andrese.ramirez@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/ (http://www.nltk.org/book/)), ejercicios:

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

In [10]:
```
import nltk
import matplotlib.pyplot as plt
plt.rcParams["figure.figsize"] = [18.0, 10.0]
```

In [2]:
```
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
```

SOLUCIÓN PUNTOS DEL CAPÍTULO UNO

1. 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 [3]:
```
fdist = FreqDist(w for w in text5 if len(w) == 4)
print(list(fdist))
```

```
['JOIN', 'PART', 'that', 'what', 'here', '....', 'have', 'like', 'wit
h', 'chat', 'your', 'good', 'just', 'lmao', 'know', 'room', 'from', 'th
is', 'well', 'back', 'hiya', 'they', 'dont', 'yeah', 'want', 'love', 'g
uys', 'some', 'been', 'talk', 'nice', 'time', 'when', 'haha', 'make',
'girl', 'need', 'U122', 'MODE', 'will', 'much', 'then', 'over', 'work',
'were', 'take', 'U121', 'U115', 'song', 'even', 'does', 'seen', 'U156',
'U105', 'more', 'damn', 'only', 'come', 'hell', 'long', 'them', 'name',
'tell', 'away', 'sure', 'look', 'baby', 'call', 'play', 'U110', 'U114',
'NICK', 'down', 'cool', 'sexy', 'many', 'hate', 'said', 'last', 'ever',
'hear', 'life', 'live', 'feel', 'very', 'mean', 'give', 'same', 'must',
'stop', 'LMAO', '!!!!', 'hugs', 'What', 'find', 'cant', 'left', '????',
'shit', 'nite', 'busy', 'hair', 'lost', 'U104', 'fine', 'real', 'game',
'fuck', 'sits', 'eyes', 'lets', 'heya', 'kill', 'read', 'shut', 'wait',
'goes', 'keep', 'true', 'pick', 'free', 'else', 'near', 'nope', 'U168',
'hope', 'head', 'male', 'than', 'gets', 'cold', 'hehe', 'bout', 'stay',
'used', 'awww', 'told', 'This', 'U102', 'doin', 'kids', 'perv', 'wont',
'face', 'home', 'year', 'babe', 'into', 'yall', '.. .', 'U119', 'U107',
'hard', 'show', 'U101', 'once', 'Well', 'help', 'mind', 'Yeah', 'week',
'Liam', 'U132', 'pics', 'such', 'type', 'best', 'neck', 'dang', 'dead',
'runs', 'aint', 'rock', 'days', 'mine', 'book', 'crap', 'soon', 'care',
'full', 'kiss', 'hour', 'nick', 'sick', '; ..', 'hmmm', 'U139', 'word',
'heyy', 'case', 'wana', 'hows', 'went', 'lady', 'blue', 'says', 'suck',
'made', 'wife', 'sang', 'U144', 'fast', 'rule', 'dude', 'okay', 'alot',
'hand', 'took', 'wear', 'Hiya', 'kick', 'ahhh', 'dear', 'That', 'U108',
'U169', 'U129', 'U116', 'most', 'thru', 'U165', 'list', 'seem', 'sing',
'next', 'done', 'ride', 'comp', 'main', '))))', 'goin', 'U520', 'pink',
'poor', 'gone', 'oops', 'knew', '<---', 'ball', 'send', 'Song', 'blah',
'They', 'part', 'U103', 'U120', 'Last', 'whos', 'food', 'U142', 'sock',
'U197', 'legs', 'fire', 'warm', 'late', 'hang', 'miss', 'boys', 'land',
'nose', 'lick', 'caps', 'wish', 'U128', 'came', 'cali', 'roll', 'easy',
'lose', 'When', 'soul', 'luck', 'also', 'kool', 'fall', 'boss', 'beer',
'ohhh', '####', 'wall', 'Have', 'meet', 'till', 'feet', 'xbox', 'idea',
'heck', 'joke', 'fool', 'felt', 'yoko', 'meds', 'both', 'Lime', 'glad',
'U133', 'U126', 'jerk', 'ugly', 'date', 'ummm', 'quit', 'rest', 'door',
'none', 'self', 'pass', 'line', 'cute', 'holy', 'hook', 'Like', 'each',
'open', 'high', 'ouch', 'evil', 'fart', 'grrr', 'pain', 'pfft', 'sigh',
'shes', 'ROOM', ',,,,', 'lord', 'mmmm', 'ones', 'huge', 'woot', 'shot',
'team', 'ways', 'beat', 'kent', 'U130', 'U196', 'U219', 'turn', 'lame',
'U123', 'U154', 'U988', 'puff', 'U146', 'U989', 'U117', 'U819', 'U820',
'clap', 'itch', 'guyz', 'U136', 'gold', 'ring', 'isnt', 'U141', 'Only',
'U148', 'Your', 'deal', 'wash', 'U109', 'piff', 'jump', 'band', 'orgy',
'slap', 'soft', 'bend', 'toss', 'amen', 'rain', 'deop', 'roof', '((((',
'CHAT', 'ahem', 'hola', 'butt', 'imma', 'town', 'hawt', '2006', 'Elev',
'Wind', 'AKDT', 'lead', 'DING', 'note', 'gawd', 'half', 'mary', 'ello',
'hick', 'wine', 'hiii', 'bare', 'vote', 'Same', 'wack', 'snow', 'hurt',
'move', 'road', 'walk', 'yawn', 'hail', 'nana', 'U106', 'hump', 'elle',
'yada', 'tune', 'hank', 'slow', 'rubs', 'skin', 'died', 'U145', 'swim',
'U163', 'army', 'THAT', 'wazz', 'toes', 'U153', 'golf', 'drew', 'cast',
'Days', 'opps', 'U138', 'plan', 'Just', 'deaf', 'deep', 'phil', 'hmph',
'U155', 'Poor', 'Lies', 'bite', 'mins', 'eats', '>:->', 'cell', 'cmon',
'wats', 'kind', 'mike', 'whoa', 'dumb', 'park', 'Sure', 'Come', 'O.k.',
'mama', 'Nice', 'hold', 'ohio', 'whip', 'twin', 'burp', 'blew', 'temp',
'corn', 'pool', 'cash', 'ears', 'From', 'porn', 'heal', 'Dang', 'ciao',
'DOES', 'typo', 'Stop', 'eric', 'Drew', 'sore', 'Live', 'High', 'hits',
'KoOL', 'past', 'Love', 'meat', '!!!.', 'argh', 'limp', 'rent', 'cars',
'Tell', 'shop', 'U172', 'five', 'sell', '<<<<', 'city', 'yard', 'grrl',
'chip', 'bear', 'foot', 'uses', 'DONT', 'sort', 'lies', 'whud', 'hott',
```

```
'Down', 'Lets', 'club', 'adds', 'Here', 'born', 'wOOt', 'area', '?!?!',
'Ohio', 'U112', 'humm', 'newp', 'gays', 'zone', 'hint', 'spin', 'ewww',
'pies', 'doll', 'drop', 'gimp', 'spot', 'ages', 'clue', 'mass', 'Ummm',
'Gosh', 'flow', 'kewl', 'hall', 'haze', '1996', 'John', 'john', 'sooo',
'cost', 'trip', 'babi', 'rich', 'U100', 'n9ne', 'Ahhh', '??!!', 'U111',
'moon', 'STOP', 'any1', 'yeas', 'wooo', '<333', 'tick', 'tock', 'WITH',
'FROM', 'side', 'Heyy', 'howz', "ex's", 'Cool', 'U170', 'U175', 'root',
'tyvm', 'luvs', 'fits', 'rofl', 'sand', 'ltns', 'flaw', 'aunt', 'lawl',
'Okay', 'HAVE', 'NONE', 'YOUR', 'Lmao', 'Tisk', 'U190', 'tisk', 'draw',
'docs', 'Slip', 'Fade', 'bowl', 'bong', 'ogan', 'cams', 'gooo', 'yeee',
'ahah', 'jeep', 'Deep', 'Show', 'Turn', 'Hand', 'VBox', 'ELSE', 'serg',
'bein', 'whys', 'tape', 'sexs', 'form', 'HUGE', 'nads', 'owww', 'gags',
'Meep', 'LAst', "pm's", '1.99', 'lool', 'kina', 'sext', 'lazy', 'calm',
'arms', 'smax', 'VVil', 'este', 'chik', 'Boyz', 'coat', 'Eyes', 'Dawn',
'LIVE', 'mauh', 'ques', '4.20', 'gosh', 'ruff', 'mame', 'nada', 'push',
'prob', 'wild', 'whew', 'dark', 'waht', 'test', 'boot', 'hiom', 'HAHA',
'dman', 'jail', 'cops', 'hogs', 'peek', 'MORE', 'TIME', 'loud', 'o.k.',
'Sexy', 'Ctrl', 'hots', 'Need', 'frst', '1200', 'crop', 'bomb', 'Pour',
'pour', 'Swim', 'Hard', 'eeek', 'tjhe', '10th', 'heee', 'peel', 'fock',
'Kold', 'exit', 'kold', '3:45', 'MRIs', 'buff', 'plus', 'tory', 'knee',
'OOPS', 'oooh', 'lala', 'fake', 'ssid', 'poot', 'poop', 'bird', 'plow',
'thnx', 'card', 'Hugs', 'Lord', 'uyes', 'benz', '<~~~', 'disc', 'LONG',
'Been', 'Will', 'bloe', 'blow', 'hooo', 'thje', 'Jess', 'term', 'Tina',
'ooer', 'HALO', 'Awww', 'anal', 'Drop', 'dojn', 'wubs', 'mkay', 'spat',
'gees', 'hawT', 'yes.', 'puts', 'fish', 'size', '39.3', '1980', '64.8',
'syck', 'tere', 'U542', 'sent', '45.5', '98.5', '1299', '1900', '1930',
'Werd', 'Rofl', 'mode', 'nawt', 'sign', 'woof', 'sum1', 'ghet', 'brad',
'offa', 'Dood', 'out.', 'LOUD', 'sink', 'FINE', 'cums', 'loss', 'Life',
'Damn', 'wrap', 'hide', "PM's", 'Talk', 'okey', 'worl', 'Hold', 'cepn',
'lots', 'Mary', 'nawp', 'addy', 'lake', 'slip', 'mite', 'wood', 'orta',
'wins', 'ebay', 'coem', 'giva', '1.98', 'ally', 'Judy', 'cyas', 'shup',
'tooo', "pm'n", 'choc', 'wher', 'whoo', 'dint', 'tend', 'menu', 'lust',
'nods', 'NAME', 'kept', 'scuk', 'raed', 'Then', 'bugs', 'nerd', 'Hill',
'Evil', 'saME', '2Pac', 'Time', 'pimp', 'haaa', '98.6', "it's", 'Mono',
'mono', 'Bone', 'Hero', 'Came', '.op.', 'Hott', 'Joey', 'Jane', 'span',
'wore', 'QUIT', 'pasa', 'barn', 'Kick', 'feat', 'Back', 'dork', 'laid',
'Home', 'herd', 'Born', 'Away', 'Tide', 'jush', 'Cute', 'GrlZ', 'lung',
'SOME', 'Lion', 'brat', ':o *', 'MUAH', 'fawk', 'dust', 'Help', 'seth',
'Heya', 'bone', 'abou', 'tthe', 'Even', 'herE', 'Hail', 'halo', 'pork',
'1cos', "yw's", 'mark', 'dotn', 'PMSL', 'pmsl', 'gift', 'outs', 'Paul',
'outa', 'York', 'Care', 'Chat', 'fear', 'dies', 'givs', 'bust', 'xmas',
'enuf', 'LoVe', 'eeww', 'dick', 'fair', 'lyin', 'lois', 'cuss', 'LATE',
'THEY', 'GOOD', 'rape', 'geez', 'tart', 'hgey', 'caan', 'lol.', 'Elle',
'nude', 'allo', 'yesh', 'wind', 'Reub', '!???', 'heat', 'kmph', 'pope',
'yess', '!...', 'duet', 'wuts', 'west', 'quiz', 'scar', 'Girl', 'pair',
'Rang', 'rang', 'bell', 'dawg', 'febe', 'Prof', 'Kewl', 'jude', 'Yoko',
'seee', 'whou', 'idnt', 'perk', 'http', '2DAY', 'yell', 'mang', 'SSRI',
'cure', 'wean', 'post', 'anti', 'noth', 'tall', 'pray', 'weed', 'icky',
'Rick', 'spit', 'lube', 'mami', 'east', '18ST', 'seat', 'cock', 'SExy',
'otay', 'firs', 'site', 'U113', 'dump', 'toop', 'four', 'U118', 'sets',
'asss', 'paid', 'Iowa', 'Teck', '"...', 'jeff', 'crib', 'drug', 'cook',
'9:10', 'ladz', 'aime', 'hong', 'kong', 'Oops', 'tits', 'gret', 'guns',
'inch', 'sean', 'howl', 'Take', 'z-ro', 'U137', 'Haha', '1985', 'slam',
'pine', 'puke', 'waaa', 'urls', 'star', 'Save', 'teck', 'Room', 'sori',
'Long', 'poem', 'jack', 'Rule', 'CAPS', 'junk', 'tips', 'rush', 'Nooo',
'Troy', 'tail', 'Seee', '6:38', 'dyed', 't he', 'beam', 'daft', 'twit',
'scum', 'U134', 'Type', 'WHOA', 'toke', 'ribs', 'Eggs', 'Wyte', 'moms',
```

```
'Over', 'West', 'Rock', 'goof', 'U143', 'able', 'vamp', 'Nope', 'Kent',
'ther', 'U147', 'TEXT', 'SIZE', 'gear', 'CALI', 'Matt', 'Rush', 'AWAY',
'NTMN', 'Kiss', 'U158', 'grea', 'Look', 'guts', 'wrek', 'Fort', '2:55',
'AKST', '4:03', 'wire', 'soda', 'gray', 'tlak', 'ltnc', "ok'd", 'sayn',
'evah', 'bike', 'hill', 'ohwa', 'caca', 'prep', 'pull', 'dirt', 'vent',
'100%', 'safe', 'dogs', 'bull', 'asks', 'Road', 'chit', 'grin', 'bred',
'rats', 'Sat.', 'samn', 'Phil', 'nuff', 'rose', 'Ruth', 'grew', 'mena',
'ROFL', 'lapd', 'surf', 'City', 'hazy', 'thot', 'acid', 'wide', 'keys',
'salt', 'mess', 'base', 'byes', "RN's", 'yout', 'numb', 'thah', 'mahn',
'King', 'TALK', 'GIRL', 'WHEN', 'HOTT', 'HERE', 'soup', '6:51', '9.53',
'Mine', 'vega', 'pigs', 'king', 'poof', 'Nova', 'mofo', 'Ohhh', 'Holy',
'sips', 'clay', 'None', 'Male', 'bacl', 'body', 'akon', 'yoll', 'boom',
'News', 'Maps', 'page', 'Tiff', 'Chop', 'DAMN', 'TYPR', 'poll', 'boed',
'Dude', 'Does', 'pwns', 'Very', 'Good', 'Food', 'sexi', 'bois', 'KNOW',
'GUYS', 'YALL', 'EVEN', 'SEEN', 'WILL', 'COME', 'FACE', 'JUST', 'Kids',
'6:41', 'bied', '6:53', 'U149', '7:45', 'Uhhh', 'tenn', 'pure', 'U164',
'U150', 'U181', 'gals', 'woah', 'ussy', 'tiff', 'Heys', "<3's", 'lisa',
'brwn', 'hurr', 'Were']
```

1. 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?

In [4]:
```
extension = sum(len(w) for w in text4)
palabras = len(text4)
promedio_palabras = extension / palabras
print(promedio_palabras)
```

```
4.380087718712658
```

El código serviría para contar el número de carácteres y se podría usar para sacar el promedio de extensión de las palabras de un texto.

1. 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 [5]:
```
def percent(word, text):
    return 100 * text.count(word) / len(text4)
print (str(percent("Syme", text9)) + '%')
```

```
0.3437986074487473%
```

SOLUCIÓN PUNTOS DEL CAPÍTULO DOS

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

In [6]:
```
from nltk.corpus import gutenberg
persuasion = gutenberg.words('austen-persuasion.txt')
print(len([word for word in persuasion if word.isalpha()]))
print(len(set(word.lower() for word in persuasion if word.isalpha())))
```

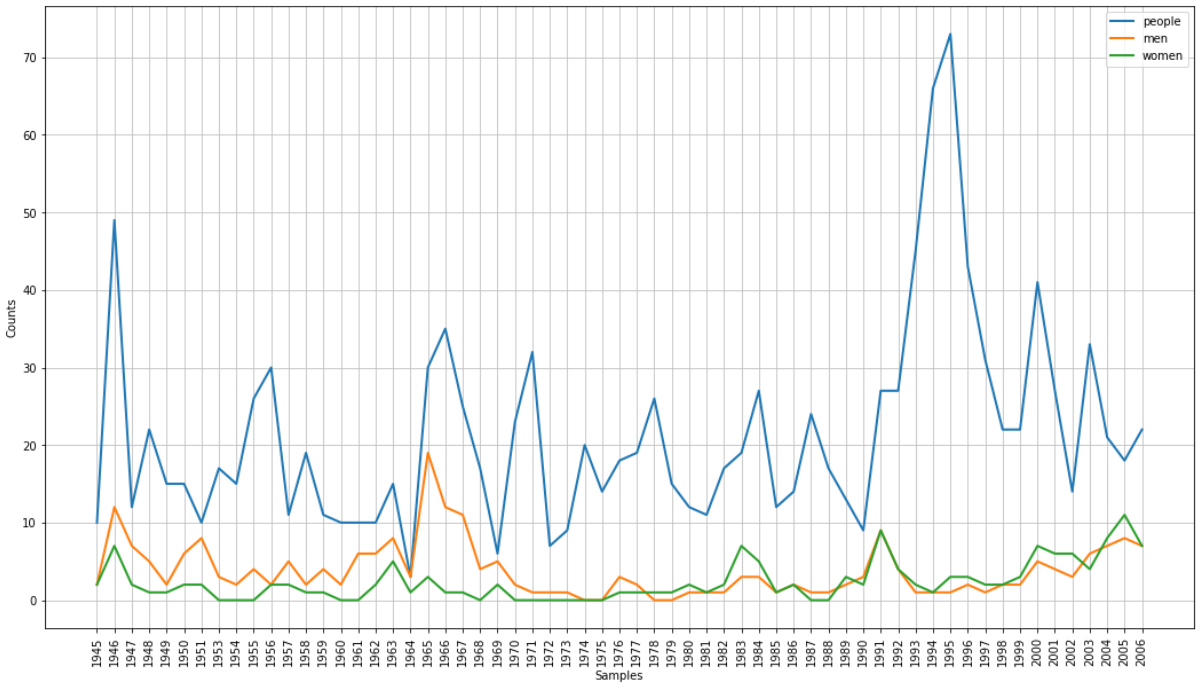
```
84121
5739
```

1. 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 [7]:
```
from nltk.corpus import state_union
state_union.fileids()
[fileid[:4] for fileid in state_union.fileids()]
cfd = nltk.ConditionalFreqDist((target, fileid[:4])
    for fileid in state_union.fileids()
    for w in state_union.words(fileid)
    for target in ['men', 'women', 'people']
    if w.lower() == target)
cfd.tabulate()
```

```
       1945 1946 1947 1948 1949 1950 1951 1953 1954 1955 1956 1957 1958
1959 1960 1961 1962 1963 1964 1965 1966 1967 1968 1969 1970 1971 1972 1
973 1974 1975 1976 1977 1978 1979 1980 1981 1982 1983 1984 1985 1986 19
87 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 200
1 2002 2003 2004 2005 2006
   men     2   12    7    5    2    6    8    3    2    4    2    5    2
4    2    6    6    8    3   19   12   11    4    5    2    1    1    1
0    0    3    2    0    0    1    1    1    3    3    1    2    1    1
2    3    9    4    1    1    1    2    1    2    2    5    4    3    6
7    8    7
people   10   49   12   22   15   15   10   17   15   26   30   11   19
11   10   10   10   15    3   30   35   25   17    6   23   32    7
9   20   14   18   19   26   15   12   11   17   19   27   12   14   24
17   13    9   27   27   45   66   73   43   31   22   22   41   27   1
4   33   21   18   22
 women    2    7    2    1    1    2    2    0    0    0    2    2    1
1    0    0    2    5    1    3    1    1    0    2    0    0    0    0
0    0    1    1    1    1    2    1    2    7    5    1    2    0    0
3    2    9    4    2    1    3    3    2    2    3    7    6    6    4
8   11    7
```

In [11]:
```
cfd.plot()
```



Out[11]: <matplotlib.axes._subplots.AxesSubplot at 0x1a13897cd0>

Al gráficar los datos obtenidos por la salida del modelo, se puede observar que "people" siempre ha sido usada a lo largo del tiempo, con picos pronunciados tral el final de la Segunda Guerra Mundial en el 45 y 46, pero sobretodo en los años 90. Para el caso de la palabra "men" es interesante el pico del 64 al 68 que se corresponde con los años más intensos de la intervención en la guerra de vietnam, siendo también un pico alto para la palabra "people". La palabre "women" tiene sus mayores picos a partir de la decada de los 90 manteniendo un ritmo sotenido de uso. AL igual que "men".

1. 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 [12]:
```
from nltk.corpus import brown
modals = ["can", "could", "may", "might", "must", "will"]
cfd = nltk.ConditionalFreqDist(
            (genre, word.lower())
            for genre in brown.categories()
            for word in brown.words(categories=genre))
genres = ["religion", "mystery", "science_fiction", "romance", "humor",
"fiction"]
cfd.tabulate(conditions=genres, samples=modals)
```

	can	could	may	might	must	will
religion	84	59	79	12	54	72
mystery	45	145	15	57	31	25
science_fiction	16	49	4	12	8	17
romance	79	195	11	51	46	49
humor	17	33	8	8	9	13
fiction	39	168	10	44	55	56

Es interesante observar que los modals que tienden a ser menos exactos se usan más en generos como el romance, misterio, fiction y ciencia ficción. Esto tal vez por el hecho de que se refieren a temas más de tipo incierto o que pudieron o pueden acontecer o que se dejan como una posiblidad. Es excepcional el caso del will en el sentido de que se usa en los generos ya mencionados pero en religion también, supongo por su caracter escatológico.

In [13]:
```
from nltk.corpus import brown
other = ["who", "what", "when", "where", "why", "how"]
cfd = nltk.ConditionalFreqDist(
            (genre, word.lower())
            for genre in brown.categories()
            for word in brown.words(categories=genre))
genres = ["religion", "mystery", "science_fiction", "romance", "humor",
"fiction"]
cfd.tabulate(conditions=genres, samples=other)
```

```
                  who  what  when where   why   how
       religion   102    86    68    21    20    28
        mystery    94   146   154    71    52    48
science_fiction    13    41    28    15     8    16
        romance    95   171   163    58    62    77
          humor    49    46    62    16    13    25
        fiction   112   186   192    89    42    89
```

In [14]:
```
from nltk.corpus import brown
other = ["love", "sad", "death", "woman", "man"]
cfd = nltk.ConditionalFreqDist(
            (genre, word.lower())
            for genre in brown.categories()
            for word in brown.words(categories=genre))
genres = ["religion", "mystery", "science_fiction", "romance", "humor",
"fiction"]
cfd.tabulate(conditions=genres, samples=other)
```

```
                  love   sad death woman   man
       religion     13     3    42     3    68
        mystery      7     4    18    32   107
science_fiction      4     3     2     4    18
        romance     36     6    12    34   100
          humor      5     0     1    10    23
        fiction     16     3    17    34   112
```

In [ ]: