STAT 4410/8416 Homework 3

Mamoundou Dramera

Due on Tuesday March 22, 2022

1. a)

```
lAddress = readLines("lincoln-last-speech.txt",warn =FALSE)
substr(lAddress[1],1,70)
```

```
## [1] "We meet this evening, not in sorrow, but in gladness of heart. The eva"
```

b)

```
library(stringr)
library(dplyr)
vWord <- unlist(str_extract_all(tolower(lAddress), '([\\w-]+)'))
head(vWord)
```

```
## [1] "we"      "meet"    "this"    "evening" "not"     "in"
```

c)

```
library(tm)
sWord = stopwords("en")
head(sWord)
```

```
## [1] "i"      "me"     "my"     "myself" "we"     "our"
```

d)

```
clearWord =  data.frame(word=vWord[ !as.vector(vWord %in% sWord)])
head(clearWord)
```

```
##         word
## 1       meet
## 2    evening
## 3     sorrow
## 4   gladness
## 5      heart
## 6 evacuation
```

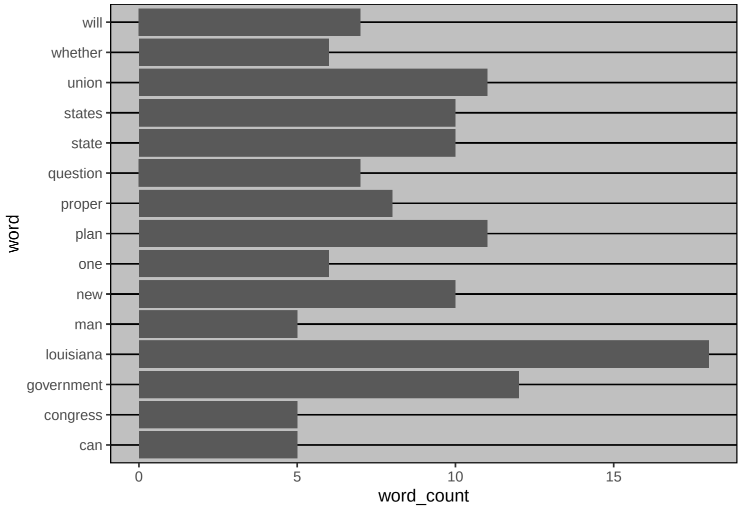
e)

```
fword = data.frame(clearWord %>% count(word)%>% arrange(desc(n)))
names(fword) = c("word", "word_count")
head(fword, 5)
```

```
##          word word_count
## 1  louisiana         18
## 2 government         12
## 3       plan         11
## 4      union         11
## 5        new         10
```

f)

```
library(ggplot2)
library(ggthemes)
ggplot(fword[0:15,], aes(x = word, y = word_count)) + geom_bar(stat="identity") +
   coord_flip() + theme_excel()
```

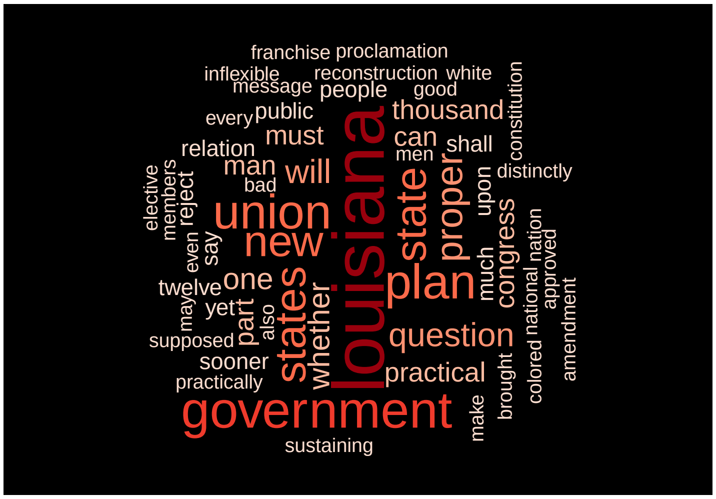


g) Explaination:

the `+coord_flip()` allows to see all the words on the legend of X-Axis which is overlapped.

h) [1 pt] The plot in question (f) uses bar plot to display the data. Can you think of another plot that delivers the same information but looks much simpler? Demonstrate your answer by generating such a plot.

```
library(wordcloud)
wCorpus = Corpus(VectorSource(clearWord$word))
par(bg = "black")
wordcloud(wCorpus, max.words=300,
   random.order = FALSE, rot.per = 0.35,
   random.color = FALSE, colors=brewer.pal(8, 'Reds'))
```



i) a)

```r
stopWordsCount = length(sort(table(sWord)))
stopWordsCount
```

```
## [1] 174
```

b)

```r
cText = vWord[!(vWord %in% c(sWord,'ll', 've'))]
lAddressCount = length(sort(table(cText), decreasing = TRUE))
lAddressCount
```

```
## [1] 540
```

c)

```r
leng = (stopWordsCount / lAddressCount) * 100
leng
```

```
## [1] 32.22222
```

d) [1 pt] Explain in your own words what does the percentage indicate in this context?

2. **

a)

```r
vText = c('google','logo','dig', 'blog', 'boogie' )
pattern = 'o?go?'
gsub(pattern, '.', vText)
```

```
## [1] "..le"  "l."    "di."   "bl."   "bo.ie"
```

b)

```r
vPhone = c('874','6783','345345', '32120', '468349', '8149674' )
pattern_b ='^\\d{5,6}$'
gsub(pattern_b,'found', vPhone)
```

```
## [1] "874"     "6783"    "found"   "found"   "found"   "8149674"
```

c)

```
myText = "#y%o$u @g!o*t t9h(e) so#lu!tio$n c%or_r+e%ct"
pattern ='[^a-zA-z ]|_'
gsub(pattern,'', myText)
```

```
## [1] "you got the solution correct"
```

d) [2 pts]

```
myText = "Each of the three and four character words will be gone now"
pattern = '\\b\\w{3,4}\\b'
gsub(pattern,'...', myText)
```

```
## [1] "... of ... three ... ... character words ... be ... ..."
```

e)

```
bigText = 'There are four 20@20 numbers hid989den in the 500 texts'
library(stringr)
pattern = '[^a-zA-z0-9 ]|_'
bigText = gsub(pattern,'', bigText)
pattern1 = "\\-*\\d+\\.*\\d*"
str_extract_all(bigText, pattern1)
```

```
## [[1]]
## [1] "2020" "989"  "500"
```

f)

```
myText = 'The salries are reported (in millions) for every company.'
library(stringr)
pattern_f = "(?<=\\().+?(?=\\))"
ext_str = str_extract_all(myText, pattern_f)
ext_str
```

```
## [[1]]
## [1] "in millions"
```

```
str_count(ext_str,"\\w+")
```

```
## [1] 2
```

g)

```
‘‘‘r
myText = c("H_bill.xls", "Big_H_pay.xls", "Use_case_fine-book.pdf")
pattern = ’[^_]+\\.’
myString = unlist(str_extract(myText, pattern))
sub("\\.","",myString)
‘‘‘

‘‘‘
## [1] "bill"      "pay"       "fine-book"
‘‘‘
```

h)

```
‘‘‘r
myText = ’Received 10 apples with 200ml water at 8pm with 15 lb meat and 2lb salt’
pattern = ’\\d+(ml| *lb)’
myString = unlist(str_extract_all(myText, pattern))
myNumb = str_extract(myString,"\\d+")
myNumb
‘‘‘

‘‘‘
## [1] "200" "15"  "2"
‘‘‘
```

i)

```
‘‘‘r
myText = ’Math symbols are $written$ in $between$ dollar $signs$’
pattern = "(?<=\\$)([a-zA-Z-]+)(?=\\$)"
str = str_extract_all(myText,pattern)
str
‘‘‘

‘‘‘
## [[1]]
## [1] "written" "between" "signs"
‘‘‘

‘‘‘r
lengths(gregexpr("(\\S+)", str))
‘‘‘

‘‘‘
## [1] 3
‘‘‘
```

j)

```r
myText =  c("equation1: 21-12=9, equation2 is: 2*3=6, do not extract 2w3=6")
ext =  gregexpr("\\b\\w+(?:[+-\\*]\\w+)+=\\w+\\b", myText)
regmatches(myText, ext)
```

```
## [[1]]
## [1] "21-12=9" "2*3=6"
```

k)

```r
myText = 'there are five wizard boxing matches to be judged'
pattern = "[a-z]"
cExtracted = str_extract_all(myText, pattern)
myLetters = unlist(cExtracted)
letters %in% myLetters
```

```
##  [1]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE
## [13]  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [25] FALSE  TRUE
```

```
sum(letters %in% myLetters)
```

```
## [1] 21
```

```
letters[!letters %in% myLetters]
```

```
## [1] "k" "l" "p" "q" "y"
```

3. Extracting data from the web: Our plan is to extract data from web sources. This includes email addresses, phone numbers or other useful data. The function `readLines()` is very useful for this purpose.

a)

```
myText = readLines('https://www.unomaha.edu/college-of-arts-and-sciences/mathematics/about-us/directory/
head(myText)
```

```
## [1] " "
## [2] "<!DOCTYPE html> "
## [3] "<!--[if lt IE 7]>      <html class=\"no-js lt-ie9 lt-ie8 lt-ie7\" lang=\"en-US\" xml:lang=\"en\
## [4] "<!--[if IE 7]>         <html class=\"no-js lt-ie9 lt-ie8\" lang=\"en-US\" xml:lang=\"en\"> <![e
## [5] "<!--[if IE 8]>         <html class=\"no-js lt-ie9\" lang=\"en-US\" xml:lang=\"en\"> <![endif]--
## [6] "<!--[if gt IE 8]><!--> <html class=\"no-js\" lang=\"en-US\" xml:lang=\"en\"><!--<![endif]-->  "
```

b)

```
pattern = 'http:[^"]*'
disp = str_extract_all(myText, pattern)
unlist(disp)
```

```
##  [1] "http://www.w3.org/2000/svg"
##  [2] "http://www.w3.org/2000/svg"
##  [3] "http://www.w3.org/2000/svg"
##  [4] "http://www.w3.org/2000/svg"
##  [5] "http://www.w3.org/2000/svg"
##  [6] "http://www.w3.org/2000/svg"
##  [7] "http://www.w3.org/2000/svg"
##  [8] "http://unobookstore.com/"
##  [9] "http://buffettinstitute.nebraska.edu"
## [10] "http://waterforfood.nebraska.edu"
```

c)

```
pattern = '[_a-z0-9-]+(\\.[_a-z0-9-]+)*\\@[_a-z0-9-]+\\.[_a-z0-9-]+'
disp = str_extract_all(myText, pattern)
unlist(disp)
```

```
##  [1] "mbaccouch@unomaha.edu"        "mbaccouch@unomaha.edu"
##  [3] "rbrusky@unomaha.edu"          "rbrusky@unomaha.edu"
##  [5] "xycheng@unomaha.edu"          "xycheng@unomaha.edu"
##  [7] "jeffreydepue@unomaha.edu"     "jeffreydepue@unomaha.edu"
##  [9] "elder@unomaha.edu"            "elder@unomaha.edu"
## [11] "sfrom@unomaha.edu"            "sfrom@unomaha.edu"
## [13] "keithgallagher@unomaha.edu"   "keithgallagher@unomaha.edu"
## [15] "jjhazuka@unomaha.edu"         "jjhazuka@unomaha.edu"
## [17] "dholley@unomaha.edu"          "dholley@unomaha.edu"
## [19] "yinghu@unomaha.edu"           "yinghu@unomaha.edu"
## [21] "ninfante@unomaha.edu"         "ninfante@unomaha.edu"
## [23] "nkass@unomaha.edu"            "nkass@unomaha.edu"
## [25] "blove@unomaha.edu"            "blove@unomaha.edu"
## [27] "mmajumder@unomaha.edu"        "mmajumder@unomaha.edu"
## [29] "vmatache@unomaha.edu"         "vmatache@unomaha.edu"
## [31] "michaelmatthews@unomaha.edu"  "michaelmatthews@unomaha.edu"
## [33] "lmcfee@unomaha.edu"           "lmcfee@unomaha.edu"
## [35] "kenzimedeiros@unomaha.edu"    "kenzimedeiros@unomaha.edu"
## [37] "lindarau@unomaha.edu"         "lindarau@unomaha.edu"
## [39] "prault@unomaha.edu"           "prault@unomaha.edu"
## [41] "jrech@unomaha.edu"            "jrech@unomaha.edu"
## [43] "meriley@unomaha.edu"          "meriley@unomaha.edu"
## [45] "jrogers@unomaha.edu"          "jrogers@unomaha.edu"
## [47] "aroslanowski@unomaha.edu"     "aroslanowski@unomaha.edu"
## [49] "vrykov@unomaha.edu"           "vrykov@unomaha.edu"
## [51] "nsahu@unomaha.edu"            "nsahu@unomaha.edu"
## [53] "gsand@unomaha.edu"            "gsand@unomaha.edu"
## [55] "larissaschroeder@unomaha.edu" "larissaschroeder@unomaha.edu"
## [57] "aswift@unomaha.edu"           "aswift@unomaha.edu"
## [59] "kluhing@unomaha.edu"          "kluhing@unomaha.edu"
## [61] "dvelcsov@unomaha.edu"         "dvelcsov@unomaha.edu"
## [63] "ftorresvitor@unomaha.edu"     "ftorresvitor@unomaha.edu"
## [65] "congwang@unomaha.edu"         "congwang@unomaha.edu"
## [67] "ecook@unomaha.edu"            "ecook@unomaha.edu"
## [69] "heatherlarson@unomaha.edu"    "heatherlarson@unomaha.edu"
## [71] "cteller@unomaha.edu"          "cteller@unomaha.edu"
## [73] "sdowning@unomaha.edu"         "sdowning@unomaha.edu"
## [75] "jheidel@unomaha.edu"          "jheidel@unomaha.edu"
## [77] "maloney@cox.net"              "maloney@cox.net"
## [79] "lstephens@unomaha.edu"        "lstephens@unomaha.edu"
## [81] "zhenyuanwang@unomaha.edu"     "zhenyuanwang@unomaha.edu"
## [83] "unomathematics@unomaha.edu"   "unomathematics@unomaha.edu"
```

e)

```r
pattern = '\\(*\\d{3}\\)*( |-)*\\d{3}\\.*( |-)*\\d{4}'
pList = str_extract_all(myText, pattern)
unlist(pList)
```

```
## [1] "1645109341"   "1645109340"   "1645109342"   "2130042793"   "402-554.6325"
## [6] "0893001933"   "1645109352"   "1645109352"
```

f)

```
gText = readLines("https://ggplot2-book.org/individual-geoms.html",warn=F)
pattern = 'geom_\\w+'
gList = str_extract_all(gText, pattern)
g.List = unique(unlist(gList))
g.List
```

```
##  [1] "geom_ribbon"  "geom_area"    "geom_bar"     "geom_path"    "geom_line"
##  [6] "geom_point"   "geom_polygon" "geom_tile"    "geom_rect"    "geom_raster"
## [11] "geom_text"    "geom_smooth"  "geom_boxplot" "geom_violin"
```

```
length(g.List)
```

```
## [1] 14
```

4. a) [

```
data = read.csv("bigDataSample.csv")
dat = data[,grepl("human",colnames(data))]
head(dat)
```

```
##   var_human_1_g var_human_1_p var_human_1_b var_human_1_e var_human_1_n
## 1      18.99545            21             1    21.6321136      26.03268
## 2      15.02303            34             3     0.3838458      26.92529
## 3      37.44410            28             2    33.4801022      39.30039
## 4      36.33714            26             2     2.8761174      33.75177
## 5      21.06330            25             1     3.1657313      26.19248
## 6      16.52637            35             2     5.3108922      25.07192
```

b)

```
colClean = function(dat){colnames(dat) <-gsub("var_human_1_","",colnames(dat)); dat}
newDat = colClean(dat)
head(newDat)
```

```
##          g  p b          e        n
## 1 18.99545 21 1 21.6321136 26.03268
## 2 15.02303 34 3  0.3838458 26.92529
## 3 37.44410 28 2 33.4801022 39.30039
## 4 36.33714 26 2  2.8761174 33.75177
## 5 21.06330 25 1  3.1657313 26.19248
## 6 16.52637 35 2  5.3108922 25.07192
```

c)

```
library(dplyr)
sdat = newDat%>%
   group_by(b)%>%
   summarise_all(funs(mean))
kable(sdat,digits = 2)
```

b	g	p	e	n
0	28.75	23.76	12.21	29.44
1	22.48	25.28	10.42	29.34
2	23.85	24.95	9.62	30.63
3	23.81	25.41	10.48	30.25

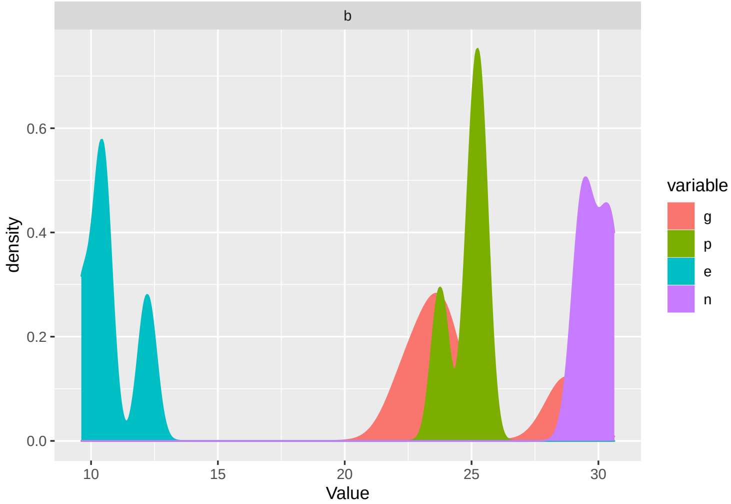
d)

```
library(reshape2)
mdat = melt(data.frame(sapply(sdat, as.double)), 'b')
head(mdat)
```

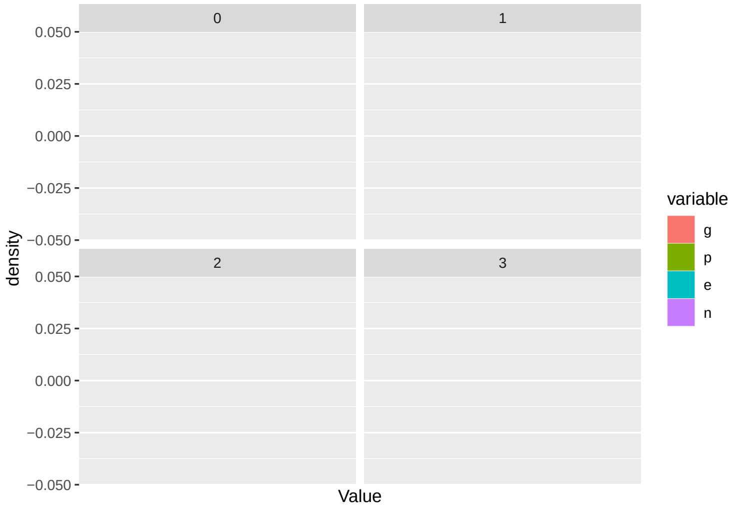
```
##   b variable    value
## 1 0        g 28.74877
## 2 1        g 22.47859
## 3 2        g 23.85395
## 4 3        g 23.81182
## 5 0        p 23.75862
## 6 1        p 25.28302
```

e)

```
library("ggplot2")
ggplot(mdat, aes(x=value, fill=variable, colour=variable)) +
geom_density() + facet_wrap(~'b') +
xlab("Value")
```



```
ggplot(mdat, aes(x=value, fill=variable, colour=variable)) +
geom_density() + facet_wrap(~b) +
xlab("Value")
```



f)

```
library(data.table)
data.1 = fread("bigDataSample.csv", nrows = 0, header = TRUE)
Col = which(str_detect(colnames(data.1), '.*human.*'))
data.2 = fread("bigDataSample.csv", select = Col)
head(data.2)
```

```
##    var_human_1_g var_human_1_p var_human_1_b var_human_1_e var_human_1_n
## 1:      18.99545            21             1    21.6321136      26.03268
## 2:      15.02303            34             3     0.3838458      26.92529
## 3:      37.44410            28             2    33.4801022      39.30039
## 4:      36.33714            26             2     2.8761174      33.75177
## 5:      21.06330            25             1     3.1657313      26.19248
## 6:      16.52637            35             2     5.3108922      25.07192
```