Parse all tweets

Start by loading the tidyverse package, which contains lots of useful functions for data processing.

```
library(tidyverse)
```

Because the files are all pretty short, it makes the most sense to load them all first and then remove duplicates and split out new columns on the whole dataset.

To do this, we'll use the function `readxl::read_excel` to read each individual file. The call to `readxl::read_excel` will be nested inside of a call to `purrr::map_dfr`, which will loop over all files in the directory, running the read function on each and then compile the results in a single dataset by row-binding.

The first step is to grab the names of files you want to load

```
all_tweets_raw <- list.files('Virus Sheets', pattern = '*.xlsx', full.names = TRUE)
```

Tweets are currently stored in 21 files.

Next, read all the files and smush them into a single dataset

```
all_tweets <- map_dfr(all_tweets_raw, readxl::read_excel)
```

At this point, there are 16,492 tweets in the dataset, of which at least 12,617 are unique, based only on values in the first column (username, @, and date).

The next step is to remove true duplicates (in case some users posted multiple tweets on a given day).

```
all_tweets_nodupes <- all_tweets %>%
  distinct(Field1, Field2)
```

The final count on tweets after duplicates are removed is 12,617

Finally, it's time to break the tweets into the parts we care about. `Field1` can be broken into three components that are pretty easy to identify with regular expressions. `Field2` pretty much only contains text, but it might be useful to identify hashtags and standardize all the weird types of whitespace characters too.

```
all_tweets_f1parsed <- all_tweets_nodupes %>%
  # display name is all the characters that come before the @ sign
  # date is a capital letter followed by lower case letters, a space, numbers, and the end of the strin
  # the user name (@) can be extracted by extracting all the text between those two parts
  mutate(display_name = str_extract(Field1, ".*(?=@)"),
         date         = str_extract(Field1, "[A-Z][a-z]+ [0-9]{1,2}$"),
         user_name    = str_sub(Field1,
                                start = str_length(display_name) + 1,
                                end   = -str_length(date) - 2)) %>%
  select(display_name, user_name, date, Field2)


all_tweets_parsed <- all_tweets_f1parsed %>%
  # hashtags are ~easy to spot with regex, basically a set of alphanumeric characters preceeded by a #
  # str_extract_all returns a list of character vectors with all matches for each row of data,
  # which then need to converted into a single character vector d
  mutate(hashtags = str_extract_all(Field2, "#[:alnum:]+") %>%
```

```
          map_chr(paste, collapse = " "),
        text_cleaned_spaces = str_replace_all(Field2, "[:space:]|[:blank:]", " ")) %>%
  rename(raw_text = Field2) %>%
  select(everything(), hashtags)
```

Finally, save the results to disk as an Rds file (for reuse within R) and as a tab-separated spreadsheet (excluding raw text where tabs are still present)

```
all_tweets_parsed %>%
  write_rds("parsed_tweets.rds", compress = "gz") %>%
  select(-raw_text) %>%
  write_tsv("parsed_tweets.tsv")
```