Answers Homework Assignment 1

Name

date

Let's take a look a set of Dutch speeches in English from the EUSpeech dataset. Use setwd() to set the working directory to the folder that contains Dutch speeches in the file speeches_nl.csv. Read in the speeches as follows:

```
Sys.setlocale(locale = "en_US.UTF-8")
```

```
## [1] "en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8"
```

```
library(foreign)
speeches <- read.csv(file = "speeches_nl.csv",
                     header = TRUE,
                     stringsAsFactors = FALSE,
                     sep = ",")
```

Load the `stringr` library, and do the following:

1) Write some code to take out the p-tags.

```
library(stringr)
speeches$text <- str_replace_all( speeches$text, "<.+?>", "" )
```

2) Write some code to take out all individual digits / numbers in the first speech (for example, the number 20 should stored as a 2 and a 0 separately). Print the total sum of these numbers. What is this total?

```
numbers <- str_extract_all(speeches$text, "\\d", simplify = TRUE)
total <- sum(as.numeric(numbers[1,]), na.rm = T)
print(total)
```

```
## [1] 26
```

3) Write some code to display i) the names of the speakers, and ii) the number of speeches they delivered.

```
table(speeches$speaker)
```

```
##
## J.P. Balkenende        M. Rutte
##              25             107
```

4) Write some code to count the number of times in each speech the speakers mentions "I", and save this as a variable called *self.references* in the speeches dataframe.

```
speeches$self.references <- str_count(speeches$text, " I ")
```

We have currently read in the speeches as a variable in a dataframe using the `foreign` library, but we could have done so using `quanteda` and `readtext` as well, which read in the speeches as a `corpus` object. Although a typical workflow of reading in text files involves one set of functions it is useful to know that you can go back and forth between both approaches as well.

```
library(quanteda)
```

```
## Warning: package 'quanteda' was built under R version 3.5.2
```

```
speeches.corpus <- corpus(speeches$text)
str(speeches.corpus)
```

```
## List of 4
##  $ documents:'data.frame':   132 obs. of  1 variable:
##   ..$ texts: chr [1:132] "Ladies and gentlemen,It is an honour to be here today to introduce the ther
##  $ metadata :List of 2
##   ..$ source : chr "/Users/hjms/Documents/Teaching/CEU/2019/Assignments/Assignment_1/* on x86_64 by l
##   ..$ created: chr "Mon May 13 14:22:55 2019"
##  $ settings :List of 12
##   ..$ stopwords          : NULL
##   ..$ collocations       : NULL
##   ..$ dictionary         : NULL
##   ..$ valuetype          : chr "glob"
##   ..$ stem               : logi FALSE
##   ..$ delimiter_word     : chr " "
##   ..$ delimiter_sentence : chr ".!?"
##   ..$ delimiter_paragraph: chr "\n\n"
##   ..$ clean_tolower      : logi TRUE
##   ..$ clean_remove_digits: logi TRUE
##   ..$ clean_remove_punct : logi TRUE
##   ..$ units              : chr "documents"
##   ..- attr(*, "class")= chr [1:2] "settings" "list"
##  $ tokens   : NULL
```

You know have read in the speeches as a `corpus` object in `quanteda`, and you can use its functions. Familiarize yourself with these functions by going through the online tutorial

5) Count the number of tokens *speeches.corpus* (make sure you remove punctuation), and save them as a *n.tokens* variable in the speeches dataframe. Also generate a variable *reference.ratio* which is the number of self references divided by the number of tokens. Print the mean *reference.ratio*

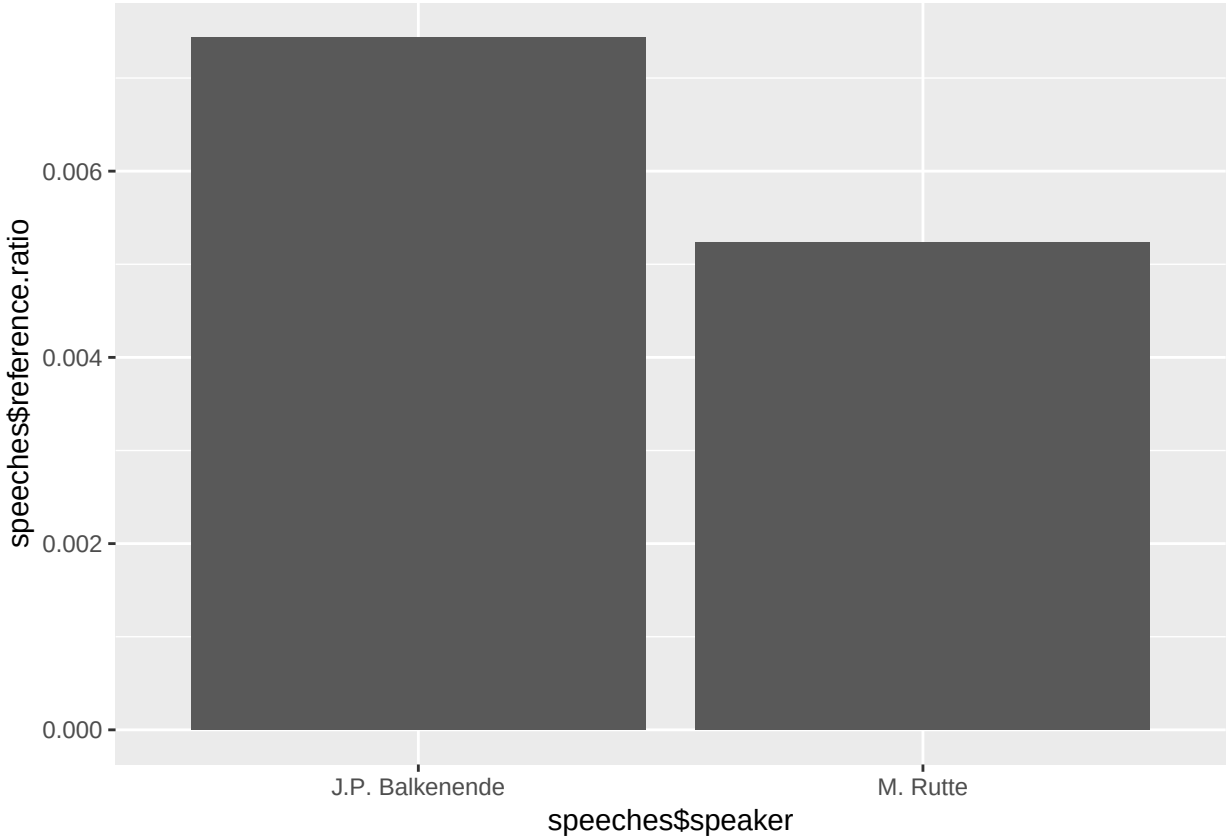
```
speeches$n.tokens <- ntoken(speeches.corpus, remove_punct = TRUE)
speeches$reference.ratio <- speeches$self.references / speeches$n.tokens
print(speeches$reference.ratio)
```

```
##   [1] 0.011585807 0.006024096 0.007421150 0.011049724 0.013937282
##   [6] 0.010557572 0.009836066 0.006009615 0.006024096 0.001706485
##  [11] 0.015280136 0.008321775 0.003361345 0.009302326 0.006493506
##  [16] 0.010025063 0.005181347 0.006322445 0.007407407 0.003937008
##  [21] 0.007051282 0.002862049 0.005936675 0.006132989 0.004205214
##  [26] 0.005102041 0.005494505 0.005361930 0.005347594 0.005714286
##  [31] 0.000000000 0.005673759 0.005943536 0.005235602 0.003252033
##  [36] 0.000000000 0.003300330 0.003807107 0.005514706 0.005256242
##  [41] 0.003764115 0.000000000 0.009756098 0.007421150 0.001760563
##  [46] 0.003110420 0.002466091 0.005000000 0.018480493 0.000000000
##  [51] 0.002894356 0.004938272 0.003787879 0.014598540 0.004385965
##  [56] 0.006688963 0.000000000 0.002857143 0.001404494 0.000000000
##  [61] 0.001876173 0.004434590 0.000000000 0.006684492 0.000000000
##  [66] 0.000000000 0.000000000 0.006644518 0.001697793 0.002512563
##  [71] 0.001919386 0.007014028 0.004916421 0.004854369 0.008385744
##  [76] 0.005976096 0.003369840 0.005228758 0.003880983 0.009720535
##  [81] 0.005305040 0.006038647 0.007900677 0.006196378 0.001336898
##  [86] 0.009790210 0.016766467 0.013048636 0.004854369 0.008169935
##  [91] 0.006069803 0.009302326 0.001515152 0.007002801 0.005943536
##  [96] 0.005172414 0.003215434 0.000000000 0.000000000 0.003851091
## [101] 0.006476684 0.003764115 0.003454231 0.004842615 0.014669927
## [106] 0.006479482 0.000000000 0.002840909 0.001410437 0.001862197
## [111] 0.000000000 0.001529052 0.006675567 0.000000000 0.006578947
```

```
## [116] 0.014084507 0.004866180 0.006289308 0.004321521 0.005228758
## [121] 0.005161290 0.005642633 0.009720535 0.005305040 0.005305040
## [126] 0.006196378 0.013011152 0.006329114 0.009409305 0.015706806
## [131] 0.010703364 0.012910798
```

6) Plot the average *reference.ratio* for both speakers using a bar chart. You may use either base `R` or `ggplot2`.

```
library(ggplot2)
g <- ggplot(speeches, aes(x = speeches$speaker,
                          y = speeches$reference.ratio))
g <- g + stat_summary(fun.y="mean", geom="bar")
print(g)
```



7) Generate a *dfm.speeches* object which is the dfm from the *speeches.corpus* object. Print the number of features.

```
dfm.speeches <- dfm(speeches.corpus)
nfeat(dfm.speeches)
```

```
## [1] 8359
```

8) Use tf-idf weighting on the dfm. Print the top 10 features of the 20th speech in the dataframe.

```
tf.idf.dfm.speeches <- dfm_tfidf(dfm.speeches)
topfeatures(tf.idf.dfm.speeches [20,])
```

```
##      africa     african  harmonised      states    progress         mdg
##    6.087420    5.831657    4.241148    4.062321    3.146061    3.037028
##        lack     picture development      donors
```

```
##     3.037028     3.037028     2.635996     2.550952
```

9) Use the textstat_lexdiv() function in `quanteda` *dfm.speeches* to obtain the TTR for all speeches, and save these as a *ttr* variable in speeches dataframe.

```
dfm.speeches <- dfm(speeches.corpus, remove = stopwords('en'))
lexdiv <- textstat_lexdiv(dfm.speeches)
speeches$ttr <- lexdiv$TTR
```

10) Plot the average *ttr* for both speakers using a bar chart. You may use either base `R` or `ggplot2`.

```
library(ggplot2)
g <- ggplot(speeches, aes(x = speeches$speaker, y = speeches$ttr))
g <- g + stat_summary(fun.y="mean", geom="bar")
print(g)
```