Processing twitter text

ANALYZING SOCIAL MEDIA DATA IN R

Vivek Vijayaraghavan
Data Science Coach

Lesson overview

- Why process tweet text?
- Steps in processing tweet text
    - removing redundant information
    - Converting text into a corpus
    - Removing stop words

Why process tweet text?

- Tweet text is unstructured, noisy, and raw
- Contains emoticons, URLs, numbers
- Clean text required for analysis and reliable results

Steps in text processing

Remove redundant information

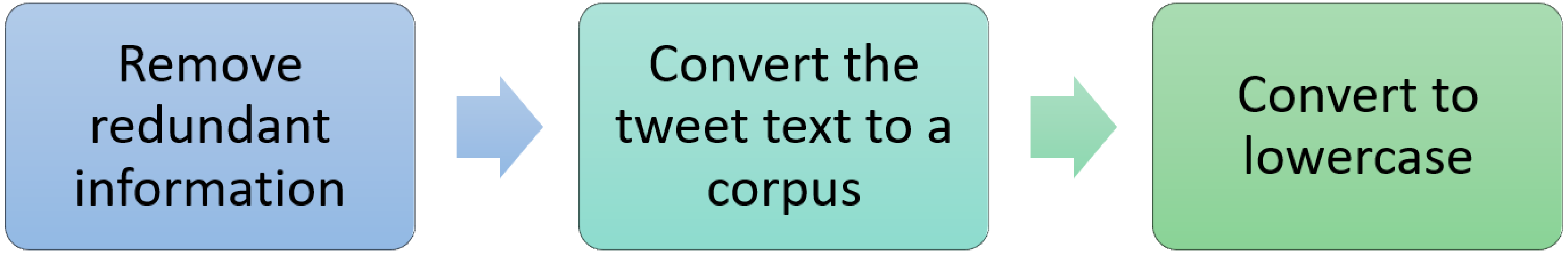
- URLs
- special characters
- punctuations
- numbers

Steps in text processing

Remove redundant information → Convert the tweet text to a corpus

- URLs
- special characters
- punctuations
- numbers

Steps in text processing



- URLs
- special characters
- punctuations
- numbers

Steps in text processing

Remove redundant information → Convert the tweet text to a corpus → Convert to lowercase → Remove common words or stop words

- URLs
- special characters
- punctuations
- numbers

Extract tweet text

```r
# Extract 1000 tweets on "Obesity" in English and exclude retweets
tweets_df <- search_tweets("Obesity", n = 1000, include_rts = F, lang = 'en')
```

```r
# Extract the tweet texts and save it in a data frame
twt_txt <- tweets_df$text
```

Extract tweet text

```
head(twt_txt, 3)
```

```
[1] "@WeeaUwU for real, obesity should not be praised like it is in today's society"

[2] "Great work by @DosingMatters in @AJHPOfficial on \"Vancomycin Vd estimation in
adults with class III obesity\". As we continue to study/learn more about dosing in
large body weight pts, we see that it's not a simple, one size, one level estimate
that works https://t.co/KkYPqS6JzG"

[3] "The Scottish Government have an ambition to halve childhood obesity by 2030.
This means reducing obesity prevalence in 2-15yo children in Scotland to 7%.
\n\n\U0001f449 In 2018, this figure was 16%\n\nFind out more in our latest blog:
https://t.co/FWp56QWjQc https://t.co/XBK8Je7F1A"
```

Removing URLs

```
# Remove URLs from the tweet text
library(qdapRegex)
twt_txt_url <- rm_twitter_url(twt_txt)
```

Removing URLs

```
twt_txt_url[1:3]
```

```
[1] "@WeeaUwU for real, obesity should not be praised like it is in today's society"

[2] "Great work by @DosingMatters in @AJHPOfficial on \"Vancomycin Vd estimation in  adu
with class III obesity\". As we continue to study/learn more about dosing in large body
weight pts, we see that it's not a simple, one size, one level estimate that works"

[3] "The Scottish Government have an ambition to halve childhood obesity by 2030.
This means reducing obesity prevalence in 2-15yo children in Scotland to 7%.
\U0001f449In 2018, this figure was 16% Find out more in our latest blog:"
```

Special characters, punctuation & numbers

```
# Remove special characters, punctuation & numbers
twt_txt_chrs  <- gsub("[^A-Za-z]", " ", twt_txt_url)
```

Special characters, punctuation & numbers

```
twt_txt_chrs[1:3]
```

```
[1] " WeeaUwU for real  obesity should not be praised like it is in today s society"

[2] "Great work by  DosingMatters in  AJHPOfficial on  Vancomycin Vd estimation in
adults with class III obesity   As we continue to study learn more about dosing in
large body weight pts  we see that it s not a simple  one size  one level estimate
that works"

[3] "The Scottish Government have an ambition to halve childhood obesity by       This
means reducing obesity prevalence in     yo children in Scotland to       In       this
figure was     Find out more in our latest blog "
```

Convert to text corpus

```
# Convert to text corpus
library(tm)
twt_corpus <- twt_txt_chrs %>%
                VectorSource() %>%
                Corpus()
```

```
twt_corpus[[3]]$content
```

```
[1] "The Scottish Government have an ambition to halve childhood obesity by
This means reducing obesity prevalence in     yo children in Scotland to       In
     this figure was     Find out more in our latest blog "
```

Convert to lowercase

- A word should not be counted as two different words if the case is different

```
# Convert text corpus to lowercase
twt_corpus_lwr <- tm_map(twt_corpus, tolower)
twt_corpus_lwr[[3]]$content
```

```
[1] "the scottish government have an ambition to halve childhood obesity by     this
means reducing obesity prevalence in     yo children in scotland to       in    this
figure was     find out more in our latest blog "
```

What are stop words?

- Stop words are commonly used words like a, an, and but

```
# Common stop words in English
stopwords("english")
```

```
 [1] "i"          "me"         "my"         "myself"
 [8] "ourselves"  "you"        "your"       "yours"
[15] "him"        "his"        "himself"    "she"
[22] "it"         "its"        "itself"     "they"
[29] "themselves" "what"       "which"      "who"
[36] "these"      "those"      "am"         "is"
[43] "be"         "been"       "being"      "have"
[50] "do"         "does"       "did"        "doing"
[57] "ought"      "i'm"        "you're"     "he's"
```

Remove stop words

- Stop words need to be removed to focus on the important words

```
# Remove stop words from corpus
twt_corpus_stpwd <- tm_map(twt_corpus_lwr, removeWords, stopwords("english"))
```

```
twt_corpus_stpwd[[3]]$content
```

```
[1] " scottish government   ambition  halve childhood obesity         means
reducing obesity prevalence      yo children  scotland                figure
find     latest blog "
```

Remove additional spaces

- Remove additional spaces to create a clean corpus

```
# Remove additional spaces
twt_corpus_final <- tm_map(twt_corpus_stpwd, stripWhitespace)
```

```
twt_corpus_final[[3]]$content
```

```
[1] " scottish government ambition halve childhood obesity means reducing obesity
prevalence yo children scotland figure find latest blog "
```

Let's practice!

ANALYZING SOCIAL MEDIA DATA IN R

Visualize popular terms

ANALYZING SOCIAL MEDIA DATA IN R

Vivek Vijayaraghavan
Data Science Coach

Lesson Overview

- Extract most frequent terms from the text corpus
- Remove custom stop words and refine corpus
- Visualize popular terms using bar plot and word cloud

Term frequency

- Extract term frequency which is the number of occurrences of each word

```
# Extract term frequency
library(qdap)
term_count  <-  freq_terms(twt_corpus_final, 60)
term_count
```

Term frequency

#	WORD	FREQ
1	obesity	1026
2	s	313
3	health	129
4	t	129
5	rates	125
6	people	121
7	child	120
8	fat	104
9	ranks	98
10	california	97
11	can	95
12	diabetes	85
13	amp	79
14	weight	79
15	food	66

#	WORD	FREQ
16	healthy	61
17	childhooc	59
18	one	56
19	like	54
20	realcanda	53
21	meghanm	52
22	overweig	51
23	will	50
24	just	49
25	diet	48
26	obese	47
27	cancer	46
28	black	45
29	disease	43
30	many	42

#	WORD	FREQ
31	problem	42
32	body	41
33	new	41
34	time	39
35	don	38
36	also	37
37	know	37
38	us	36
39	life	35
40	trump	35
41	children	34
42	risk	34
43	need	33
44	think	32
45	dr	31

#	WORD	FREQ
46	get	31
47	m	31
48	may	31
49	now	31
50	heart	30
51	eat	29
52	help	29
53	sugar	29
54	world	29
55	epidemic	28
56	re	28
57	study	28
58	eating	27
59	day	26
60	much	26

Removing custom stop words

```
# Create a vector of custom stop words
custom_stop <- c("obesity", "can", "amp", "one", "like", "will", "just",
                 "many", "new", "know", "also", "need", "may", "now",
                 "get", "s", "t", "m", "re")
```

```
# Remove custom stop words
twt_corpus_refined <- tm_map(twt_corpus_final,removeWords, custom_stop)
```

Term count after refining corpus

```
# Term count after refining corpus
term_count_clean <- freq_terms(twt_corpus_refined, 20)
term_count_clean
```

Term frequency after refining corpus

	WORD	FREQ		WORD	FREQ
1	health	129	11	healthy	61
2	rates	125	12	childhood	59
3	people	121	13	realcandaceo	53
4	child	120	14	meghanmccain	52
5	fat	104	15	overweight	51
6	ranks	98	16	diet	48
7	california	97	17	obese	47
8	diabetes	85	18	cancer	46
9	weight	79	19	black	45
10	food	66	20	disease	43

- Brand promoting an obesity management program can analyze these terms

Bar plot of popular terms

- Create a bar plot of terms that occur more than 50 times
- Bar plots summarize popular terms in an easily interpretable form

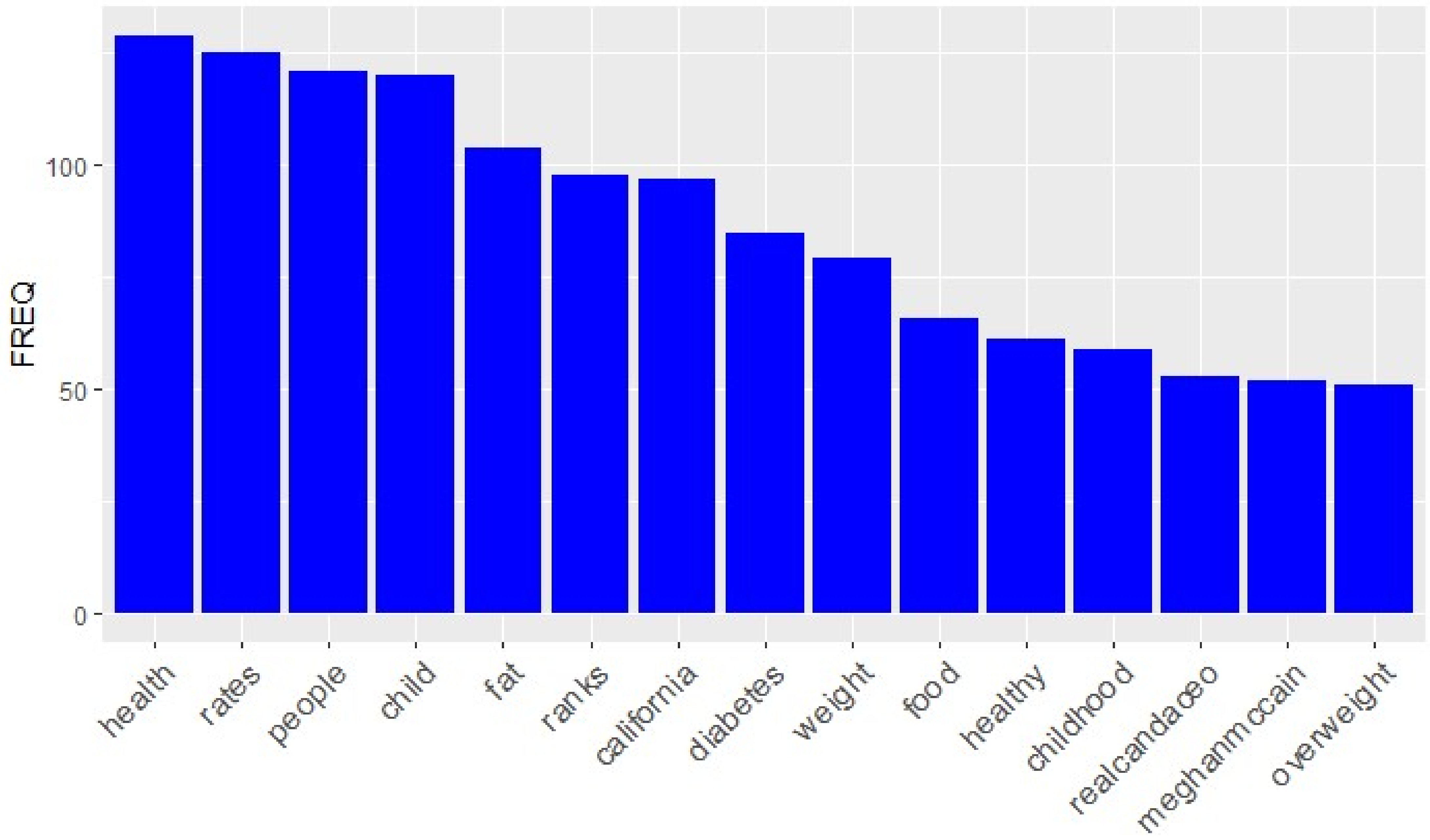
```
# Create a subset dataframe
term50 <- subset(term_count_clean, FREQ > 50)
```

Bar plot of most popular terms

```r
library(ggplot2)
```

```r
# Create a bar plot of frequent terms
ggplot(term50, aes(x = reorder(WORD,  -FREQ),  y = FREQ)) +
       geom_bar(stat = "identity", fill = "blue") +
       theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

Bar plot of popular terms



Word cloud

- Visualize the frequent terms using word clouds
- Word cloud is an image made up of words
- Size of each word indicates its frequency
- Effective promotional image for campaigns
- Communicates the brand messaging and highlights popular terms

Word cloud based on min frequency

- The `wordcloud()` function helps create word clouds

```
# Create a word cloud based on min frequency
library(wordcloud)
wordcloud(twt_corpus_refined, min.freq = 20, colors = "red",
          scale = c(3,0.5), random.order = FALSE)
```

Word cloud based on min frequency

Colorful word cloud

```
# Create a colorful word cloud
library(RColorBrewer)
wordcloud(twt_corpus_refined, max.words = 100,
          colors = brewer.pal(6,"Dark2"), scale = c(2.5,.5),
          random.order = FALSE)
```

Colorful word cloud

Let's practice!

ANALYZING SOCIAL MEDIA DATA IN R

Topic modeling of tweets

ANALYZING SOCIAL MEDIA DATA IN R

Vivek Vijayaraghavan
Data Science Coach

Lesson Overview

- Fundamentals of topic modeling
- Create a document term matrix or DTM
- Build a topic model from the DTM

Topic and Document

TOPIC

- Collection of dominant keywords representative of the topic.
- **Example:** Keywords "travel", "vacation", "hotel" representative of the topic "tourism".

Topic and Document

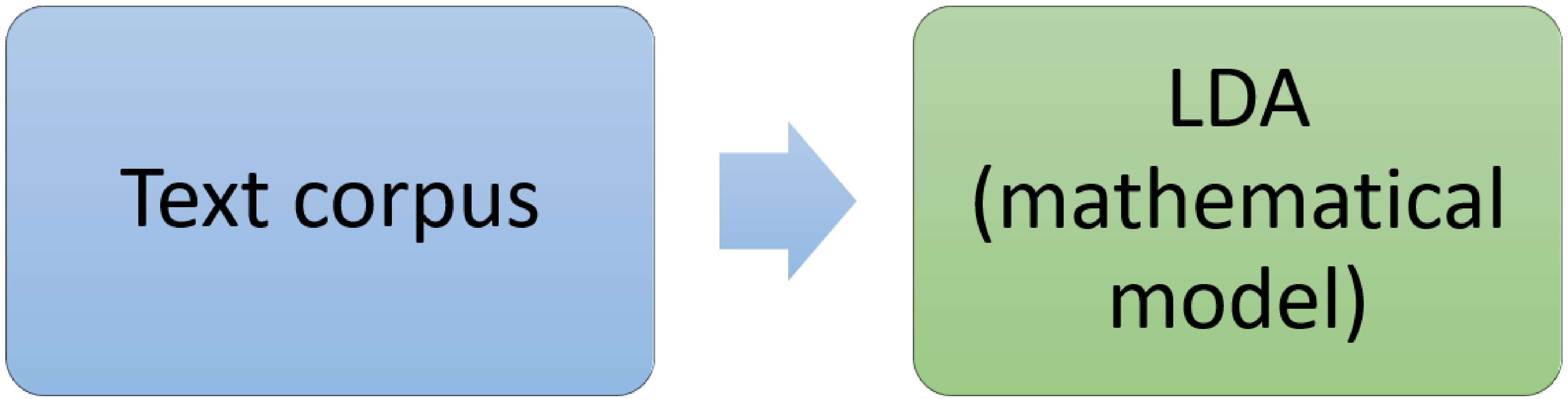
TOPIC	DOCUMENT
• Collection of dominant keywords representative of the topic. • **Example:** Keywords "travel", "vacation", "hotel" representative of the topic "tourism".	• Term used to describe one text record. • **Example:** A tweet on tourism is a document.

Topic modeling

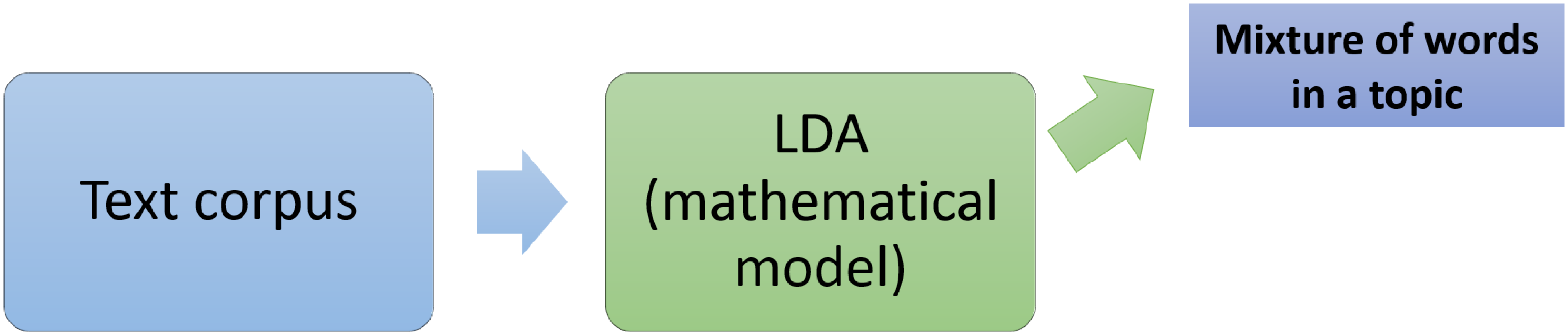
- Task of automatically discovering topics
- Extract core discussion topics from large datasets
- Quickly summarize vast information into topics

How LDA works

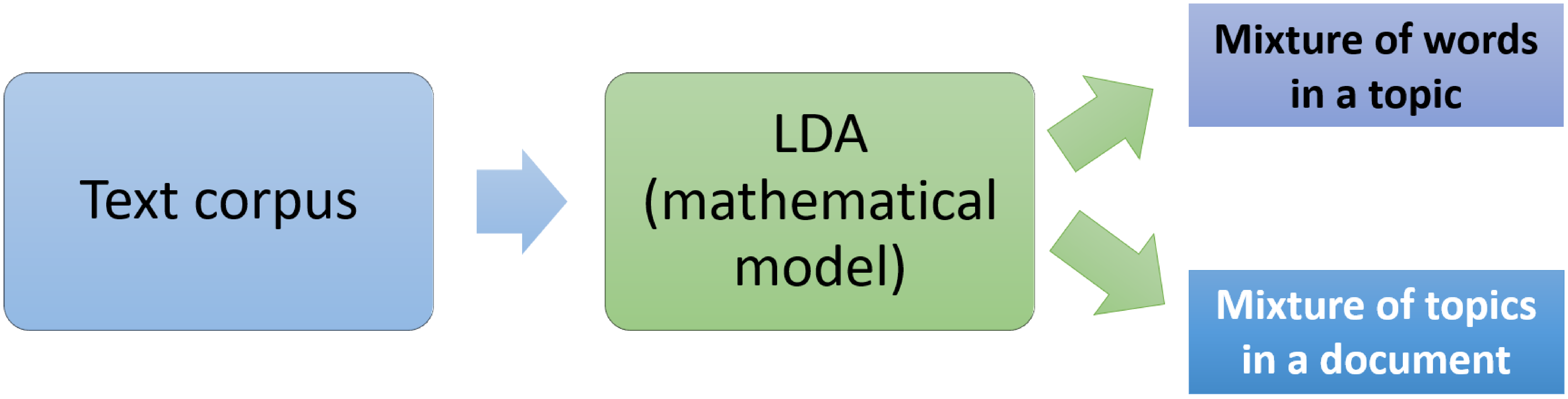
- Latent Dirichlet Allocation algorithm for topic modeling



How LDA works



How LDA works



Document term matrix (DTM)

- Create a document term matrix
- DTM is a matrix representation of a corpus
- Documents are rows and words or terms are columns

Terms

Documents

social media analysis → Document1

twitter data analysis → Document2

	social	media	analysis	twitter	data
Document1	1	1	1	0	0
Document2	0	0	1	1	1

Document Term Matrix (DTM)

Create a document term matrix

```
# Create a document term matrix
dtm <- DocumentTermMatrix(twt_corpus_refined)
```

Create a document term matrix

```
# Inspect the DTM
inspect(dtm)
```

Create a document term matrix

```
<<DocumentTermMatrix (documents: 1000, terms: 5079)>>
Non-/sparse entries: 12862/5066138
Sparsity           : 100%
Maximal term length: 29
Weighting          : term frequency (tf)
Sample             :
     Terms
Docs    california child diabetes fat food health people ranks rates weight
  131            0     0        0   0    0      0      0     0     0      0
  161            0     0        0   2    0      0      0     0     0      1
  295            0     0        0   0    1      0      1     0     0      0
  418            0     0        0   0    0      0      0     0     1      0
  604            0     0        1   0    0      1      0     0     0      0
```

Preparing the DTM

- Filter the DTM for rows that have a row sum greater than 0

```
# Find the sum of word counts in each Document
rowTotals <- apply(dtm , 1, sum)
```

```
# Select rows from DTM with row totals greater than zero
tweet_dtm_new <- dtm[rowTotals> 0, ]
```

Build the topic model

- Create the topic model using the `LDA()` function

```
# Build the topic model
library(topicmodels)
lda_5 <- LDA(tweet_dtm_new, k = 5)
```

Build the topic model

- Extracted 5 topics from the tweet corpus

```
# View top 10 terms in the topic model
top_10terms <- terms(lda_5,10)
top_10terms
```

View top 10 terms in the topic model

```
      Topic 1         Topic 2         Topic 3     Topic 4      Topic 5
 [1,] "disease"       "people"        "black"     "child"      "weight"
 [2,] "health"        "health"        "fat"       "rates"      "diet"
 [3,] "cancer"        "diabetes"      "trump"     "ranks"      "food"
 [4,] "meghanmccain" "overweight"    "childhood" "california" "diabetes"
 [5,] "realcandaceo" "fat"           "health"    "fat"        "health"
 [6,] "food"          "meghanmccain" "professor" "eat"        "bmi"
 [7,] "risk"          "realcandaceo" "gender"    "people"     "problem"
 [8,] "heart"         "body"          "studies"   "epidemic"   "eating"
 [9,] "weight"        "weight"        "healthy"   "health"     "disease"
[10,] "diabetes"      "obese"         "problem"   "healthy"    "family"
```

- An obesity management program can center its theme around a core topic

Let's practice!

ANALYZING SOCIAL MEDIA DATA IN R

Twitter sentiment analysis

ANALYZING SOCIAL MEDIA DATA IN R

Vivek Vijayaraghavan
Data Science Coach

Lesson Overview

- What is sentiment analysis?
- Perform sentiment analysis on tweets
- Interpret to understand people's feelings and opinions

Sentiment analysis

- Retrieve information on perception of a product or brand
- Extract and quantify positive, negative and neutral opinions
- Emotions like trust, joy, and anger from the text

Significance of sentiment analysis

- Customer perceptions influence purchasing decisions
- Helps understand the pulse of what customers feel
- Proactive approach to listen to the customer and engage directly

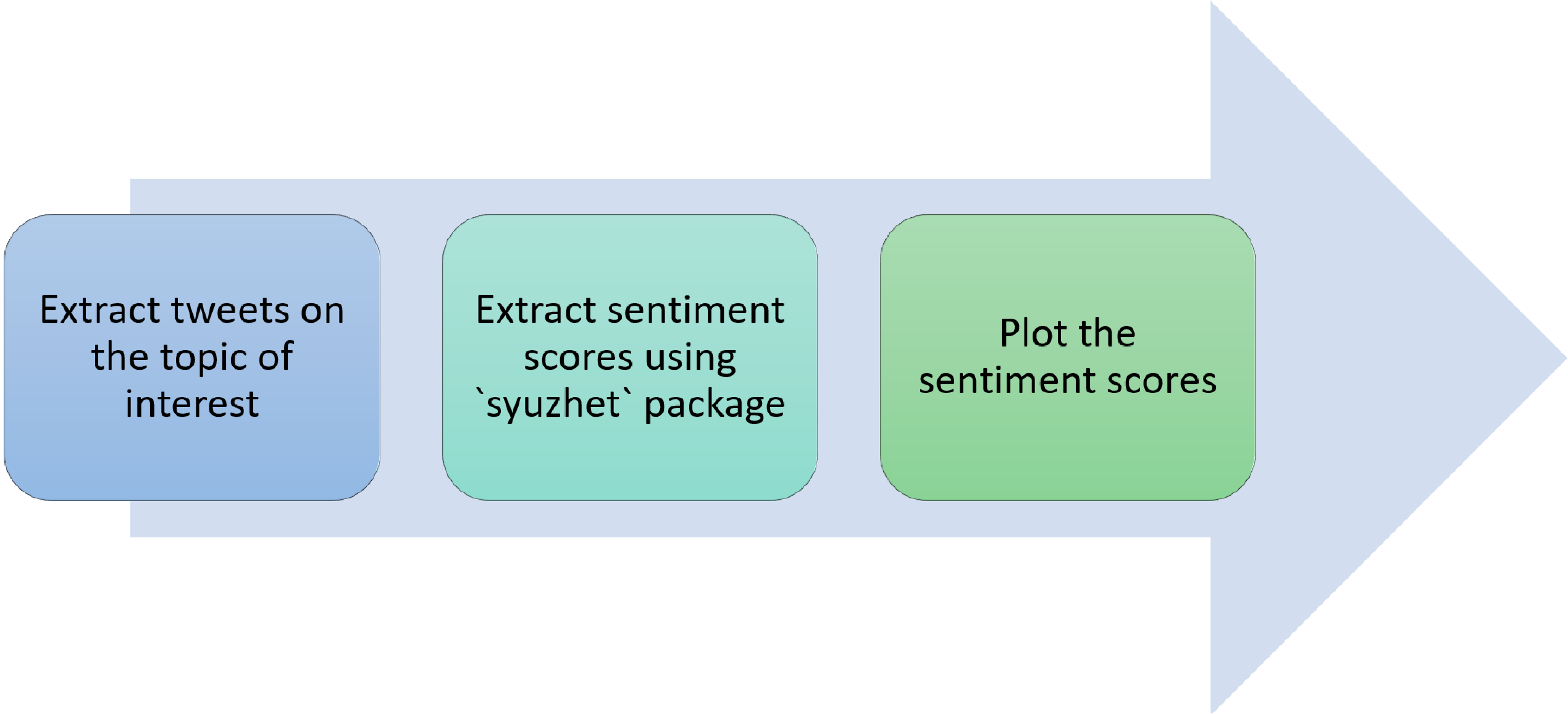
How sentiment analysis works

- Pre-defined sentiment libraries to calculate scores
- Trained and scored based on meaning or intent of words
- Each word is scored based on its nearness to a positive or negative word
- Same concept is extended to words expressing specific emotions

Sentiment analysis steps

Sentiment analysis steps

Sentiment analysis steps



Sentiment analysis steps

1. Extract tweets on the topic of interest
2. Extract sentiment scores using `syuzhet` package
3. Plot the sentiment scores
4. Visualize and interpret customer perception and emotions

Extract tweets for sentiment analysis

```
# Extract tweets on galaxy fold
twts_galxy  <-  search_tweets("galaxy fold", n = 5000,
                              lang = "en", include_rts = FALSE)
```

Perform sentiment analysis

```
# Perform sentiment analysis for tweets on galaxy fold
library(syuzhet)
sa.value <- get_nrc_sentiment(twts_galxy$text)
```

View sentiment scores

```
# View the sentiment scores
sa.value[1:5,1:7]
```

```
anger    anticipation    disgust    fear    joy    sadness    surprise
<dbl>       <dbl>         <dbl>    <dbl>  <dbl>    <dbl>       <dbl>
0             0             0        0      0        0           0
1             0             0        0      0        0           0
1             1             0        2      1        1           1
0             0             0        1      0        0           0
0             0             0        0      0        0           0
```

Sum of sentiment scores

```
# Calculate sum of sentiment scores
score <- colSums(sa.value[,])
```

Data frame of sentiment scores

```
# Convert to data frame
score_df <- data.frame(score)
```

```
# View the data frame
score_df
```

```
                score
                <dbl>
anger             211
anticipation      825
disgust           214
fear              253
joy               412
sadness           197
surprise          315
trust             641
negative          487
positive         1351
```

Data frame of sentiment scores

```
# Convert row names into 'sentiment' column
# Combine with sentiment scores
sa.score <- cbind(sentiment = row.names(score_df),
                  score_df, row.names=NULL)
```

Data frame of sentiment scores

```
# View data frame with sentiment scores
print(sa.score)
```

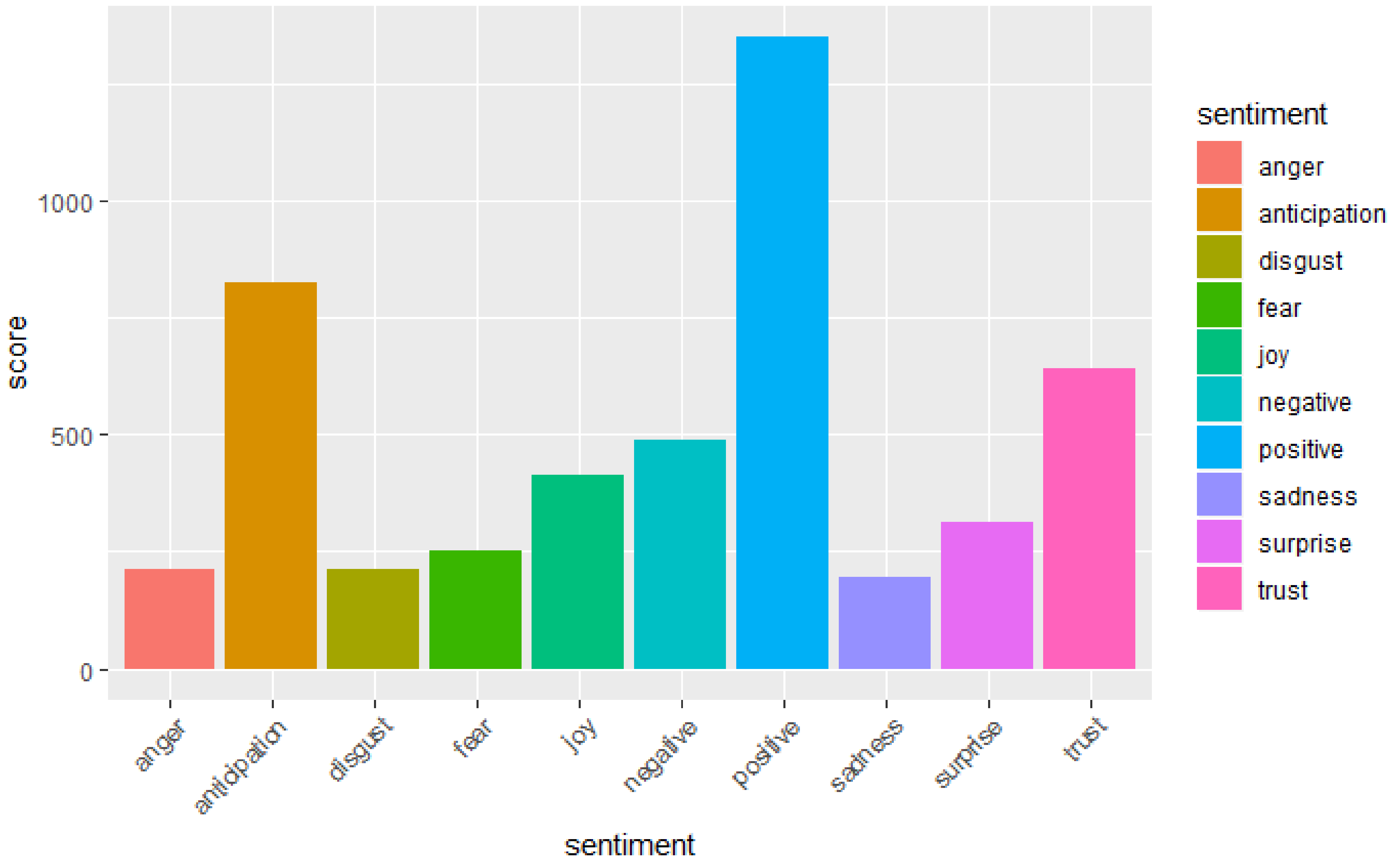
```
sentiment       score
<fctr>          <dbl>
anger            211
anticipation     825
disgust          214
fear             253
joy              412
sadness          197
surprise         315
trust            641
negative         487
positive         1351
```

Plot and visualize sentiments

- Plot and visualize sentiments using `ggplot()`

```
# Plot the sentiment scores
ggplot(data = sa.score2, aes(x = sentiment, y = score,
       fill = sentiment)) +
       geom_bar(stat = "identity") +
       theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

Visualize the sentiments



Let's practice!

ANALYZING SOCIAL MEDIA DATA IN R