hw7_490IDS_17

17

10/22/2016

(a) Words with @ symbols in them, e.g., h@te or v|c0din

```
print("^.*[[:punct:]].*$")

## [1] "^.*[[:punct:]].*$"

a = "ah@te12|b"
regexpr("^.*[[:punct:]].*$" ,a)

## [1] 1
## attr(,"match.length")
## [1] 9
## attr(,"useBytes")
## [1] TRUE

a = "v|c0dim"
regexpr("^.*[[:punct:]].*$" ,a)

## [1] 1
## attr(,"match.length")
## [1] 7
## attr(,"useBytes")
## [1] TRUE
```

(b) An IP address (Four sets of 1 to 3 digits separated by periods, e.g., 100.12.162.0)

```
print("^\\d{1,3}\\.\\d{1,3}\\.\\d{1,3}\\.\\d{1,3}$" )

## [1] "^\\d{1,3}\\.\\d{1,3}\\.\\d{1,3}\\.\\d{1,3}$"

a = "100.12.162.0"
regexpr("^\\d{1,3}\\.\\d{1,3}\\.\\d{1,3}\\.\\d{1,3}$" ,a)

## [1] 1
## attr(,"match.length")
## [1] 12
## attr(,"useBytes")
## [1] TRUE

a = "100.12.162.0a"
regexpr("^\\d{1,3}\\.\\d{1,3}\\.\\d{1,3}\\.\\d{1,3}$" ,a)

## [1] -1
## attr(,"match.length")
## [1] -1
## attr(,"useBytes")
## [1] TRUE
```

(c) An email address that ends with .com, .edu, .net, .org, or .gov

```
print("^[^@]*@[^@]*[\\.com|\\.edu|\\.net|\\.org|\\.gov]$")

## [1] "^[^@]*@[^@]*[\\.com|\\.edu|\\.net|\\.org|\\.gov]$"

a = "ah@te12b.org"
regexpr("^[^@]*@[^@]*[\\.com|\\.edu|\\.net|\\.org|\\.gov]$" ,a)

## [1] 1
## attr(,"match.length")
## [1] 12
## attr(,"useBytes")
## [1] TRUE
```

Q. 2) (19 points) Carry out the following exercises on the State of the Union speeches database (available in moodle). (a) Use readLines() to read in the speeches (available as a text file in moodle) where the return value is: character vector with one element/character string per line in the file

```
myText <- readLines("/Users/zyh/Documents/2016Fall/Data
science/hw7/stateoftheunion1790-2012.txt")
```

(b) Use regular expressions to find ***

```
stars= unlist(regmatches(myText, gregexpr("\\*\\*\\*", myText)))
```

(c) Use *** to identify the date of the speech.

```
locations = grep("^\\*\\*\\*$", myText)
dates = myText[locations[]+4]
```

(d) Use regular expressions to extract the year.

```
years = regexpr("\\<\\d\\d\\d\\d\\>",dates)
years_list = unlist(regmatches(dates, gregexpr("\\<\\d\\d\\d\\d\\>", dates)))
```

(e) Use regular expressions to extract the month.

```
months = regexpr("^[[:upper:]][[:lower:]]*\\>",dates)
months_list = unlist(regmatches(dates,
gregexpr("^[[:upper:]][[:lower:]]*\\>", dates)))
```

(f) Use *** to extract the name of the president State of the union speeches.

```
presidents = myText[locations[]+3]
```

(g) Use regular expressions and R to return the number of speeches in the dataset, and the number of presidents that gave speeches.

```
num_speech = length(presidents)
print("number of speeches")

## [1] "number of speeches"
```

```
num_speech

## [1] 222

president_list = as.data.frame(table(presidents))
president_list = unlist(c(president_list["presidents"]))
num_president = length(president_list)
print("number of presidents")

## [1] "number of presidents"

num_president

## [1] 41
```

(h) Chop the speeches up into a list there is one element for each speech. Each element is a character vector. Check: does your number of list elements match your answer above?

```
speeches = c()
for (i in 1:(length(locations)-1)){
  a = c(unlist(myText[locations[i]:locations[i+1]-1], recursive = TRUE))
  a = paste(a, collapse = " ")
  speeches[i] =a
}
a = c(unlist(myText[locations[length(locations)]:(length(myText)-1)],
recursive = TRUE))
a = paste(a, collapse = " ")
speeches[length(locations)] =a
length(speeches)

## [1] 222

print("Yes, there's 222 elements.")

## [1] "Yes, there's 222 elements."
```

(i) Eliminate apostrophes, numbers, and the phrase: (Applause.)

```
newspeeches = speeches
for (i in 1:length(newspeeches)){
  newspeeches[i] = gsub("\\<\\S+'\\S+\\>", "", newspeeches[i])
}

for (i in 1:length(newspeeches)){
  newspeeches[i] = gsub("[[:digit:]]", "", newspeeches[i])
}


for (i in 1:length(newspeeches)){
  newspeeches[i] = gsub("\\<.*Applause.*\\>", "", newspeeches[i])
}
```

(j) Make all the characters lower case.

```
newspeeches <- lapply(newspeeches, FUN = tolower)
```

(k) Split the sentences up where there are blanks and punctuation to create "words".

```
words = c()
for (i in 1:length(newspeeches)){
  words = c(words,unlist(strsplit(as.character(newspeeches[i]),
"[[:blank:]]|[[:punct:]]",perl = TRUE,useBytes = TRUE)))
}
print(length(words))

## [1] 1868868
```

(l) Drop any empty words that resulted from this split.

```
words = words[words != ""]
print(length(words))

## [1] 1664074
```

(m) Create a word vector for each speech.

```
words_vec = list()
for (i in 1:length(newspeeches)){
  a =unlist(strsplit(as.character(newspeeches[i]),
"[[:blank:]]|[[:punct:]]",perl = TRUE,useBytes = TRUE))
  a = a[a!=""]
  a = as.vector(a)
  words_vec[[i]] = a
}
```

(n) Normalize the word vectors to get term frequencies.

```
frequencies = words_vec
for (i in 1:length(frequencies)){
  a = as.vector(frequencies[[i]])
  l = length(a)
  b = table(a)
  for (j in 1:length(frequencies[[i]])){
    frequencies[[i]][j] = as.numeric(b[a[j]])/l
  }
}
```

(o) (5 points) Carry out some exploratory analysis of the data and term frequencies. For example, find the number of sentences, extract the long words, and the political party. Plot and interpret the term frequencies. What are your observations?

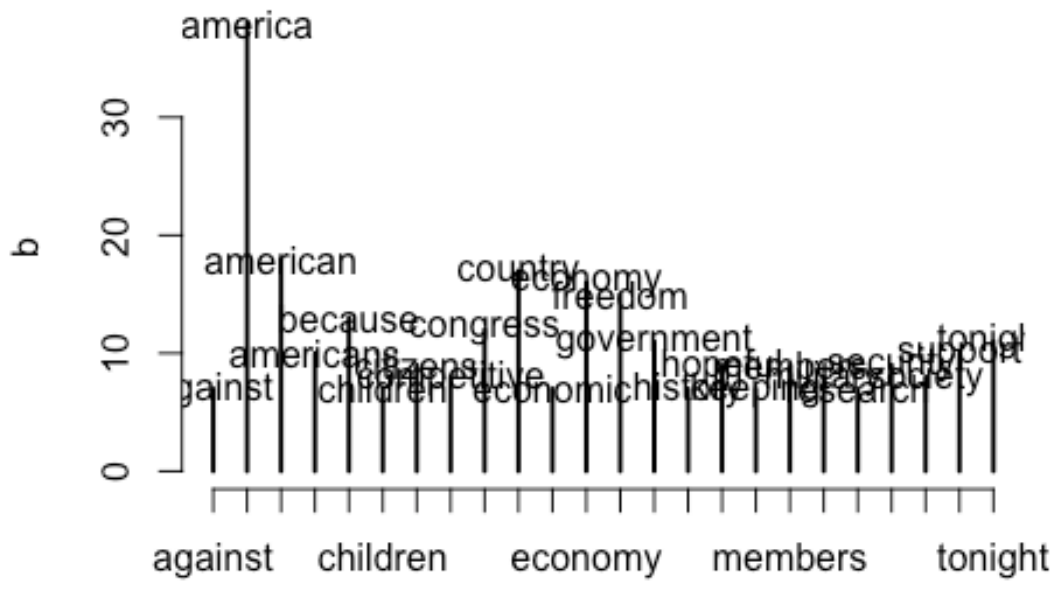
```
a = table(words_vec[[216]])
mean(a)

## [1] 3.532

sd(a)

## [1] 13.67398
```

```
c = names(a)
d = as.numeric(sapply(c,nchar))
b = a[which(d>mean(d))]
b = b[b>mean(a)+0.2*sd(a)]
plot(b)
text(b, lab=row.names(b))
```



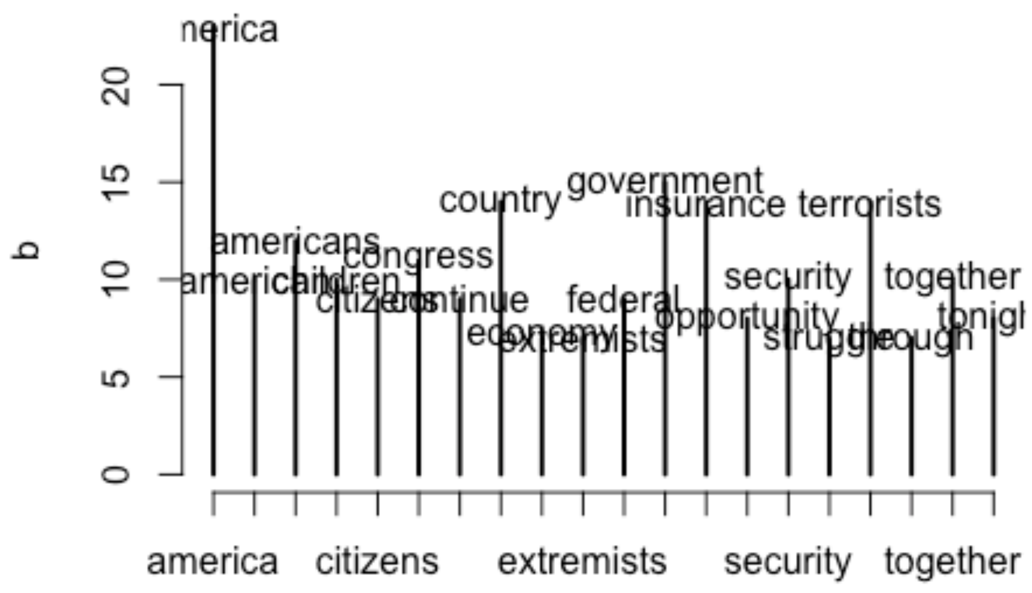
```
a = table(words_vec[[217]])
mean(a)

## [1] 3.583665

sd(a)

## [1] 14.34283

c = names(a)
d = as.numeric(sapply(c,nchar))
b = a[which(d>mean(d))]
b = b[b>mean(a)+0.2*sd(a)]
plot(b)
text(b, lab=row.names(b))
```



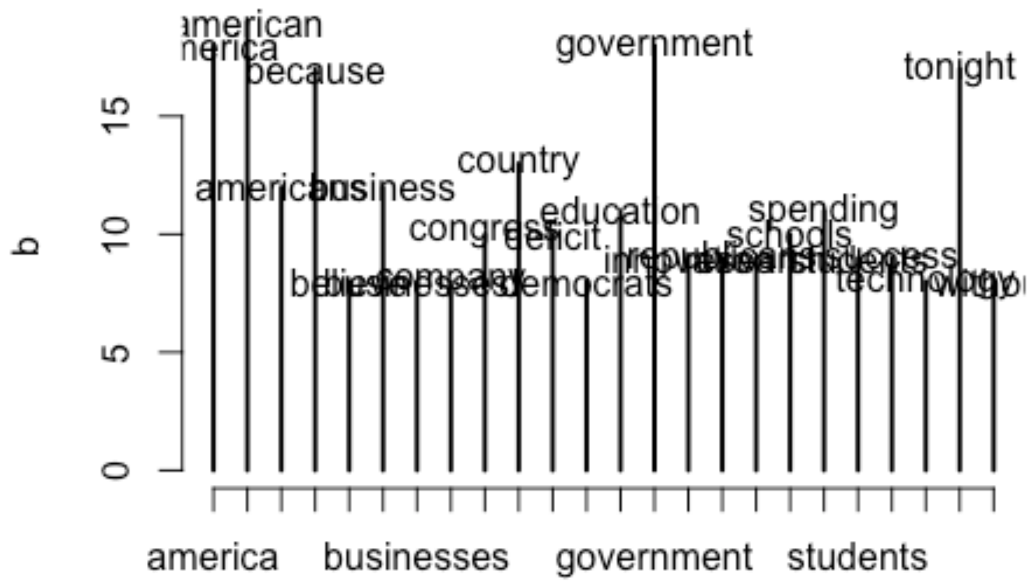
```
a = table(words_vec[[221]])
mean(a)

## [1] 4.057856

sd(a)

## [1] 15.63729

c = names(a)
d = as.numeric(sapply(c,nchar))
b = a[which(d>mean(d))]
b = b[b>mean(a)+0.2*sd(a)]
plot(b)
text(b, lab=row.names(b))
```



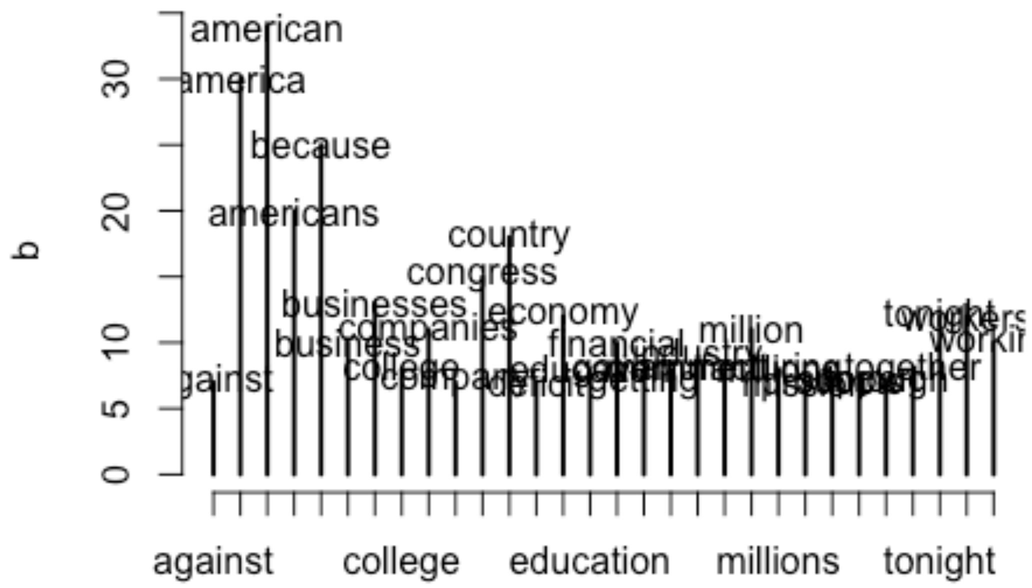
```
a = table(words_vec[[222]])
mean(a)

## [1] 4.059666

sd(a)

## [1] 14.12833

c = names(a)
d = as.numeric(sapply(c,nchar))
b = a[which(d>mean(d))]
b = b[b>mean(a)+0.2*sd(a)]
plot(b)
text(b, lab=row.names(b))
```



```r
print("Above was 4 graphs on frequent words in the speech by J.W Bush and
Obama. I filtered only long words and the frequency has to be large. The
first two are by Bush; the latter 2 are by Obama. Despite from America and
American, we can easily tell that there's a difference in their focus. J.W
Bush has been using military, security and terrorists frequently. Obama has
been using education, business, innovation, technology and companies more
frequently.
      The reason could be Bush has a main focus on anti-terrorists. History
also proved that this stat is correct, Bush vote for the war and Obama
against it. When Bush was in duty, 9/11 happened. When Obama was in duty, he
had to deal with the consequences of financial crisis, so his main focus will
be on business, thus the word business appears more.")
```

```
## [1] "Above was 4 graphs on frequent words in the speech by J.W Bush and
Obama. I filtered only long words and the frequency has to be large. The
first two are by Bush; the latter 2 are by Obama. Despite from America and
American, we can easily tell that there's a difference in their focus. J.W
Bush has been using military, security and terrorists frequently. Obama has
been using education, business, innovation, technology and companies more
frequently.\n      The reason could be Bush has a main focus on anti-
terrorists. History also proved that this stat is correct, Bush vote for the
war and Obama against it. When Bush was in duty, 9/11 happened. When Obama
```

```
was in duty, he had to deal with the consequences of financial crisis, so his
main focus will be on business, thus the word business appears more."
```