Word Cloud

Hayden Ratliff, Pulkit Rampa, Qinyuan Jiang, Abigail Castro, Elly Zarzyski

We are going to be working with functions from `tidyverse` (e.g., `stringr`, `dplyr`) and `tidytext` to do some basic text mining and build a **word cloud of the first activity you would do after the pandemic is over**. A good introduction to the `tidytext` package is the free book Text Mining with R (by Silge and Robinson)

1- Load necessary packages.

You will need tidyverse, tidytext, and ggwordcloud

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.6     v dplyr   1.0.3
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(tidytext)
```

```
## Warning: package 'tidytext' was built under R version 4.0.4
```

```
library(ggwordcloud)
```

```
## Warning: package 'ggwordcloud' was built under R version 4.0.4
```

2- Load the dataset

from here:

```
cloud_dataframe <- read_csv("text_data.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Activity = col_character()
## )
```

3- Make the data tidy

A tidy text format is a table with one *token* per row. - A token can be a: word, n-gram, sentence, line, paragraph, tweet - This will let us use the power of our tidyverse functions.

3.1 *unnest* the text column so there is one *word* per row.

Hint: use unnest_tokens in tidytext. Convert all words to lowercase.

```
cloud_tokens <- cloud_dataframe %>% unnest_tokens(word, Activity, to_lower=TRUE,
                                                  strip_punct=TRUE)
```

4- Clean the text

4.1 - Check if any of the words should be considered together, e.g. "go to"

- manual exploration
  - go to concert

4.2 - Create a general name for words that belong to the same category, e.g. mother, parents belong to the family category

4.2 Remove uninteresting words incldluing:

- stop words such as "an", "and", "of", "the", etc. Hint: use stop_words in tidytext to get a list of stop words and then use a join function to remove them from your data (what join function is appropriate here?).
- punctuation if not done in unnest
  - punctuation was stripped in 3.1
- whitespace
  - already split on whitespace
- numbers and other non-text characters e.g. apostrophes if any

```
numbers <- cloud_tokens %>% filter(str_detect(word, "^[0-9]"))
cloud_no_numbers <- cloud_tokens %>% anti_join(numbers, by="word")
```

5. Count the words

```
cloud_no_numbers %>% count(word) %>% arrange(desc(n))
```

```
## # A tibble: 131 x 2
##    word        n
##    <chr>   <int>
##  1 to         36
##  2 go         35
##  3 a          31
##  4 travel     15
##  5 concert    12
##  6 in          9
##  7 i           8
##  8 out         8
##  9 with        8
## 10 friends     7
## # ... with 121 more rows
```

6. Create the word cloud

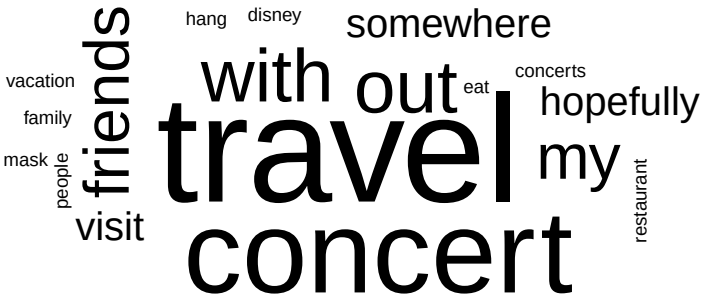
A word cloud is a graphical representation of text that sizes and colors the words. Size is usually considered to be proportional to the frequency of the word's occurrence, but in general could be related to some other measure of *importance*.

The R package `ggwordcloud` implements a wordcloud geom for use with `ggplot2`. The package has a helpful webpage with examples: `ggwordcloud` R package help

```
cloud_filtered <- cloud_no_numbers %>% filter(!word %in% c("to", "go", "a", "in", "i", "the", "on", "or
cloud_nums <- cloud_filtered %>% count(word) %>% arrange(desc(n))
```

```
cloud_nums %>% with(ggwordcloud(word, n, random.order=FALSE))
```

```
## Warning in png(filename = tmp_file, width = gw_pix, height = gh_pix, res =
## dev_dpi, : 'width=12, height=12' are unlikely values in pixels
```



7. According to your word cloud what are the most popular activities you would like to do after the pandemics?

- TRAVEL AND CONCERTS!!!!!!!!!!!!!!!!