Prompt:

The goal of this exercise to introduce to various tools that are commonly used to deal text such as text retrieval, sorting, words counts, etc. You can think of this a warm up exercise. In order to complete this assignment you must have Mac or Linux system (perhaps Ubuntu). If you have Windows, you may need to install Oracle virtual machine to install Ubuntu in it.

Tools that you will be using are the following:

- grep: search for a pattern (regular expression)
- sort
- uniq –c (count duplicates)
- tr (translate characters)
- wc (word – or line – count)
- sed (edit string -- replacement)
- cat (send file(s) in stream)
- echo (send text in stream)
- cut (columns in tab-separated files)
- paste (paste columns)
- head
- tail
- rev (reverse lines)
- comm
- join

Hints:

- This is a group assignment; one team member should submit the assignment on behalf of the team.
- You should have access to the “man” command in unix (e.g., man tr)
- You should know how to use the chain shell commands and deal with input/output
  - Input/output redirection:
    - > “output to a file”
    - < ”input from a file”
    - | “pipe”
  - CTRL-C
  - The less command (quit by typing "q")

Exercise 1: Count words in a text

- Input: text file (ny_article.txt)
- Output: list of words in the file with freq counts
- Algorithm
  - Tokenize (tr)
  - Sort (sort)

- Count duplicates (uniq –c)

Hint: read the man pages and figure out how to pipe these together

Extended Counting Exercises

- Merge upper and lower case by downcasing everything
  - Hint: Put in a second tr command
- How common are different sequences of vowels (e.g., the sequences "ieu" or just "e" in "lieutenant")?
  - Hint: Put in a second tr command

Sorting and reversing lines of text

- sort
- sort –f Ignore case
- sort –n Numeric order
- sort –r Reverse sort
- sort –nr Reverse numeric sort
- echo "Hello" | rev

Exercise 2: Counting and sorting

- Find the 50 most common words in the NYT
  - Hint: Use sort a second time, then head
- Find the words in the NYT that end in "zz"
  - Hint: Look at the end of a list of reversed words
  - tr 'A-Z' 'a-z' < filename | tr –sc 'a-z' '\n' | rev | sort | rev | uniq -c

Piping:

- Piping commands together can be simple yet powerful in Unix
- It gives flexibility.
- Traditional Unix philosophy: small tools that can be composed

Bigrams= word pairs and their counts

Algorithm:

1. Tokenize by word
2. Create two almost-duplicate files of words, off by one line, using tail
3. **Paste** them together so as to get wordi and wordi +1 on the same line
4. Count

```
tr -sc 'A-Za-z' '\n' < nyt_200811.txt >
nyt.words
tail -n +2 nyt.words > nyt.nextwords
paste nyt.words nyt.nextwords > nyt.bigrams
head –n 5 nyt.bigrams
    KBR     said
    said    Friday
    Friday  the
    the     global
    global  economic
```

Exercise 3: bigrams

- Find the 10 most common bigrams

  (For you to look at:) What part-of-speech pattern are most of them?

- Find the 10 most common trigrams

Topic: grep

- Grep finds patterns specified as regular expressions
  globally search for regular expression and print
- Finding words ending in –ing:
  grep 'ing$' nyt.words |sort | uniq –c

grep is a filter – you keep only some lines of the input

- grep gh keep lines containing ‘‘gh’’
- grep '^con' keep lines beginning with ‘‘con’’
- grep 'ing$' keep lines ending with ‘‘ing’’
- grep –v ghkeep lines NOT containing “gh”

grep versus egrep (grep –E)

- egrep or grep -E [extended syntax]
- In egrep, +, ?, |, (, and) are automatically metacharacters
- In grep, you have to backslash them
- To find words ALL IN UPPERCASE:
- egrep '^[A-Z]+$' nyt.words |sort|uniq -c
- == grep '^[A-Z]\+$' nyt.words |sort|uniq -c

(confusingly on some systems grep acts like egrep)

Counting lines, words, characters

- wc nyt_200811.txt
  70334 509851 3052306 nyt_200811.txt
- wc -l nyt.words
  70334 nyt_200811.txt

Exercise:

Why is the number of words different?

Exercises on grep & wc

- How many all uppercase words are there in this NYT file?
- How many 4-letter words?
- How many different words are there with no vowels
  What subtypes do they belong to?
- How many "1 syllable" words are there
  That is, ones with exactly one sequence of vowels

Type/instance distinction: different words (types) vs. instances

(sometimes called "type/token" distinction but we now save "token" for BPE tokens)