Tutorial - Text data

Ariane Ducellier

2023-12-05

```
Load R packages.
library(corrplot)
## corrplot 0.92 loaded
library(RColorBrewer)
library(tm)
## Loading required package: NLP
library(wordcloud)
Make a word cloud
# Set number of colors and palette
pal = brewer.pal(6, "RdGy")
# Choose minimum frequency and the range of the size of the words
wordcloud("The objective of this course is to provide students with a comprehensive understanding of da
## Warning in tm_map.SimpleCorpus(corpus, tm::removePunctuation): transformation
## drops documents
## Warning in tm_map.SimpleCorpus(corpus, function(x) tm::removeWords(x,
## tm::stopwords())): transformation drops documents
                                  data creating
                                         shiny interactivity
                comprehensive typespackages cognition
                  students of fundamentals effective clouds make
               word insights
                     geographic
                                                  ក្ត ក្ត<mark>ុင្ខlear</mark> techniques tooltips
            better design a
                             scales de datadriven static communicatepopular stories de datadriven cover datadriven cover de datadriven cover de datadriven cover de datadriven cover de datadriven cover datadriven cover de datadriven cover datad
          explore
            focusthrough
 enabling 8
                                                                                                                   theory
                               colors learning USe based creed understanding
                                 objective course chart
visual objectives main base effectively engagement clues zoomingpanning visualize decisions networks

visualizations principles informed compelling coordinates
```

To use a list of words and their frequencies.

visualization

America law democracy O

policy unemployment Republicans

To read a text file and preprocess it, before doing the word cloud.

```
file = readLines("../data/syllabus.txt")
doc = Corpus(VectorSource(file))
doc = tm_map(doc, tolower)
## Warning in tm_map.SimpleCorpus(doc, tolower): transformation drops documents
doc = tm_map(doc, removePunctuation)
## Warning in tm_map.SimpleCorpus(doc, removePunctuation): transformation drops
## documents
doc = tm map(doc, removeNumbers)
## Warning in tm_map.SimpleCorpus(doc, removeNumbers): transformation drops
## documents
doc = tm_map(doc, removeWords, stopwords("english"))
## Warning in tm_map.SimpleCorpus(doc, removeWords, stopwords("english")):
## transformation drops documents
wordcloud(as.character(doc), scale=c(2, 0.5))
## Warning in tm_map.SimpleCorpus(corpus, tm::removePunctuation): transformation
## drops documents
## Warning in tm_map.SimpleCorpus(corpus, function(x) tm::removeWords(x,
## tm::stopwords())): transformation drops documents
  interactive Will
```

Make two word clouds

```
files = DirSource("../data/debate/")
data = Corpus(DirSource("../data/debate/"))
data = tm_map(data, tolower)
data = tm_map(data, removePunctuation)
data = tm_map(data, removeNumbers)
data = tm_map(data, removeWords, c(stopwords("english"), "biden", "trump"))
data = TermDocumentMatrix(data)
data = as.matrix(data)
colnames(data) = c("biden", "trump")
comparison.cloud(data, max.words=100, title.size=2, colors=c("blue", "red"))
## Warning in text.default(x1, y1, words[i], cex = size[i], offset = 0, srt =
## rotWord * : font metrics unknown for Unicode character U+2026
## Warning in comparison.cloud(data, max.words = 100, title.size = 2, colors =
## c("blue", : obamacare could not be fit on page. It will not be plotted.
## Warning in comparison.cloud(data, max.words = 100, title.size = 2, colors =
## c("blue", : judges could not be fit on page. It will not be plotted.
## Warning in comparison.cloud(data, max.words = 100, title.size = 2, colors =
## c("blue", : november could not be fit on page. It will not be plotted.
## Warning in comparison.cloud(data, max.words = 100, title.size = 2, colors =
## c("blue", : places could not be fit on page. It will not be plotted.
## Warning in comparison.cloud(data, max.words = 100, title.size = 2, colors =
## c("blue", : radical could not be fit on page. It will not be plotted.
## Warning in comparison.cloud(data, max.words = 100, title.size = 2, colors =
## c("blue", : never could not be fit on page. It will not be plotted.
## Warning in comparison.cloud(data, max.words = 100, title.size = 2, colors =
## c("blue", : also could not be fit on page. It will not be plotted.
## Warning in comparison.cloud(data, max.words = 100, title.size = 2, colors =
## c("blue", : ever could not be fit on page. It will not be plotted.
## Warning in comparison.cloud(data, max.words = 100, title.size = 2, colors =
## c("blue", : thing could not be fit on page. It will not be plotted.
## Warning in comparison.cloud(data, max.words = 100, title.size = 2, colors =
## c("blue", : gave could not be fit on page. It will not be plotted.
## Warning in comparison.cloud(data, max.words = 100, title.size = 2, colors =
## c("blue", : statement could not be fit on page. It will not be plotted.
```

```
together totally states united everybody people president knows number money people home american going doesn't going doesn't take able of can way now one guy tax true of Svote of covid your take of three didn't con think half see three didn't con think half okay really to know country forest really to know country forest call came of car closed million trump of military closed million trump of military con the control of the control of
```

Plot correlations between texts

```
data(crude)
data = tm_map(crude, content_transformer(tolower))
data = tm_map(data, removePunctuation)
data = tm_map(data, removeNumbers)
data = tm_map(data, removeWords, stopwords("english"))
data = TermDocumentMatrix(data)
data = as.matrix(data)
crf = cor(data)
corrplot(crf, method = c("ellipse"))
```

