LING 334
Text Processing for Linguists

Week 6

Assignment 4 Review,
Decomposition Exercises
Notes on Assignment 4

- Using *flags* (like `remove_blank`):
  - “flags” are arguments that give options rather than data
  - Try to have core functionality only be written once; helpful if you ever need to change anything

- `letter_counts` - no need to tokenize, loop over words etc:
  - Can simply do `for character in s`
  - Remember strings are sequences
Notes on Assignment 4

● You can use `random.random()` in a conditional directly rather than saving it in a variable that you only use once

```python
if random.random() > 0.5:
```

● Avoid **hardcoding**: e.g., in the dice sums problem:

```python
sum_counts = {0: 0, 1: 0, 2: 0, 3: 0, 4: 0...}
```
Notes on Assignment 4

- `string.split()` splits in a greedy way, e.g. maximum amount of whitespace

- What’s the difference?

  ```
  s.split()   vs.   s.split(" ")
  ```
Notes on Assignment 4

- Variable naming:
  try to have names reflect the contents/purpose

- Which is better?
  
  for word in line.split()
  
  or
  
  for words in line.split()
Notes on Assignment 4

- Related style point: make objects what we will use them for for
  - e.g., `proportion_of_oneoff_types`
    Accumulate counts on an integer vs.
    Accumulate a list of oneoff types and get its length
Notes on Assignment 4

• Remember you can chain operations:

  ○ plain = s.strip()
      lower = plain.lower()
      list = lower.split()    # also list is not a good var name
      for word in list:
      vs.

  ○ for word in s.strip().lower().split():
Notes on Assignment 4

- Efficiency! Sometimes hard to spot. Where’s the problem?

```python
words = []
for line in open(f):
    tokens = tokenize(line)
    for token in tokens:
        if token in words:
            continue
        else:
            words.append(token)
return len(words)
```
Notes on Assignment 4

• if token in words:
  ○ If \texttt{words} is a list, this has to do a sequential check through the entire list every time this is called.
    ■ Number of operations $= \text{size of list}$
  ○ If \texttt{words} is a set, this is an instantaneous operation, due to a nice thing called hashing
    ■ Number of operations $= 1$ (roughly)
Decomposition

Breaking down an abstract problem into smaller parts we can handle

variables
loops
conditionals
functions
methods
modules

Who rhymes more often, Beyoncé or Taylor Swift?
Question-Answer pair
worked example
If time:

Anagram Finder

worked example
Basic steps:

- **wget** assignment link into a Quest assignment5 directory
- **Do unzip assignment.zip**
- **Go to** [https://jupyter.questanalytics.northwestern.edu](https://jupyter.questanalytics.northwestern.edu) (must be on NU VPN)
- Navigate to your assignment5 dir and open ‘Assignment 5.ipynb’