RegEx_example

October 8, 2019

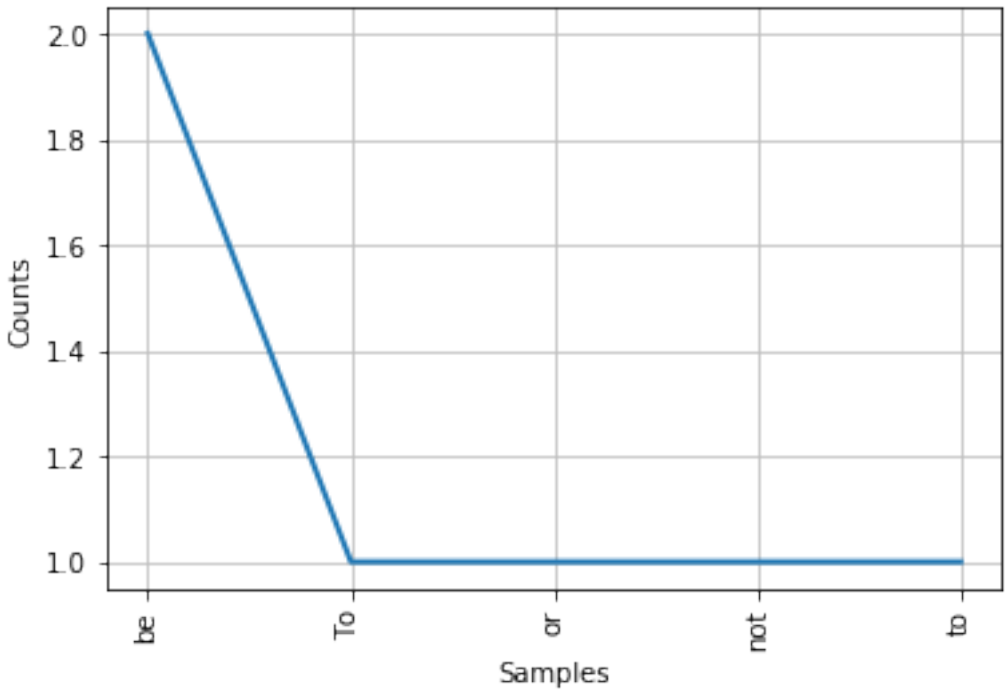
In [1]:
```
# -*- coding: utf-8 -*-
"""
Created on Thu Dec 20 22:10:47 2018

@author: profa
"""
## nltk examples
import nltk
from nltk.tokenize import word_tokenize


text="To be or not to be"

tokens = [t for t in text.split()]
print(tokens)

freq = nltk.FreqDist(tokens)
```

```
['To', 'be', 'or', 'not', 'to', 'be']
```

In [2]:
```
for key,val in freq.items():
    print (str(key) + ':' + str(val))
```

```
To:1
be:2
or:1
not:1
to:1
```

In [6]:
```
freq.plot(20, cumulative=False)
```




```
In [4]: mytext = "Hiking is dfsd fun! Hiking with dogs is more fun :)"
        print(word_tokenize(mytext))
```

```
['Hiking', 'is', 'dfsd', 'fun', '!', 'Hiking', 'with', 'dogs', 'is', 'more', 'fun', ':', ')']
```

```
In [5]: ##########################################################
        ##########################################################

        ## Lets say we are unhappy with the tokenizer we are using
        ## and wish to explicitly identify rules to define tokens
        ## Try re  and regular expressions!!
        ## https://docs.python.org/3.4/library/re.html

        #%%
        import re
        line = "Lets assume we scrapted some text data from a website or corpus \
                Lets try to find all of the valid email addresses such as \
                asdfal2@als.com, Users1@gmail.de  \
                but not Dariush@@dasd-asasdsa.com.lo nor @someDomain.com \
                what regex could we use ?!?!?!"

        print("\n\nword_tokenizer results ... ")
        print(word_tokenize(line))
```

```
word_tokenizer results ...
['Lets', 'assume', 'we', 'scrapted', 'some', 'text', 'data', 'from', 'a', 'website', 'or', 'co
```

```
In [9]: print("\n\nre results with regex defined appropriately ... ")
        match = re.findall(r'[\w\.-]+@[\w\.-]+', line)
        for i in match:
            print(i)
```

```
re results with regex defined appropriately ...
asdfal2@als.com
Users1@gmail.de
```

```
In [ ]: #%%
        ## In-Class Exercise
        ## https://docs.python.org/3.4/library/re.html
        ################################################
        ################################################
        ################################################

        ## Use re to find tokens within a string of the following form.
        ## Test on input strings to confirm correctness.
        ## State Any Assumptions you may make.
        ## 1) Dollar Amounts
        ## 2) U.S. phone numbers
        ## 3) Websites
        ##
        ##
        ##
```