WSP_R_Text_Analysis

Lizzie Jones

05/05/2021

R Markdown - WSP Text Analysis

```
# Practicing using Q11 (words to describe WSP)
#Clean text
head(words_df)
```

```
## # A tibble: 6 x 4
##   UniqueID_short SurveyType Word_num  Words
##            <int> <fct>      <chr>     <fct>
## 1              1 Proactive  Q11_word1 "One leg"
## 2              1 Proactive  Q11_word2 "Babies"
## 3              1 Proactive  Q11_word3 ""
## 4              2 Proactive  Q11_word1 "White"
## 5              2 Proactive  Q11_word2 "Long legs"
## 6              2 Proactive  Q11_word3 "A bird"
```

```
words_df$Words <- gsub("[^[:graph:]]", " ", words_df$Words) #get rid of non graphical characters
words_df$Words <- gsub("rt", "", words_df$Words)# Replace blank space ("rt")
words_df$Words <- gsub("[[:punct:]]", "", words_df$Words)# Remove punctuation
words_df$Words <- gsub("[ |\t]{2,}", "", words_df$Words)# Remove tabs
words_df$Words <- gsub("^ ", "", words_df$Words)# Remove blank spaces at the beginning
words_df$Words <- gsub(" $", "", words_df$Words)# Remove blank spaces at the end
words_df$Words <- tolower(words_df$Words)#convert all text to lower case

Corpus_words <- Corpus(VectorSource(words_df$Words))
Corpus_words <- tm_map(Corpus_words, removeNumbers)
Corpus_words <- tm_map(Corpus_words, removeWords, stopwords("english")) #removes common english stopwords
# Corpus_words <- tm_map(Corpus_words, removeWords, c("muffin"))  #You can specify words to remove
# Corpus_words <- tm_map(Corpus_words, PlainTextDocument)

#build a term-document matrix
library("tm")
TDM_words = tm::TermDocumentMatrix(Corpus_words, control = list(minWordLength = 1))
m = as.matrix(TDM_words)
v = sort(rowSums(m), decreasing = TRUE)
d = data.frame(word = names(v),freq=v)

# Create a wordcloud
wordcloud(Corpus_words, scale=c(5,0.5), max.words=100, random.order=FALSE, rot.per=0.25,
          use.r.layout=FALSE, colors=brewer.pal(8, "Dark2"))
```

Word frequency analysis

```
# Frequent word analysis
# We can find the words that appear at least 100 times by calling the findFreqTerms() function on the term.doc.ma
trix
HiFreq_words <- findFreqTerms(TDM_words, 100)
HiFreq_words
```

```
##  [1] "babies"      "white"       "long"        "bird"        "rare"
##  [6] "large"       "big"         "majestic"    "graceful"    "beautiful"
## [11] "elegant"     "tall"        "impressive"  "interesting"
```

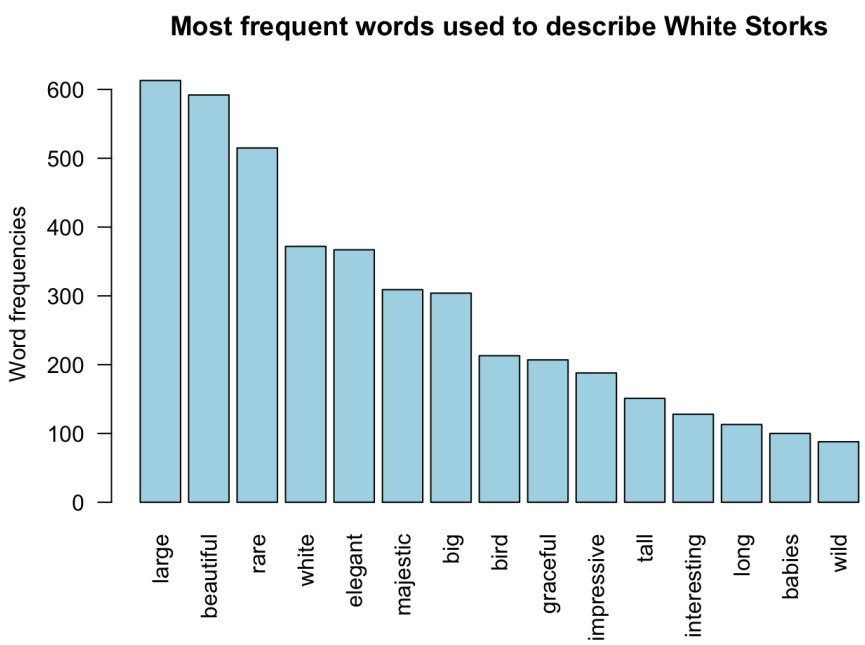
```
# Now you also see how associated a word is to another word or a list of words.
findAssocs(TDM_words, HiFreq_words, 0.4)
```

```
## $babies
## numeric(0)
##
## $white
## numeric(0)
##
## $long
## legs
## 0.51
##
## $bird
## numeric(0)
##
## $rare
## numeric(0)
##
## $large
## numeric(0)
##
## $big
## numeric(0)
##
## $majestic
## numeric(0)
##
## $graceful
## numeric(0)
##
## $beautiful
## numeric(0)
##
## $elegant
## numeric(0)
##
## $tall
## numeric(0)
##
## $impressive
## numeric(0)
##
## $interesting
## numeric(0)
```

```
# or, just compute word strength associations
findAssocs(TDM_words, "long", 0.5) # Looks like the word “long” and “legs” are very frequently associated (51% of the time)
```

```
## $long
## legs
## 0.51
```

```
barplot(d[1:15,]$freq, las = 2, names.arg = d[1:15,]$word,
        col ="lightblue", main ="Most frequent words used to describe White Storks",
        ylab = "Word frequencies")
```



Sentiment analysis

```
# Polarity / Sentiment Analysis
head(all_data$Q15_WSP_support_open)
```

```
## [1] It's always good to have as much diverse life as possible, and if they used to strive here, why not again? If handled correctly of course.
## [2]
## [3]
## [4] The more rewilding the better.
## [5] I absolutely support this, however it does concern me that they're reliant upon wetland ecosystems, which we have so little of. It's a natural follow on to the reintroduction of the beaver of course, and imagining beaver wetlands with white storks feeding within them is thrilling! But the widespread (government approved) support of free-living beavers seems to be at a much slower pace than the potential speed of breeding and dispersal of white storks. But I'm all for bringing appropriate species back, like the white stork, asap despite this.
## [6]
## 1916 Levels:  ...
```

```
# Clean the data
all_data$Q15_WSP_support_text <- gsub("[^[:graph:]]", " ", all_data$Q15_WSP_support_open) #get rid of non graphical characters
all_data$Q15_WSP_support_text <- gsub("^ ", "", all_data$Q15_WSP_support_text)# Remove blank spaces at the beginning
all_data$Q15_WSP_support_text <- gsub(" $", "", all_data$Q15_WSP_support_text)# Remove blank spaces at the end

# Reasons for support/not support WSP
class(all_data$Q15_WSP_support_text)
```

```
## [1] "character"
```

```
sentiment(get_sentences(all_data$Q15_WSP_support_text))
```

```
##       element_id sentence_id word_count   sentiment
##    1:          1           1         21  0.29459415
##    2:          1           2          5  0.35777088
##    3:          2           1         NA  0.00000000
##    4:          3           1         NA  0.00000000
##    5:          4           1          5  0.64398758
##   ---
## 4184:       3556           1         NA  0.00000000
## 4185:       3557           1         14 -0.32071349
## 4186:       3558           1          6  0.44907312
## 4187:       3559           1         14  0.05345225
## 4188:       3560           1          4  0.00000000
```

```
# There are lots more ways of doing this (see the QDAP package vignette). Here we take a cleaned character vector used earlier (i.e. words_df$Words) and compare its sentiment against a grouping variable (e.g. SurveyType)
# poldat_surveytype <- with(all_data, polarity(words_df$Words, all_data$SurveyType))
# plot(poldat)
```