Thoughts on Storks - Open-ended question analysis

Lizzie Jones[1]

26/07/2021

Text analysis and visualisations for open questions

This rMarkdown explores and analyses the open questions using text, work frequency and sentiment analysis techniques which are beyond the scope of NVivo (or where R is more effective).

The key open-ended questions include:

- Q8. How did you feel when you saw WS in the wild?
- Q9a. [If respondent has heard of the white stork project and its efforts to reintroduce white storks to southern England], please briefly summarise what you have heard.
- Q11a-c. Three words used to describe white storks
- Q15. Do you support the White Stork Project?
- Q16. What are yours views on the management of White Storks?
- Q17.1-17.13. Which (if any) methods of white stork project management would you support?

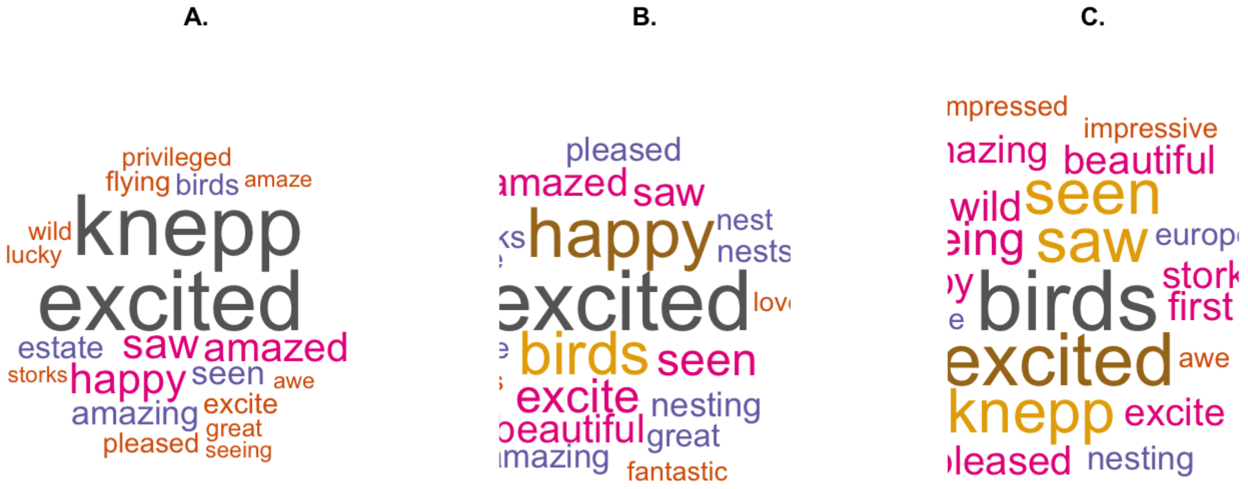
In this rMarkdown script I clean and explore each question in turn in the sections below, visualise the data and comment on any interesting findings.

Q8. How did you feel when you saw white stork in the wild?

Respondents were first asked if they had every seen a White Stork in the wild (Responses = Yes/No/) Investigating the question “How did you feel when you saw a white stork in the wild?” (N = 1123) Notes from codebook - No need for NVivo thematic analysis? Run some form of sentiment analysis

Question options - “Seen in the UK”, “Seen outside the UK”. Respondents were then asked if they wanted to explain/elaborate on their answer using an open-ended question.

```r
#Creating a non-local/local factor column for proximity to any WS release site
final_data <- mutate(final_data, SiteProximity =
                       ifelse(ReleaseSite == "No", "Not local", "Local"))
### Q8. How did you feel when you saw WS in the wild?
# Create words df to seperately clean, capitalise first letter etc
feel_df <- final_data %>%
  dplyr::select(UniqueID_all, SurveyType, Q8.WhereSeen, Q8.2_feelings)

# feel_syns <- qdap::synonyms(feel_df$Q8.2_feelings)

### Create two dataframes: one for each data collection for easier comparison
feelings_UK <- feel_df[which(feel_df$Q8.WhereSeen == "UK"),]
feelings_nonUK <- feel_df[which(feel_df$Q8.WhereSeen == "OutsideUK"),]
feelings_both <- feel_df[which(feel_df$Q8.WhereSeen == "Both"),]

# Create the custom function that will be used to clean the corpus: clean_coupus
clean_corpus_feel <- function(corpus){
  corpus <- tm_map(corpus, stripWhitespace)
  corpus <- tm_map(corpus, removePunctuation)
  corpus <- tm_map(corpus, content_transformer(tolower))
  corpus <- tm_map(corpus, removeWords, stopwords("en"))
  corpus <- tm_map(corpus, lemmatize_words)
    return(corpus)
}

# Seen in the wild within the UK (n=472)
corpus_seenUK <- Corpus(VectorSource(feelings_UK$Q8.2_feelings))
corpus_seenUK_clean <- clean_corpus_feel(corpus_seenUK)
corpus_seenUK_clean <- tm_map(corpus_seenUK_clean, removeWords,
                   c("they", "the", "the ", "its", "also", "I'm", "don't"," see", "can","see ",
"white",
                     "think", "one", "really", "bird", "see", "stork", "feel", "was"))
# Seen in the wild outside the UK (n=671)
corpus_seen_nonUK <- Corpus(VectorSource(feelings_nonUK$Q8.2_feelings))
corpus_seen_nonUK_clean <- clean_corpus_feel(corpus_seen_nonUK)
corpus_seen_nonUK_clean <- tm_map(corpus_seen_nonUK_clean, removeWords,
                   c("they", "the", "the ", "its", "also", "I'm", "don't"," see", "can","see ",
"white",
                     "think", "one", "really", "bird", "see", "stork", "feel", "was"))
# Seen in the wild in the UK and outside the UK (n=299)
corpus_seen_both <- Corpus(VectorSource(feelings_both$Q8.2_feelings))
corpus_seen_both_clean <- clean_corpus_feel(corpus_seen_both)
corpus_seen_both_clean <- tm_map(corpus_seen_both_clean, removeWords,
                   c("they", "the", "the ", "its", "also", "I'm", "don't"," see", "can","see ",
"white",
                     "think", "one", "really", "bird", "see", "stork", "feel", "was"))

## Plot image and save as a PDF using the file viewer
par(mfrow=c(1,3))
wordcloud(corpus_seenUK_clean, max.words=20, random.order=FALSE, rot.per=0,
          use.r.layout=FALSE, colors=brewer.pal(8, "Dark2"))
title(main="A.",col="black",font=2,line=-4)
wordcloud(corpus_seen_nonUK_clean, max.words=20, random.order=FALSE, rot.per=0,
          use.r.layout=FALSE, colors=brewer.pal(8, "Dark2"))
title(main="B.",col="black",font=2,line=-4)
wordcloud(corpus_seen_both_clean, max.words=20, random.order=FALSE, rot.per=0,
          use.r.layout=FALSE, colors=brewer.pal(8, "Dark2"))
title(main="C.",col="black",font=2,line=-4)
```


A.
privileged
flying
birds
amaze
wild
knepp
lucky
excited
estate
saw
amazed
storks
happy
seen
awe
amazing
excite
great
pleased
seeing
B.
pleased
amazed
saw
happy
nest
nests
excited
lov
birds
seen
excite
nesting
beautiful
great
mazing
fantastic
C.
mpressed
impressive
azing
beautiful
wild
seen
eing
saw
europ
y
birds
stork
first
e
excited
awe
knepp
excite
leased
nesting

Q11. What descriptive words do you associate with white storks?

For this question I have taken the 3 ‘words to describe’ columns and combined the words to create a new ‘long-format’ dataset to conduct preliminary word frequency analysis, word clouds etc. and also added SurveyType and ReleaseSite as possible grouping vairables. I then clean this dataframe (‘words_df’) by removing common English stop words, punctuation, blank spaces and convert all text to lower case. The word cloud is created using the ‘wordcloud’ package.

```
# Words used to describe WS - will probably combine into one column, labelled by Respondent ID
 and SurveyType
summary(final_data$Q11_word1)
```

```
##                     Large         Rare      Elegant    Beautiful          Big
##          670          296          208          152          151          145
##        White     Majestic         Bird         Tall   Beautiful      Graceful
##          121          116           95           80           62           62
##        large   Impressive    Majestic        Babies     Elegant          Wild
##           50           46           38           26           26           26
##       Large  Magnificent          Huge  Interesting  Magnificent    beautiful
##           24           24           23           21           21           20
##       Iconic      elegant     Stunning          big   Endangered  Impressive 
##           18           17           17           16           14           14
##     majestic      Unusual         rare        Regal     Striking    long legs
##           13           13           12           11           11           10
##     Amazing        Nature         Nice       Pretty  Spectacular        white
##            9            9            9            9            9            9
##       White        Beauty     Exciting    Graceful          Long   impressive
##            9            8            8            8            8            7
##   Large bird      Magical   Attractive     Chimneys       Exotic Fascinating 
##            7            7            6            6            6            6
##        Grace     graceful     Historic         huge      Lovely         Lucky
##            6            6            6            6            6            6
##     Magestic    Migratory    Rewilding         tall         Beak         Big 
##            6            6            6            6            5            5
##        Birth  Charismatic         Cool         Good        Leggy       Native
##            5            5            5            5            5            5
##     Peaceful    Stunning        Unique      Awesome         Hope       iconic
##            5            5            5            4            4            4
## Interesting     Long legs   Margarine       Natural    Symbolic       Ancient
##            4            4            4            4            4            3
##  Attractive      Awesome        babies      Babies!        Birds        Black
##            3            3            3            3            3            3
## Breathtaking         Cute     Elegance  Endangered   Fascinating        Grand
##            3            3            3            3            3            3
##        Lanky    Long beak  Long legged      (Other)
##            3            3            3          527
```

```
# summary(final_data$Q11_word2)
# summary(final_data$Q11_word3)

### Investigating relationships between 2 words
# https://uc-r.github.io/word_relationships

#### Investigating individual-word frequency
# Create a long-format words df (max 3 rows per responseID) to seperately clean, capitalise fir
st letter etc
words_df <- final_data %>%
  dplyr::select(UniqueID_short, SurveyType, Q11_word1, Q11_word2, Q11_word3) %>%
  pivot_longer(
   cols = starts_with("Q11_"),
   names_to = "Word_num",
   values_to = "Words",
   values_drop_na = TRUE)

# Clean text
words_df$Words <- tolower(words_df$Words)#convert all text to lower case
words_df$Words <- gsub("[^[:graph:]]", " ", words_df$Words) #get rid of non graphical character
s
words_df$Words <- gsub("a ", "", words_df$Words)# Remove single 'a' words
words_df$Words <- gsub("one", "", words_df$Words)# Remove one words
words_df$Words <- gsub("[[:punct:]]", "", words_df$Words)# Remove punctuation
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$Words <- gsub("[ |\t]{2,}", "", words_df$Words)# Remove tabs
# words_df$Words <- gsub(" ", "", words_df$Words)# Replace blank spaces with _
head(words_df, 50)
```

```
## # A tibble: 50 x 4
##    UniqueID_short SurveyType Word_num  Words
##             <int> <fct>      <chr>     <chr>
##  1              1 Proactive  Q11_word1 "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 "bird"
##  7              3 Proactive  Q11_word1 ""
##  8              3 Proactive  Q11_word2 ""
##  9              3 Proactive  Q11_word3 ""
## 10              4 Proactive  Q11_word1 "hope"
## # … with 40 more rows
```

```
# Create corpus, lemmatise and remove english stopwords
corpus_words <- Corpus(VectorSource(words_df$Words))
corpus_words <- tm_map(corpus_words, lemmatize_words)
corpus_words <- tm_map(corpus_words, removeWords, stopwords("english")) #removes common english
stopwords
corpus_words <- tm_map(corpus_words, removeWords,
                   c("they", "the", "the ", "its", "also", "I'm", "don't"," see", "can","see ",
"white",
                     "think", "one", "really", "bird", "see", "stork", "feel", "was"))
```

Word frequency analysis (Words used to describe White Storks)

Calculating the most frequent words and most common word associations. The bar plot and wordcloud indicate two ways of displaying the top 50 most frequently used words.

```r
# Frequent word analysis
#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)

#View frequencies
head(d, 50)
```

```
##                    word freq
## large             large  608
## beautiful     beautiful  592
## rare               rare  514
## elegant         elegant  367
## majestic       majestic  309
## big                 big  304
## graceful       graceful  207
## impressive   impressive  188
## tall               tall  149
## interest       interest  129
## baby               baby  110
## long               long  110
## wild               wild   88
## magnificent magnificent   87
## endanger       endanger   82
## beak               beak   68
## iconic           iconic   64
## excite           excite   60
## huge               huge   58
## legs               legs   53
## unusual         unusual   53
## stun               stun   49
## fascinate     fascinate   45
## noisy             noisy   42
## migratory     migratory   40
## native           native   40
## strike           strike   38
## exotic           exotic   37
## charismatic charismatic   36
## nest               nest   36
## amaze             amaze   36
## special         special   34
## awesome         awesome   32
## regal             regal   31
## black             black   31
## unique           unique   30
## chimney         chimney   28
## nature           nature   28
## magical         magical   27
## important     important   27
## pretty           pretty   26
## spectacular spectacular   26
## beauty           beauty   25
## natural         natural   24
## wetland         wetland   23
## attractive   attractive   23
## hope               hope   22
## rewilding     rewilding   21
## good               good   21
## leg                 leg   20
```

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

```
##  [1] "baby"       "long"       "rare"       "large"      "big"
##  [6] "majestic"   "graceful"   "beautiful"  "elegant"    "tall"
## [11] "impressive" "interest"
```

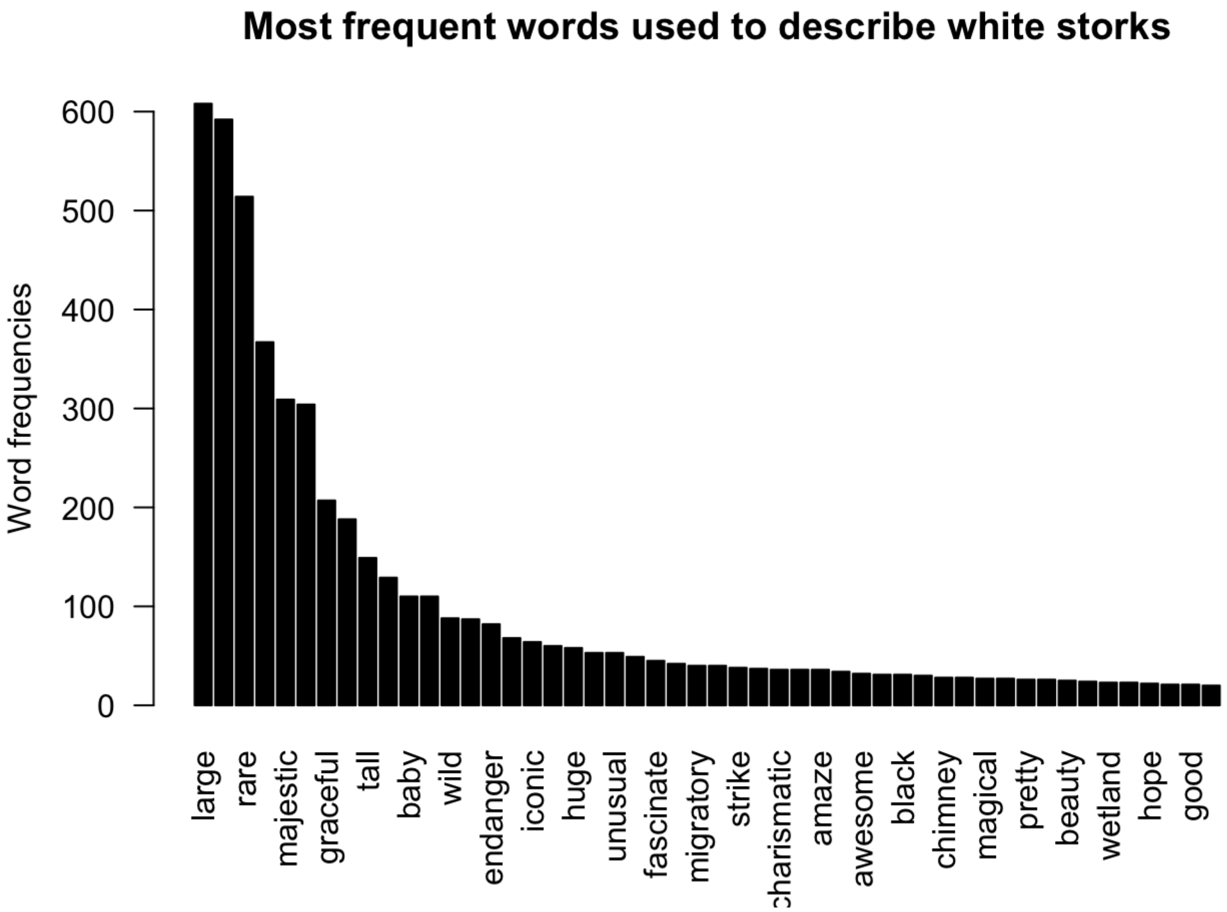
```
# Now you also see how associated a word is to another word or a list of words.
findAssocs(TDM_words, HiFreq_words, 0.4)
```

```
## $baby
## numeric(0)
##
## $long
## legs
## 0.59
##
## $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)
##
## $interest
## numeric(0)
```

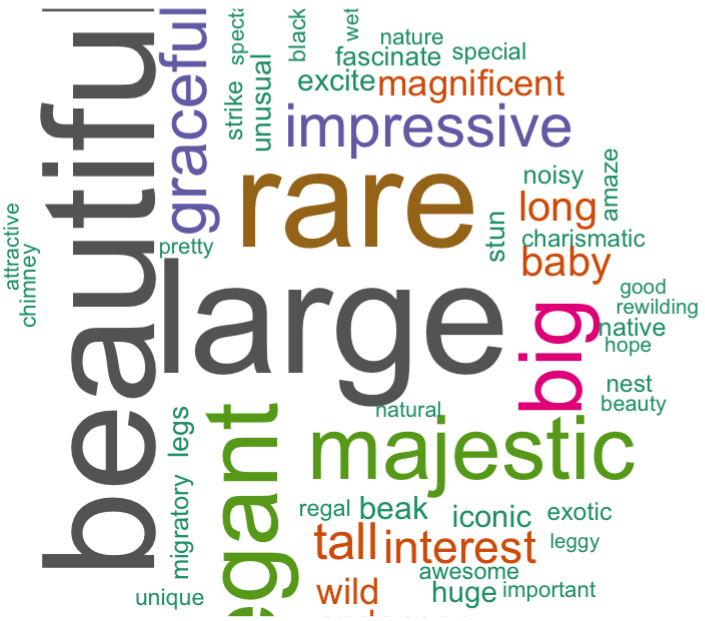
```
# or, just compute word strength associations
findAssocs(TDM_words, "long", 0.5) # Looks like the word “long” and “legs” are very frequently associated (51% of the time)
```

```
## $long
## legs
## 0.59
```

```
barplot(d[1:50,]$freq, las = 2, names.arg = d[1:50,]$word,
        col ="black", main ="Most frequent words used to describe white storks",
        ylab = "Word frequencies")
```



```
# Create a wordcloud
wordcloud(corpus_words, scale=c(5,0.5), max.words=50, random.order=FALSE, rot.per=0.25,
          use.r.layout=FALSE, colors=brewer.pal(10,"Dark2"))
```



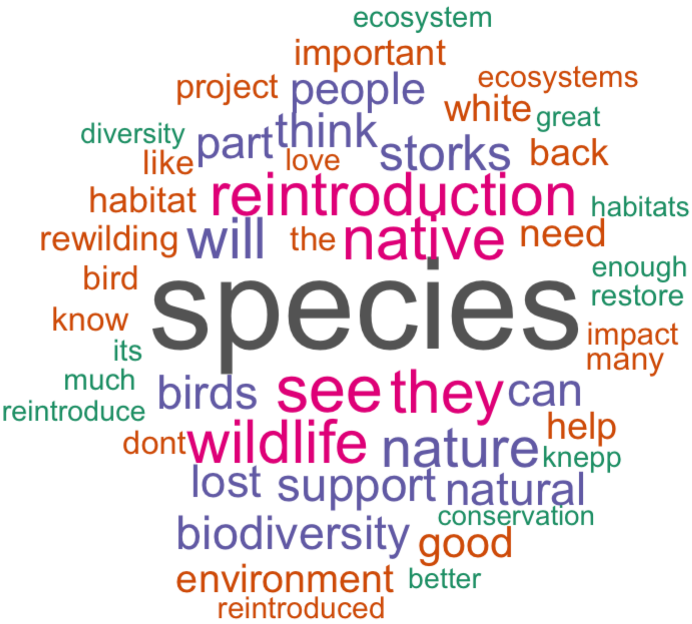
```
### Sentiment: Words to describe WS
# # please note that different methods may have different scales
# words_sentiment <- syuzhet::get_sentiment(corpus_words, method="syuzhet")
# # see the first row of the vector
# head(words_sentiment)
# # see summary statistics of the vector
# summary(words_sentiment)
```

Q15a. Support for white stork reintroduction

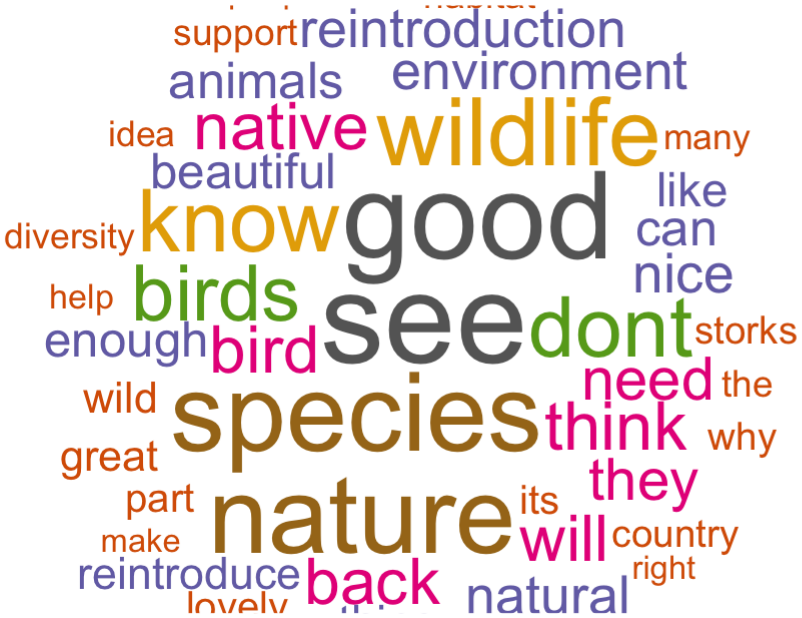
Question = Do you support the reintroduction of white Storks to southern England?

Calculate most frequent words and create a wordcloud.

```
### Q15. Do you support the WSP?

# Create words df to seperately clean, capitalise first letter etc
support_df <- final_data %>%
  dplyr::select(UniqueID_all, SurveyType, SiteProximity, Q15_WSP_support_open)

# Clean the data
support_df$Q15_WSP_support_text <- gsub("[^[:graph:]]", " ", support_df$Q15_WSP_support_open)
support_df$Q15_WSP_support_open <- gsub("[[:punct:]]", "", support_df$Q15_WSP_support_open)# Remove punctuation
support_df$Q15_WSP_support_text <- gsub(",", " ", support_df$Q15_WSP_support_open) # Remove commas after words
support_df$Q15_WSP_support_text <- gsub("'", "", support_df$Q15_WSP_support_open) # Remove apostrophes
support_df$Q15_WSP_support_text <- gsub("^ ", "", support_df$Q15_WSP_support_text) # Remove blank spaces at the beginning
support_df$Q15_WSP_support_text <- gsub(" $", "", support_df$Q15_WSP_support_text) # Remove blank spaces at the end

# # Create corpus, lemmatise and remove english stopwords
# corpus_support <- Corpus(VectorSource(support_df$Q15_WSP_support_text))
# corpus_support <- tm_map(corpus_support, lemmatize_words)
# corpus_support <- tm_map(corpus_support, removeWords,stopwords("english")) #removes common english stopwords
#
# #build a term-document matrix
# TDM_support = tm::TermDocumentMatrix(corpus_support, control = list(minWordLength = 1))
# m_support = as.matrix(TDM_support)
# v_support = sort(rowSums(m_support), decreasing = TRUE)
# d_support = data.frame(word = names(v_support),freq=v_support)
#
# # Reasons for support/not support WSP
# class(support_df$Q15_WSP_support_text)
# sentiment(get_sentences(support_df$Q15_WSP_support_text))

### Create two dataframes: one for each data collection for easier comparison
proact_support <- support_df[which(support_df$SurveyType == "Proactive"),]
natrep_support <- support_df[which(support_df$SurveyType == "NatRep"),]

# Proactive survey
Corpus_pro_support <- Corpus(VectorSource(proact_support$Q15_WSP_support_text))
Corpus_pro_support <- tm_map(Corpus_pro_support, removeNumbers)
Corpus_pro_support <- tm_map(Corpus_pro_support, removeWords, stopwords("english")) #removes common english stopwords
Corpus_pro_support <- tm_map(Corpus_pro_support, removeWords, c("they", "the", "also"))
Corpus_pro_support <- tm_map(Corpus_pro_support, lemmatize_words)
wordcloud_pro_support <- wordcloud(Corpus_pro_support,  max.words=50, random.order=FALSE, rot.per=0,
          use.r.layout=FALSE, colors=brewer.pal(8, "Dark2"))
```



```
# Nat Rep survey
Corpus_nat_support <- Corpus(VectorSource(natrep_support$Q15_WSP_support_text))
Corpus_nat_support <- tm_map(Corpus_nat_support, removeNumbers)
Corpus_nat_support <- tm_map(Corpus_nat_support, removeWords, stopwords("english")) #removes co
mmon english stopwords
Corpus_nat_support <- tm_map(Corpus_nat_support, removeWords, c("they", "the", "also"))
Corpus_nat_support <- tm_map(Corpus_nat_support, lemmatize_words)
wordcloud_nat_support <- wordcloud(Corpus_nat_support,  max.words=50, random.order=FALSE, rot.p
er=0,
          use.r.layout=FALSE, colors=brewer.pal(8, "Dark2"))
```


support
reintroduction
animals
environment
idea
native
wildlife
many
beautiful
like
diversity
know
good
can
nice
help
birds
enough
bird
see
dont
storks
wild
species
need
the
think
why
great
they
part
nature
its
make
will
country
reintroduce
back
right
natural
lovely

Q16a. Expressing views on WS management

Question = Do you feel that you can express your views on the ongoing white stork reintroduction in a way that will influence management decisions?

```
# Polarity / Sentiment Analysis

### Q16. What are yours views on the management of White Storks?
head(final_data$Q16_views_management_open)
```

```
## [1]
## [2]
## [3] question seems unclear
## [4] I work in conservation and rewilding so am knowledgable and trained on the subject.
## [5] I've not looked into the project as much as I should, which is dreadful of me. To be hon
est I think that's because I live in the north, and we forever seem to be far behind in the pro
gress of bold reintroductions - even free-living beavers are barely discussed here, so I suppos
e it seems a long way off to have White Storks (though I hope not).
## [6]
## 1284 Levels:  ...
```

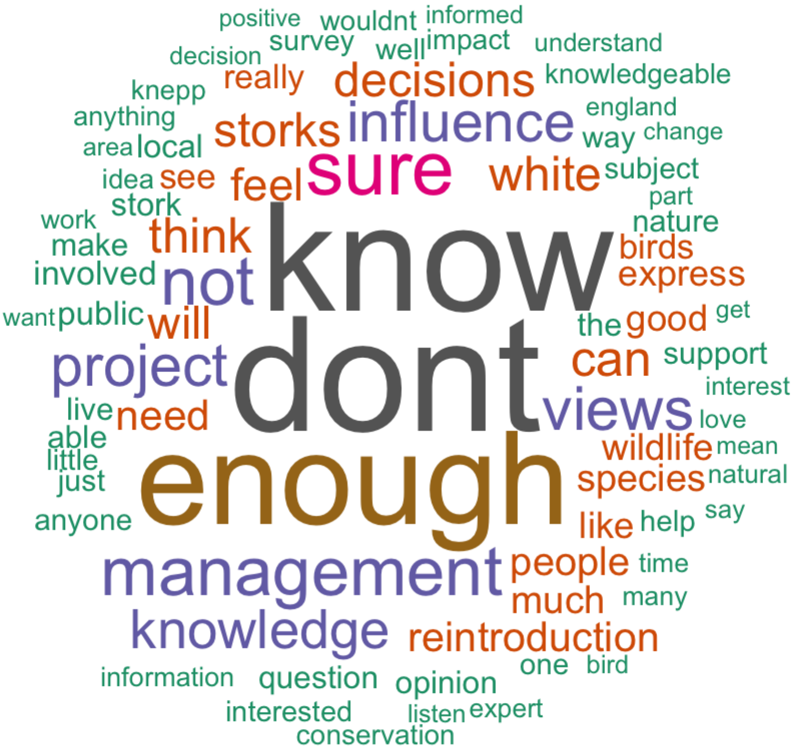
```
# Clean the data
final_data$Q16_views_management_text <- gsub("[^[:graph:]]", " ", final_data$Q16_views_manageme
nt_open)
final_data$Q16_views_management_text <- gsub("[[:punct:]]", "", final_data$Q16_views_management
_text)# Remove punctuation
final_data$Q16_views_management_text <- gsub("^ ", "", final_data$Q16_views_management_text)
final_data$Q16_views_management_text <- gsub(" $", "", final_data$Q16_views_management_text)

# Reasons for support/not support WSP
class(final_data$Q16_views_management_text)
```

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

```
sentiment(get_sentences(final_data$Q16_views_management_text))
```

```
##       element_id sentence_id word_count  sentiment
##    1:          1           1         NA  0.0000000
##    2:          2           1         NA  0.0000000
##    3:          3           1          3 -0.2886751
##    4:          4           1         14  0.2806243
##    5:          5           1         66  0.1107823
##   ---
## 3527:       3527           1         NA  0.0000000
## 3528:       3528           1         13  0.0000000
## 3529:       3529           1         NA  0.0000000
## 3530:       3530           1          3  0.0000000
## 3531:       3531           1          4  0.0000000
```

```
# Wrd frequencies
Corpus_management <- Corpus(VectorSource(final_data$Q16_views_management_text))
Corpus_management <- tm_map(Corpus_management, removeNumbers)
Corpus_management <- tm_map(Corpus_management, removeWords, stopwords("english")) #removes common english stopwords
Corpus_management <- tm_map(Corpus_management, removeWords, c("they", "the", "also"))  #You can specify words to remove

#build a term-document matrix
TDM_management = tm::TermDocumentMatrix(Corpus_management, control = list(minWordLength = 1))
m_management = as.matrix(TDM_management)
v_management = sort(rowSums(m_management), decreasing = TRUE)
d_management = data.frame(word = names(v_management),freq=v_management)

# Create a wordcloud
wordcloud(Corpus_management, scale=c(5,0.5), max.words=80, random.order=FALSE, rot.per=0,
          use.r.layout=FALSE, colors=brewer.pal(8, "Dark2"))
```



Q17. Support for methods of WSP management?

Question = Q17.1-17.13. Which (if any) methods of white stork project management would you support?

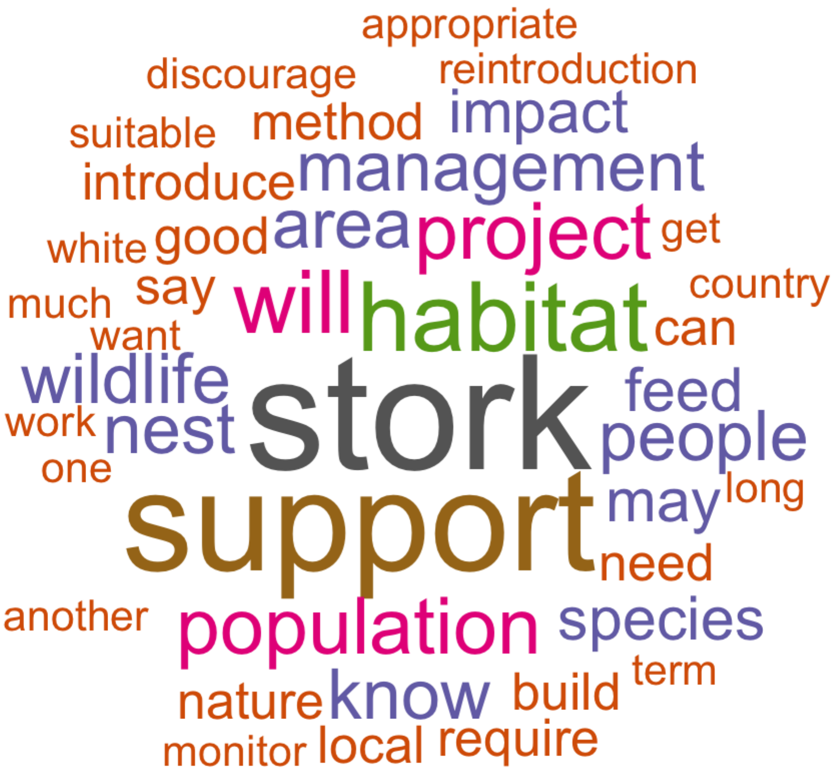
```
# Polarity / Sentiment Analysis

### Q17. Support for methods of WSP management?
# Clean the data
final_data$Q17.13a_other_open <- gsub("[^[:graph:]]", " ", final_data$Q17.13a_other_open)
final_data$Q17.13a_other_open <- gsub("[[:punct:]]", "", final_data$Q17.13a_other_open)# Remove punctuation
final_data$Q17.13a_other_open <- gsub("^ ", "", final_data$Q17.13a_other_open)
final_data$Q17.13a_other_open <- gsub(" $", "", final_data$Q17.13a_other_open)

# Reasons for support/not support WSP
class(final_data$Q17.13a_other_open)
```

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

```
sentiment(get_sentences(final_data$Q17.13a_other_open))
```

```
##       element_id sentence_id word_count sentiment
##    1:          1           1         NA         0
##    2:          2           1         NA         0
##    3:          3           1         NA         0
##    4:          4           1         NA         0
##    5:          5           1         NA         0
##   ---
## 3527:       3527           1         NA         0
## 3528:       3528           1         NA         0
## 3529:       3529           1         NA         0
## 3530:       3530           1         NA         0
## 3531:       3531           1         NA         0
```

```
# Word frequencies
Corpus_methods <- Corpus(VectorSource(final_data$Q17.13a_other_open))
Corpus_methods <- tm_map(Corpus_methods, removeNumbers)
Corpus_methods <- tm_map(Corpus_methods, lemmatize_strings)
Corpus_methods <- tm_map(Corpus_methods, removeWords, stopwords("english")) #removes common english stopwords
Corpus_methods <- tm_map(Corpus_methods, removeWords, c("they", "the", "also"))  #You can specify words to remove

#build a term-document matrix
TDM_methods = tm::TermDocumentMatrix(Corpus_methods, control = list(minWordLength = 1))
m_methods = as.matrix(TDM_methods)
v_methods = sort(rowSums(m_methods), decreasing = TRUE)
d_methods = data.frame(word = names(v_methods),freq=v_methods)

# Create a wordcloud
wordcloud(Corpus_methods, scale=c(5,0.5), max.words=80, random.order=FALSE, rot.per=0,
          use.r.layout=FALSE, colors=brewer.pal(8, "Dark2"))
```



```
# We can find the words that appear at least 100 times by calling the findFreqTerms() function
 on the term.doc.matrix
HiFreq_methods <- findFreqTerms(TDM_methods, 100)
HiFreq_methods
```

```
## character(0)
```

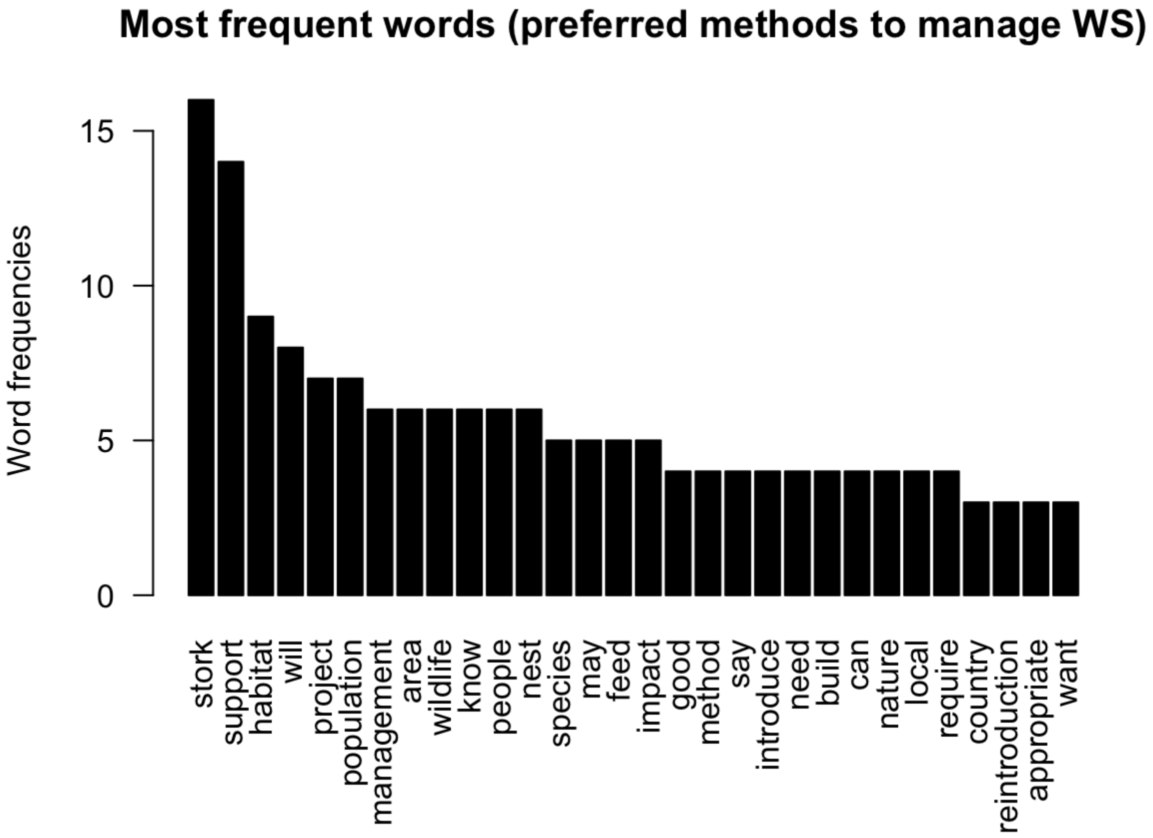
```
# Now you also see how associated a word is to another word or a list of words.
findAssocs(TDM_methods, HiFreq_methods, 0.4)
```

```
## named list()
```

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

```
## $long
##       dubious      maintain   reintroduce       release         value
##          0.89          0.89          0.89          0.89          0.89
##    population          term          feed         order supplementary
##          0.82          0.80          0.67          0.63          0.63
```

```
# Increase margin size
par(mar=c(8,6,4,4))
barplot(d_methods[1:30,]$freq, las = 2, names.arg = d_methods[1:30,]$word,
        col ="black", main ="Most frequent words (preferred methods to manage WS)",
        ylab = "Word frequencies")
```



[1] University of Brighton, l.jones4@brighton.ac.uk (mailto:l.jones4@brighton.ac.uk)↩