Liqi Zhu's homework 1

1

```
tr -sc 'A-Za-z' '\n' < alice.txt  | tr '[:upper:]' '[:lower:]' > words.txt
sed 's/[aeiou].*$//g' words.txt | sort | uniq -c | sort -r | head
tr -sc 'A-Za-z' '\n' < alice.txt  | tr '[:upper:]' '[:lower:]' | rev > words2.txt
sed 's/[aeiou].*$//g' words2.txt | rev | sort | uniq -c | sort -r | head
```

Results:

```
$ sed 's/[aeiou].*$//g' words.txt | sort | uniq -c | sort -r | head
   3873
   1308 th
    675 w
    634 s
    625 t
    597 h
    414 d
    403 c
    388 f
    383 m

$ sed 's/[aeiou].*$//g' words2.txt | rev | sort | uniq -c | sort -r | head
   4083
    995 t
    910 n
    771 r
    760 d
    759 s
    541 nd
    450 ng
    392 f
    229 ll
```

2

```
tr -sc 'A-Za-z' '\n' < alice.txt > alice.words
tail -n +2 alice.words > alice.nextwords
tail -n +3 alice.words > alice.nnwords
paste alice.words alice.nextwords | sort | uniq -c | sort -r > alice.bigrams
```

```
paste alice.words alice.nextwords alice.nnwords | sort | uniq -c | sort -r >alice.trigrams

head -n 10 alice.bigrams
head -n 10 alice.trigrams
```

Results:

```
$ head -n 10 alice.bigrams
     83 Project Gutenberg
     71 of      the
     60 said    the
     56 Gutenberg       tm
     49 in      the
     46 in      a
     45 said    Alice
     40 to      the
     38 and     the
     29 with    the
```

Most of them are Prepositions.

```
$ head -n 10 alice.trigrams
     56 Project Gutenberg       tm
     27 the     Project Gutenberg
     18 Gutenberg       tm      electronic
     14 the     White   Rabbit
     13 said    the     King
     13 Project Gutenberg       Literary
     13 Literary        Archive Foundation
     13 Gutenberg       Literary        Archive
     12 tm      electronic      works
     12 the     terms   of
```

Caused by the copyright info.

3

```
cat dekker/*.txt > dekker.txt
cat johnson/*.txt > johnson.txt
cat marlowe/*.txt > marlowe.txt
cat middleton/*.txt > middleton.txt
cat webster/*.txt > webster.txt
tr -sc 'A-Za-z' '\n' < shakes.txt | tr '[:upper:]' '[:lower:]' | uniq > shakes.words
wc -l shakes.words > words.results.txt
tr -sc 'A-Za-z' '\n' < dekker.txt | tr '[:upper:]' '[:lower:]' | uniq > dekker.words
wc -l dekker.words >> words.results.txt
tr -sc 'A-Za-z' '\n' < johnson.txt | tr '[:upper:]' '[:lower:]' | uniq > johnson.words
```

```
wc -l johnson.words >> words.results.txt
tr -sc 'A-Za-z' '\n' < marlowe.txt | tr '[:upper:]' '[:lower:]' | uniq > marlowe.words
wc -l marlowe.words >> words.results.txt
tr -sc 'A-Za-z' '\n' < middleton.txt | tr '[:upper:]' '[:lower:]' | uniq > middleton.words
wc -l middleton.words >> words.results.txt
tr -sc 'A-Za-z' '\n' < webster.txt | tr '[:upper:]' '[:lower:]' | uniq > webster.words
wc -l webster.words >> words.results.txt
cat dekker.txt johnson.txt marlowe.txt middleton.txt webster.txt > allother.txt
tr -sc 'A-Za-z' '\n' < allother.txt | tr '[:upper:]' '[:lower:]' | uniq > allothers.words
wc -l allothers.words >> words.results.txt
```

Results:

```
$ cat words.results.txt
34701 shakes.words
10989 dekker.words
25123 johnson.words
14020 marlowe.words
8385 middleton.words
9820 webster.words
38669 allothers.words
```

I picked Writers and playwrights working in the same era as Shakespeare include Christopher Marlowe, Thomas Middleton, John Webster, Ben Jonson and Thomas Dekker. The dataset is all these six authors' works I can find on Gutenberg.org. With limited number of examples, I would rather choose the authors in the same time period as Shakespeare since english changes as time period changes.

From the count of uniq words for each author, Shakespeare's count of uniq words is obviously larger than any of the other 5. Then I combine all the works of other authors, counts of uniq words in allothers.words is a little larger than Shakespeare. Compared to the authors in the same time period, Shakespeares seems to have more diversity in words while words' diversity of others' combination is still larger than shakespear.

Considering one of his best competitor C.Marlowe died young with much less works finished, also the feature of using oral argot is part of Shakespeare's works, which makes the words count larger than it should be. I can only conclude that based on unbalanced volume of works inplies Shakespeare has the most diversity, while it may not be true.

Limitations such as the size of the dataset. The dataset of the authors is not perfect where Shakespeare is the only author with the full works on Gutenberg.org.
This approach can't combine similar words neither, different tenses or personification are counted seperately, so the author with a larger volume of works tend to have more uniq words. Error can be decreased if there's enough data.

4

```
cat inaugural/*.txt > all.txt
tr -sc '[A-Za-z]' '\n' < all.txt | sort | uniq -c | sort -r > all.words
cat inaugural/178*.txt > 178_.txt
tr -sc '[A-Za-z]' '\n' < 178_.txt | sort | uniq -c | sort -r
cat inaugural/179*.txt > 179_.txt
tr -sc '[A-Za-z]' '\n' < 179_.txt | sort | uniq -c | sort -r
cat inaugural/180*.txt > 180_.txt
tr -sc '[A-Za-z]' '\n' < 180_.txt| sort | uniq -c | sort -r
cat inaugural/181*.txt > 181_.txt
tr -sc '[A-Za-z]' '\n' < 181_.txt| sort | uniq -c | sort -r
cat inaugural/182*.txt > 182_.txt
tr -sc '[A-Za-z]' '\n' < 182_.txt | sort | uniq -c | sort -r
cat inaugural/183*.txt > 183_.txt
tr -sc '[A-Za-z]' '\n' < 183_.txt | sort | uniq -c | sort -r
cat inaugural/184*.txt > 184_.txt
tr -sc '[A-Za-z]' '\n' < 184_.txt| sort | uniq -c | sort -r
cat inaugural/185*.txt > 185_.txt
tr -sc '[A-Za-z]' '\n' < 185_.txt | sort | uniq -c | sort -r
cat inaugural/186*.txt > 186_.txt
tr -sc '[A-Za-z]' '\n' < 186_.txt | sort | uniq -c | sort -r
cat inaugural/187*.txt > 187_.txt
tr -sc '[A-Za-z]' '\n' < 187_.txt | sort | uniq -c | sort -r
cat inaugural/188*.txt > 188_.txt
tr -sc '[A-Za-z]' '\n' < 188_.txt | sort | uniq -c | sort -r
cat inaugural/189*.txt > 189_.txt
tr -sc '[A-Za-z]' '\n' < 189_.txt | sort | uniq -c | sort -r
cat inaugural/190*.txt > 190_.txt
tr -sc '[A-Za-z]' '\n' < 190_.txt | sort | uniq -c | sort -r
cat inaugural/191*.txt > 191_.txt
tr -sc '[A-Za-z]' '\n' < 191_.txt | sort | uniq -c | sort -r
cat inaugural/192*.txt > 192_.txt
tr -sc '[A-Za-z]' '\n' < 192_.txt | sort | uniq -c | sort -r
cat inaugural/193*.txt > 193_.txt
tr -sc '[A-Za-z]' '\n' < 193_.txt | sort | uniq -c | sort -r
cat inaugural/194*.txt > 194_.txt
tr -sc '[A-Za-z]' '\n' < 194_.txt | sort | uniq -c | sort -r
cat inaugural/195*.txt > 195_.txt
tr -sc '[A-Za-z]' '\n' < 195_.txt | sort | uniq -c | sort -r
cat inaugural/196*.txt > 196_.txt
tr -sc '[A-Za-z]' '\n' < 196_.txt | sort | uniq -c | sort -r
cat inaugural/197*.txt > 197_.txt
tr -sc '[A-Za-z]' '\n' < 197_.txt | sort | uniq -c | sort -r
```

```
cat inaugural/198*.txt > 198_.txt
tr -sc '[A-Za-z]' '\n' < 198_.txt | sort | uniq -c | sort -r
cat inaugural/199*.txt > 199_.txt
tr -sc '[A-Za-z]' '\n' < 199_.txt | sort | uniq -c | sort -r
cat inaugural/200*.txt > 200_.txt
tr -sc '[A-Za-z]' '\n' < 200_.txt | sort | uniq -c | sort -r
egrep -wc 'war' *_.txt > topics.result.txt
egrep -wc 'jobs' *_.txt >> topics.result.txt
egrep -wc 'government' *_.txt >> topics.result.txt
egrep -wc 'people' *_.txt >> topics.result.txt
egrep -wc 'world' *_.txt >> topics.result.txt
egrep -wc 'state' *_.txt >> topics.result.txt
egrep -wc 'nation' *_.txt >> topics.result.txt
egrep -wc 'country' *_.txt >> topics.result.txt
egrep -wc 'citizen' *_.txt >> topics.result.txt
egrep -wc 'power' *_.txt >> topics.result.txt
egrep -wc 'public' *_.txt >> topics.result.txt
egrep -wc 'freedom' *_.txt >> topics.result.txt
egrep -wc 'constitution' *_.txt >> topics.result.txt
egrep -wc 'spirit' *_.txt >> topics.result.txt
egrep -wc 'law' *_.txt >> topics.result.txt
egrep -wc 'justice' *_.txt >> topics.result.txt
egrep -wc 'liberty' *_.txt >> topics.result.txt
egrep -wc 'political' *_.txt >> topics.result.txt
egrep -wc 'foreign' *_.txt >> topics.result.txt
egrep -wc 'policy' *_.txt >> topics.result.txt
egrep -wc 'history' *_.txt >> topics.result.txt
egrep -wc 'republic' *_.txt >> topics.result.txt
egrep -wc 'commerce' *_.txt >> topics.result.txt
egrep -wc 'security' *_.txt >> topics.result.txt
egrep -wc 'business' *_.txt >> topics.result.txt
egrep -wc 'civil' *_.txt >> topics.result.txt
egrep -wc 'welfare' *_.txt >> topics.result.txt
egrep -wc 'territory' *_.txt >> topics.result.txt
egrep -wc 'population' *_.txt >> topics.result.txt
cat topics.result.txt | sed 's/\t/,/g;s/[[:space:]]//g' >result2.csv
```

Results:
I picked topics of [war, jobs, government, people, world, state, nation, country, citizen, power, public, freedom, constitution, spirit, law, justice, liberty, political, foreign, policy, history, republic, commerce, security, business, civil, welfare, territory] from the frist sort-r function on all words.
Then I devided time period into decades, count the frequency of each topic in decades, then fill in spreedsheet as follow.

	war	jobs	government	people	world	state	nation	country	citizen	power	public	freedom	constitution	spirit	law	justice	liberty	political	foreign	policy	history	republic	commerce	security	business	civil	welfare	territory
1780s	0	0	4	3	1	0	2	3	0	1	4	0	0	0	0	0	1	0	0	1	0	0	0	1	0	1	0	0
1790s	1	0	7	11	3	0	8	7	1	4	4	0	1	2	0	4	3	1	6	0	1	0	2	0	0	0	1	0
1800s	3	0	3	3	5	7	6	10	0	7	11	5	1	4	5	5	6	3	3	1	4	0	4	0	0	3	0	1
1810s	14	0	2	10	1	6	8	10	4	3	4	0	0	4	0	4	0	4	8	4	1	0	3	3	0	0	1	2
1820s	17	0	9	16	2	5	13	13	1	14	13	5	1	7	3	6	4	3	10	8	0	0	10	1	0	4	4	4
1830s	1	0	7	20	4	1	1	13	1	6	7	0	0	7	1	3	3	5	6	5	2	1	1	1	0	2	5	1
1840s	5	0	15	30	5	11	7	32	4	22	15	8	1	11	3	3	12	7	10	13	2	4	4	3	1	6	2	4
1850s	3	0	9	14	4	2	11	11	1	9	9	3	2	6	2	3	5	4	6	3	7	2	3	4	0	3	0	3
1860s	5	0	2	13	3	0	6	10	3	4	7	0	0	0	10	1	1	4	2	3	1	0	1	4	1	3	0	0
1870s	3	0	8	13	5	0	8	22	2	5	9	0	0	2	3	1	0	8	3	2	4	0	2	0	0	5	3	2
1880s	8	0	21	47	5	1	17	11	8	16	16	6	1	3	17	9	7	7	3	7	5	1	5	4	6	7	3	4
1890s	3	0	5	31	4	0	3	14	3	10	13	1	0	4	6	2	1	3	4	4	2	0	1	0	8	2	1	0
1900s	6	0	11	16	5	2	6	10	0	6	7	5	1	6	13	5	5	7	4	14	1	1	5	3	15	2	2	0
1910s	3	0	4	7	8	0	6	1	0	3	0	2	0	5	1	6	2	2	0	1	1	0	2	0	2	0	0	0
1920s	12	0	21	27	30	3	6	23	3	6	10	12	1	4	17	18	7	12	3	9	2	4	0	4	7	2	7	0
1930s	2	0	13	16	5	0	7	4	1	6	6	0	0	6	1	1	1	4	2	2	0	0	0	1	1	0	2	1
1940s	4	0	5	18	26	2	5	5	1	1	0	17	0	8	1	5	4	0	1	0	6	0	1	10	1	0	2	1
1950s	4	0	1	20	28	0	6	8	3	8	0	16	0	2	4	3	1	4	0	0	6	0	2	6	0	0	0	0
1960s	8	0	6	19	23	1	11	4	3	6	1	8	0	6	2	4	5	0	1	0	7	0	1	0	0	0	0	1
1970s	3	0	7	9	18	0	9	4	0	1	0	7	0	7	2	1	2	0	1	0	7	0	0	0	0	0	1	0
1980s	3	1	17	28	26	1	14	6	1	4	2	19	0	2	1	1	5	5	1	0	13	0	0	6	1	1	1	0
1990s	1	1	7	16	20	0	15	1	1	7	1	5	0	5	2	1	3	3	1	0	4	0	2	1	1	1	2	0
2000s	3	3	8	13	15	1	30	18	2	10	6	27	0	6	3	8	11	1	0	1	11	0	1	3	1	3	0	0

The first interesting fact is that actually big topics don't appear much. Instead of real topics, general words such as 'government, public, nation, state, power, freedom, spirit' have very high frequency. There's also huge amounts of words with similar or related meaning such as 'nation/country', 'people/citizen/public', 'freedom/liberty/justice', 'business, commerce'.

We can also find that for each topic, there's differences amoung decades. Wars' frequency is high in 1810s 1820s and decades of WWI, WWII and cold war. Jobs' topic didn't appear until 1980s. Law's frequency has been low since 1920s. These are all related to history events.

We can also find there's some words never missed a speech such as 'people, government, nation, country, power', somewords almost appear in every speech such as 'war, freedom, spirit, justice, liberty, foreign'. There's also powerful but rare topics such as 'constitution'.

What supreised me is the low fequency of 'welfare' and 'territory' topic, big topics but really low frequency.