Dictionaries

Case study: Word frequency analysis

- We want to answer some questions about the English language
- How common is the word “happy”?
- What’s the most commonly used word?
- What percent of words have greater than four words?

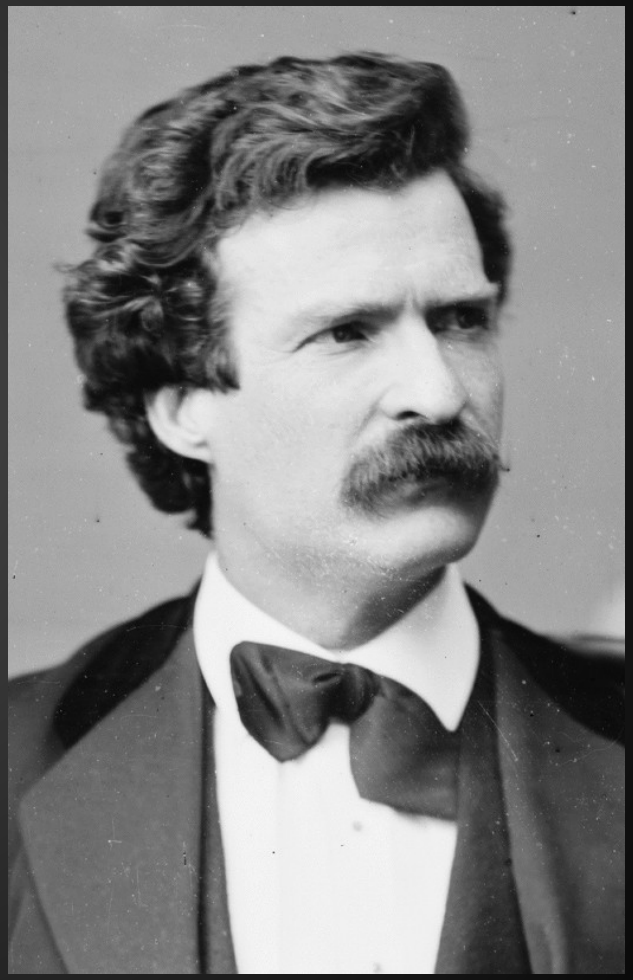
How can we do this using Python?

Steps

1. Find some text data (we'll use The Adventures of Tom Sawyer)
2. Load the text into a Python program
3. Split the text into a list of words
4. Scan the list of words and keep track of relevant data (for example, word count)
5. Analyze the data to answer questions
6. Store data to use later

Step 1: Find text data

- After an intense Googling session, I found http://www.gutenberg.org/ for free books!
- Many classics were translated from French (e.g. Alexandre Dumas, Jules Verne, etc)
- Mark Twain is a true American hero



Step 2: Load the text into Python

- This brings us to…FILE I/O
- I/O stands for input/output
- We'll store The Adventures of Tom Sawyer as one giant string in our Python program

Step 2: Load the text into Python

```
load_file = open('tom_sawyer.txt')
giant_string = load_file.read()
load_file.close()
print giant_string
```

Step 3: Split the text

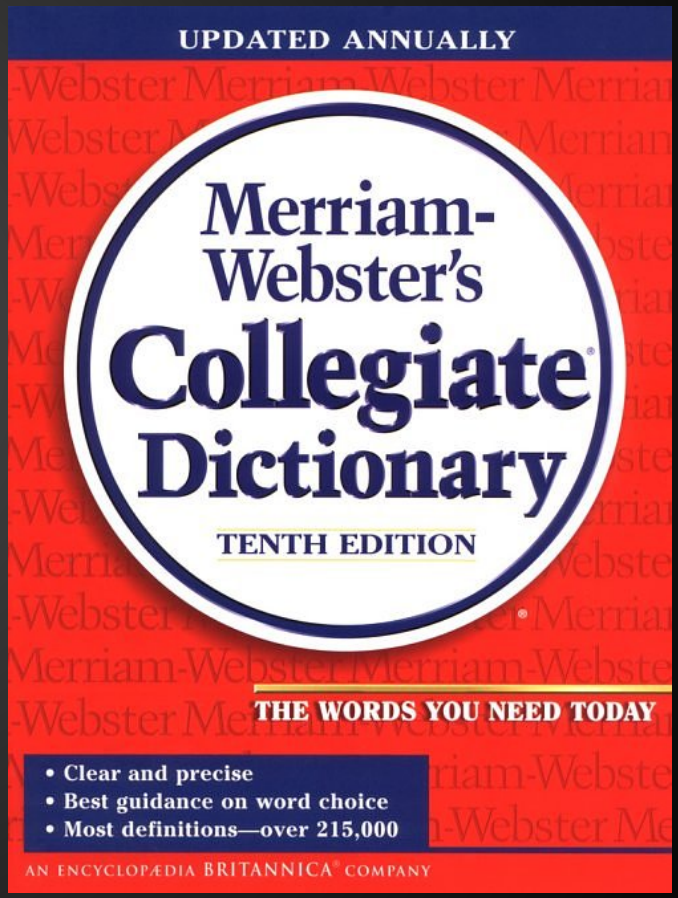
- We want to analyze on a word-by-word basis
- Therefore, we have to convert our giant string into a list of all the words
- You've already done this with string.split

Step 3: Split the text

```
load_file = open(‘tom_sawyer.txt’)
giant_string = load_file.read()
load_file.close()
word_list = giant_string.split()
print word_list
```

Step 4: Scan list of words, track data

- We need a DICTIONARY!
- Lists were our first “data structure”, dictionaries are our second
- In our dictionary we’ll store “key-value pairs” where “key” is the word and “value” is the word count
- In a real dictionary, “key” is the word and “value” is the definition



Interlude: Dictionary syntax

```
word_counts = {‘the’: 9, ‘bat’: 4}    # initialization
word_counts[‘cave’] = 0      # add a new kv-pair
word_counts[‘cave’] += 1    # update value
print word_counts[‘cave’]   # access value
```

Step 4: Scan list of words, track data

```
load_file = open('tom_sawyer.txt')
giant_string = load_file.read()
load_file.close()
word_list = giant_string.split()
word_counts = {}
for word in word_list:
    if word not in word_counts:
        word_counts[word] = 0
    word_counts[word] += 1
print word_counts
```

Step 5: Analyze data

```
print “How many times does ‘happy’ appear?”
print word_counts[‘happy’]
print “How many total words are there?”
print len(word_list)
print “What percent of words are ‘happy’?”
print 100.0 * word_counts[‘happy’] / len(word_list)
```

Step 5: Analyze data

```
print “What’s the most commonly used word?”
max_count = 0
max_word = ‘’
for word in word_counts:
    count = word_counts[word]
    if count > max_count:
        max_count = count
        max_word = word
print max_word, max_count
```

Step 5: Analyze data

```
print “How many words are more than four letters?”
long_word_count = 0
for word in word_counts:
    count = word_counts[word]
    if len(word) > 4:
        long_word_count += count
print “How many words have more than 4 letters?”
print long_word_count
print “What percent are more than 4 letters?”
print 100.0 * long_word_count / len(word_list)
```

Step 6: Store data to use later

- If we want to run another analysis, we should just use the word counts we already computed instead of the original Mark Twain novel
- So we will store these counts to a file
- This is more File I/O!

Interlude: JSON

- We have a dictionary, and we want to save it to a file
- So first we have to “serialize” the data, or convert it to a string
- JSON: JavaScript Object Notation

```
import json
some_dict = {‘hurshal’: 23, ‘patel’: 47}
dict_string = json.dumps(some_dict)
print dict_string
```

Step 6: Store data to use later

```
import json
store_file = open(‘word_counts.json’, ‘w’)
text_to_store = json.dumps(word_counts)
store_file.write(text_to_store)
store_file.close()
```

Step 7: (Optional) Load saved data!

```
import json
word_counts_file = open('word_counts.json')
word_counts = json.loads(word_counts_file.read())
word_counts_file.close()
print word_counts
```

Questions? Comments? Concerns?