CTA Example

Christopher Barrie

2022-03-30

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button at the top of this panel, a document will be generated that includes both content as well as the output of any embedded R code chunks within the document.

## Example

We generally include all of the libraries required at the top of the script as so.

library(haven)  
library(quanteda)

## Package version: 3.2.1  
## Unicode version: 13.0  
## ICU version: 67.1

## Parallel computing: 8 of 8 threads used.

## See https://quanteda.io for tutorials and examples.

library(quanteda.textstats)  
library(ggplot2)

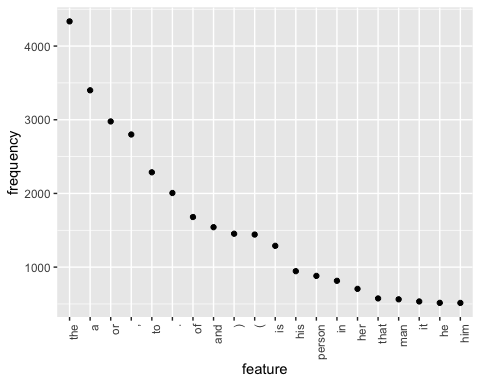
We can read the main “Motif” dataset from this dataset (stored in Stata .dta format) as follows.

folklore <- read\_dta("/Users/cbarrie6/Dropbox/Teaching/Edinburgh/teaching/CTA\_21-22/CTA-ED/data/assessment/Motif\_Master.dta")

## Including outputs

You can also embed outputs like plots that you generate from the data. For example, if I had decided to plot the highest frequency words in the motifs data, I might do the following.

folk\_dfm <- dfm(tokens(folklore$desc\_eng))  
  
folk\_dfm\_features <- textstat\_frequency(folk\_dfm, n = 20)  
  
# Sort by reverse frequency order  
folk\_dfm\_features$feature <- with(folk\_dfm\_features, reorder(feature, -frequency))  
  
ggplot(folk\_dfm\_features, aes(x = feature, y = frequency)) +  
 geom\_point() +   
 theme(axis.text.x = element\_text(angle = 90, hjust = 1))



## Static code rendering

Note that we can also add the eval = FALSE parameter to our code chunk, which means that the code won’t run when we knit the file: instead, it will just format the code in a visually appealing way and make it more legible.

If we did not want to include any of the output, and just the code, we could do the following, and include all of our code in one chunk, output as a Word document.

library(haven)  
library(quanteda)  
library(quanteda.textstats)  
library(ggplot2)  
  
# read in the data  
folklore <- read\_dta("/Users/cbarrie6/Dropbox/Teaching/Edinburgh/teaching/CTA\_21-22/CTA-ED/data/assessment/Motif\_Master.dta")  
  
# generate dfm from descriptions  
folk\_dfm <- dfm(tokens(folklore$desc\_eng))  
  
# get dfm features (word frequencies)  
folk\_dfm\_features <- textstat\_frequency(folk\_dfm, n = 20)  
  
# Sort by reverse frequency order  
folk\_dfm\_features$feature <- with(folk\_dfm\_features, reorder(feature, -frequency))  
  
# plot in descending order  
ggplot(folk\_dfm\_features, aes(x = feature, y = frequency)) +  
 geom\_point() +   
 theme(axis.text.x = element\_text(angle = 90, hjust = 1))

## Output formats

Finally, note that if you change the YAML metadata at the top of this document from output: word\_document to output: html\_document you will generate a nicely formatted html document. For this assignment, stick with output: word\_document to avoid any ELMA headaches…