Pulling from APIs - covid keywords

Isabel O'Malley

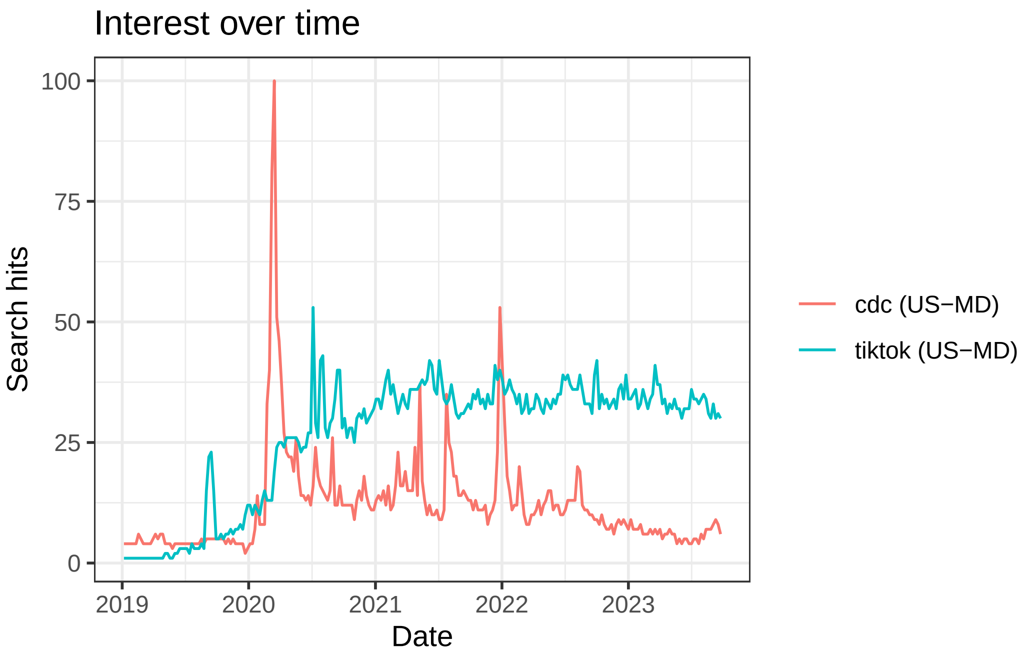
Link to github repo: https://github.com/isabelshaheen/JPSM727-assignment2.git

```
library(tidyverse)
library(gtrendsR)
library(censusapi)
```

Pulling from APIs - Covid Keywords

Our first data source is the Google Trends API. Suppose we are interested in the search trends for `CDC` and `Tiktok` in Maryland in the years 2019-2023. We could find this using the following code:

```
res <- gtrends(c("cdc", "tiktok"),
               geo = "US-MD",
               time = "2019-01-01 2023-9-30",
               low_search_volume = TRUE)
plot(res)
```



Answer the following questions for the keywords.

- Find the mean, median and variance of the search hits for the keywords.

First, we transform the `data.frame` into a `tibble`.

```
res_time <- as_tibble(res$interest_over_time)
glimpse(res_time)
```

```
Rows: 494
Columns: 7
$ date     <dttm> 2019-01-06, 2019-01-13, 2019-01-20, 2019-01-27, 2019-02-03, ~
$ hits     <int> 4, 4, 4, 4, 4, 4, 6, 5, 4, 4, 4, 4, 5, 6, 5, 6, 6, 4, 4, 4, 3~
$ keyword  <chr> "cdc", "cdc", "cdc", "cdc", "cdc", "cdc", "cdc", "cdc", "cdc"~
$ geo      <chr> "US-MD", "US-MD", "US-MD", "US-MD", "US-MD", "US-MD", "US-MD"~
$ time     <chr> "2019-01-01 2023-9-30", "2019-01-01 2023-9-30", "2019-01-01 2~
$ gprop    <chr> "web", "web", "web", "web", "web", "web", "web", "web", "web"~
$ category <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
```

Then, we use the group_by function and we find mean, SD, median, and max hits for the two keywords.

```
res_time %>%
  group_by(keyword) %>%
  summarize(mean_hits = mean(hits),
            sd_hits = sd(hits),
            median_hits = median(hits),
            max_hits = max(hits))
```

```
# A tibble: 2 x 5
  keyword mean_hits sd_hits median_hits max_hits
  <chr>       <dbl>   <dbl>       <int>    <int>
1 cdc          12.0    10.8          10      100
2 tiktok       26.4    12.9          32       53
```

```
group_by(res_time, keyword)
```

```
# A tibble: 494 x 7
# Groups:   keyword [2]
   date                 hits keyword geo   time                 gprop category
   <dttm>              <int> <chr>   <chr> <chr>                <chr>    <int>
 1 2019-01-06 00:00:00     4 cdc     US-MD 2019-01-01 2023-9-30 web          0
 2 2019-01-13 00:00:00     4 cdc     US-MD 2019-01-01 2023-9-30 web          0
 3 2019-01-20 00:00:00     4 cdc     US-MD 2019-01-01 2023-9-30 web          0
 4 2019-01-27 00:00:00     4 cdc     US-MD 2019-01-01 2023-9-30 web          0
 5 2019-02-03 00:00:00     4 cdc     US-MD 2019-01-01 2023-9-30 web          0
 6 2019-02-10 00:00:00     4 cdc     US-MD 2019-01-01 2023-9-30 web          0
 7 2019-02-17 00:00:00     6 cdc     US-MD 2019-01-01 2023-9-30 web          0
 8 2019-02-24 00:00:00     5 cdc     US-MD 2019-01-01 2023-9-30 web          0
 9 2019-03-03 00:00:00     4 cdc     US-MD 2019-01-01 2023-9-30 web          0
10 2019-03-10 00:00:00     4 cdc     US-MD 2019-01-01 2023-9-30 web          0
# i 484 more rows
```

- **Which cities (locations) have the highest search frequency for each keyword?** Note that there might be multiple rows for each city if there were hits for both keywords in that city. It might be easier to answer this question if we had the search hits info for both keywords in two separate variables. That is, each row would represent a unique city.

Pivot wider res_time to split the hits column into two variables

Make res$interest_by_city into a tibble and shorten name to res_city

Pivot wider with res_city

```
#identify duplicates
duplicates <- res_city %>%
  dplyr::group_by(location, geo, gprop, keyword) %>%
  dplyr::summarise(n = dplyr::n(), .groups = "drop") %>%
  dplyr::filter(n > 1L)

#remove duplicates
unique_res_city <- res_city %>%
  anti_join(duplicates, by = c("location", "geo", "gprop", "keyword"))

#pivot wider
res_city_w <- pivot_wider(unique_res_city,
                         names_from = keyword,
                         values_from = hits)
res_city_w
```

```
# A tibble: 278 x 5
   location           geo   gprop   cdc tiktok
   <chr>              <chr> <chr> <int>  <int>
 1 Indian Head        US-MD web     100     NA
 2 Derwood            US-MD web      91     NA
 3 Chevy Chase        US-MD web      57     64
 4 Fort Meade         US-MD web      55     NA
 5 Bethesda           US-MD web      51     NA
 6 Oakland            US-MD web      50     NA
 7 Joint Base Andrews US-MD web      49     NA
 8 Highland           US-MD web      47     NA
 9 Cabin John         US-MD web      47     NA
10 Braddock Heights   US-MD web      46     77
# i 268 more rows
```

Let's find the cities with the highest numbers of hits for our keywords using `dplyrs` `arrange()` function.

```
res_city_w %>%
  select(location, cdc) %>%
  arrange(desc(cdc))
```

```
# A tibble: 278 x 2
   location             cdc
   <chr>                 <int>
 1 Indian Head             100
 2 Derwood                  91
 3 Chevy Chase              57
 4 Fort Meade               55
 5 Bethesda                 51
 6 Oakland                  50
 7 Joint Base Andrews       49
 8 Highland                 47
 9 Cabin John               47
10 Braddock Heights         46
# i 268 more rows
```

```
res_city_w %>%
  select(location, tiktok) %>%
  arrange(desc(tiktok))
```

```
# A tibble: 278 x 2
   location                    tiktok
   <chr>                        <int>
 1 Woodstock                      100
 2 Greensboro                      97
 3 North Laurel                    92
 4 Manchester                      91
 5 Burtonsville                    90
 6 Calvert Beach-Long Beach        89
 7 East Riverdale                  89
 8 White Oak                       88
 9 Great Mills                     88
10 Bennsville                      87
# i 268 more rows
```

- Is there a relationship between the search intensities between the two keywords we used?

Convert NAs to 0

Find the correlation between the two keywords

```
cor_test_result <- cor.test(res_city_w$cdc, res_city_w$tiktok)

cor_test_result
```

```
    Pearson's product-moment correlation

data:  res_city_w$cdc and res_city_w$tiktok
t = -1.9879, df = 276, p-value = 0.04781
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.233196014 -0.001186076
sample estimates:
       cor
-0.1188125
```

Answer: The p-value is .4224 indicating there is no significant correlation between the number of google searches for “CDC” and the number of searches for “tiktok” in Maryland from 2019-2023.