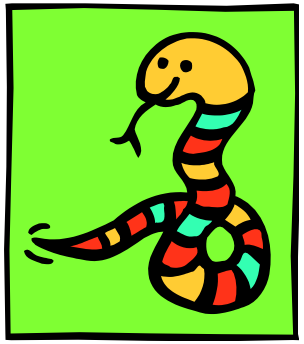
NLTK Texts and Corpora

LING 131A, Fall 2018

Marc Verhagen, Brandeis University



Today

- Assignment 1
- Assignment 2
- Quiz 1: content and examples
- Some loose ends on classes
  - extra class
  - variable access
  - class methods
- NLTK texts and corpora
- Exercise

Quiz contents

- All lecture notes
- NLTK book chapter 1 and 2
  - see LATTE for more precise info
- questions
  - multiple choice, mostly on Python
  - open-ended NLTK questions
  - a couple of open-ended Python programming questions

Loose end from last week

```
class Student:

    def __init__(self, n, a):
        self.full_name = n
        self.age = a

    def get_age(self):
        self.hair = "black"
        return self.age

    def get_hair_color(self):
        return self.hair
```

Loose end from last week

```
>>> bob = Student('Bob Smith', 23)
>>> bob.full_name  # Access an attribute.
'Bob Smith'
>>> bob.age        # Access an attribute.
23
>>> bob.hair       # Access an attribute.
??                 # This will give an error.
>>> bob.get_age()  # Access a method.
23
>>> bob.hair       # Access an attribute again.
??                 # Now it will succeed.
```

Class methods

- Regular instance methods are associated with an instance of a class

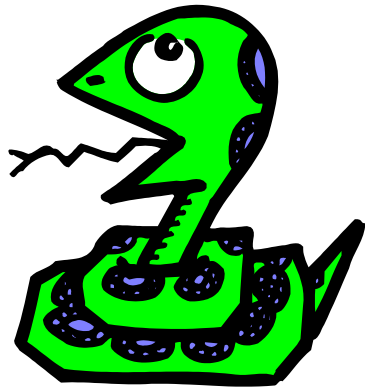

```
>>> fluffy = Dog(fluffy)
>>> fluffy.get_name()
'fluffy'
```


- Class methods are associated with the class itself

```
>>> Dog.get_count()
1
```

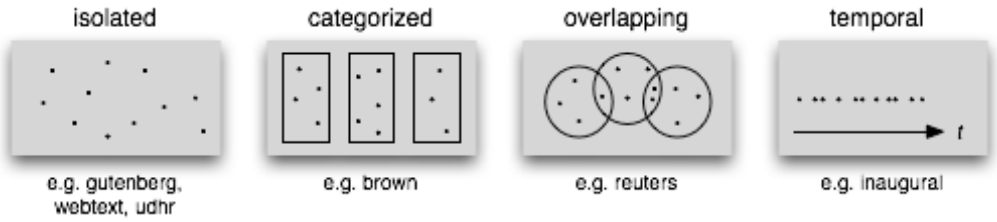
```
class Dog(object):

    count = 0

    def __init__(self, name):
        self.name = name
        self.__class__.count += 1

    @classmethod
    def get_count(cls):
        return cls.count


if __name__ == '__main__':
    d1 = Dog('fluffy')
    d2 = Dog('fido')
    print(Dog.get_count())
```

NLTK Texts and Corpora



Text Corpus

- Structured collection of texts
  - That is, a corpus is usually built for some purpose
- Used for text analysis and training ML models
- Some types:
  - raw versus annotated
  - monolingual versus multilingual
  - text only versus multi-modal
  - parallel/aligned/comparable
  - Types in NLTK



Distribution

- "You know a word by the company it keeps"

- Distribution

  - Frequency distribution
  - Neighboring words

    - Concordance/KWIC
    - Collocations
  - Similar words
    - Words that have the same neighbors

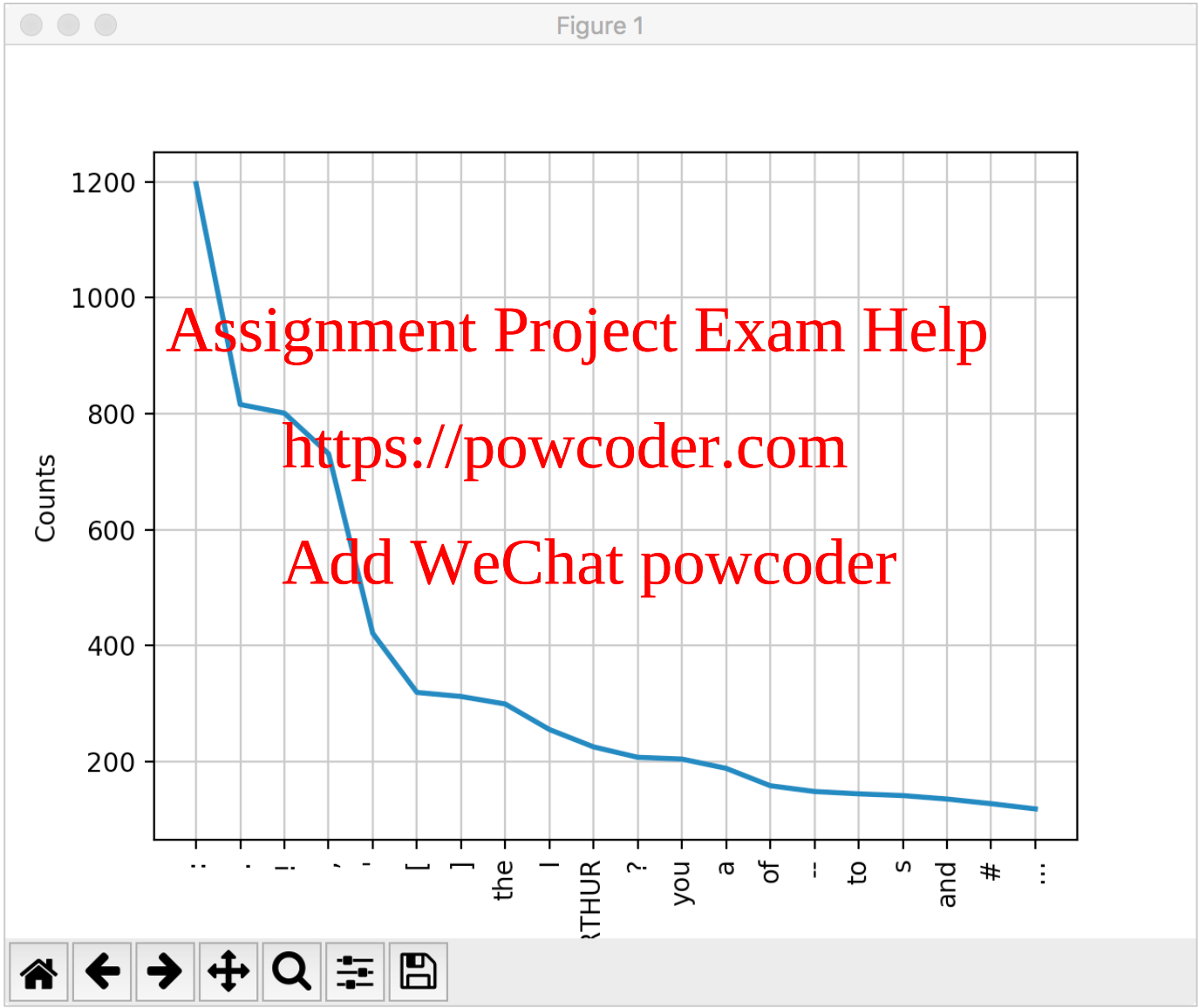
Zipf's Law

Given some text, the frequency of any word is inversely proportional to its rank in the frequency table.

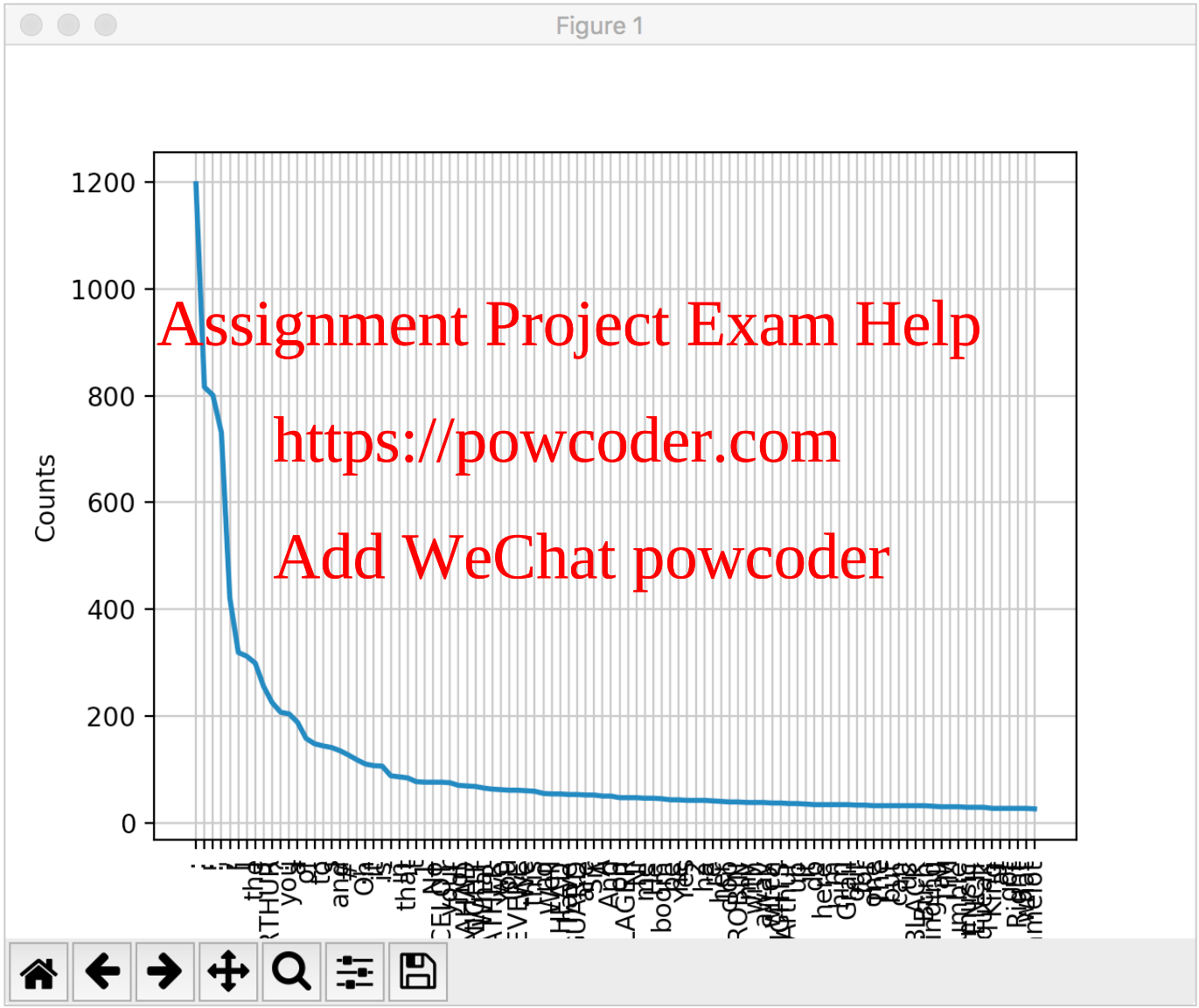
- The most frequent word will occur approximately twice as often as the second most frequent word, three times as often as the third most frequent word, etc.

- Only a small set of words (types) accounts for a large part of the text, for example, the Brown Corpus of American English text has a bout a million words (tokens) and only 135 vocabulary items are needed to account for half of them

```
FreqDist(text6).plot(20)
```



```
FreqDist(text6).plot(100)
```



Bigrams and Collocations

- Collocations are special kinds of bigrams
  - Mutual Information
  - Kenneth Ward Church and Patrick Hanks. 1990. *Word association norms, mutual information, and lexicography*. Computational Linguistics, Volume 16 Issue 1, March 1990. Pages 22-29.
  - Defined as

$$MI(x,y) = \log_2 \frac{P(x,y)}{P(x)P(y)}$$

Examples

11.05	8	8	8	Round Table
10.73	10	10	10	Pie Iesu
10.73	10	10	10	Iesu domine
7.54	7	13	1	sacred quest
6.00	7	38	1	join my
2.85	107	22	1	it will
-1.85	204	299	1	you the

https://www.youtube.com/watch?v=YgYEuJ5u1K0

sacred quest

length of text6 is 16,967

P(x,y) = 1/16,967 = 0.0000589

P(x) = 7/16,967 = 0.0004126

P(y) = 13/16,967 = 0.0007662

MI(x,y) = log2(0.0000589 / (0.0004126 * 0.0007662))

= log2(186.3133) = 7.54

$$MI(x,y) = \log_2 \frac{P(x,y)}{P(x)P(y)}$$

NLTK

- Text
- FreqDist
- CorpusReader
  - PlainTextCorpusReader
  - CategorizedTaggedCorpusReader
- ConcatenatedCorpusView
- StreamBackedCorpusView