Demographic estimates from the Facebook Marketing API

Connor Gilroy

2017-10-28

```r
library(httr)
library(jsonlite)
library(stringr)
library(tidyverse)
library(yaml)
```

We need two pieces of information from the config file: an access token, and an ads account. Make sure the ads account number is prefixed with "act_".

```
fb_cfg <- yaml.load_file("facebook_config.yml")
```

The url for accessing 'reach' estimates is based off of the ads account, so we construct it by pasting strings onto the base url

```
fb_url <- "https://graph.facebook.com/v2.10"
fb_ads_url <- str_c(fb_url, fb_cfg$ad_account_id,
                    "reachestimate", sep = "/")
```

```
targeting_spec <- '{"geo_locations": {"countries": ["US"]}}'
```

This is as simple as a targeting spec can be. We're specifying just a geographic location: one country, the United States.

The braces ({ }) and brackets ([ ]) define a hierarchy. As requests get more complex, it will become clear why we need them.

In addition to the access token and targeting spec, there are two required parameters we need to include. Their values aren't important.

```r
fb_query <- list(
  access_token = fb_cfg$access_token,
  currency = "USD",
  optimize_for = "NONE",
  targeting_spec = targeting_spec
)

r <- GET(fb_ads_url, query = fb_query)
```

Look at response

```r
prettify(content(r, as = "text"))

## {
##     "data": {
##         "users": 233000000,
##         "estimate_ready": true
##     }
## }
##

content(r, as = "parsed")$data$users

## [1] 233000000
```

In the state of Washington, how many men and women between the ages of 20 and 30 are on Facebook?

To answer this, we'll use two targeting specs from JSON files.

```
ts1 <- read_file("targeting_specs/targeting_spec_01.json")
ts2 <- read_file("targeting_specs/targeting_spec_02.json")
```

```
cat(ts1)

## {
##     "geo_locations": {
##         "regions": [{
##             "key": "3890",
##             "name": "Washington"
##         }]
##     },
##     "genders": [1],
##     "age_min": 20,
##     "age_max": 30
## }
```

Each request returns a single number. To build an interesting data set, we need to make many requests.

To do this, we'll wrap our request code in a *helper function*. The function will take a targeting spec and return a response. The other query parameters that don't change will be hard-coded.

Finally, we need to know about the concept of **rate limiting**. If we make too many requests too quickly, we'll be *rate limited*, and we won't be able to make any more requests for a while.

```r
make_fb_ads_request <- function(ts) {
  # Avoid rate limiting!
  # Don't make too many requests in a short period of time.
  Sys.sleep(5)

  fb_query <- list(
    access_token = fb_cfg$access_token,
    currency = "USD",
    optimize_for = "NONE",
    targeting_spec = minify(ts)
  )

  GET(fb_ads_url, query = fb_query)
}
```

```r
r1 <- make_fb_ads_request(ts1)
r2 <- make_fb_ads_request(ts2)
```

We need to combine the request information from the targeting spec with the estimated number of users from the response.

As with the requests, we'll do this using a function.

This function will return a data frame with a single row.

```r
process_fb_response <- function(ts, r) {
  ts_df <-
    as_data_frame(fromJSON(ts)) %>%
    unnest(geo_locations) %>%
    summarise_all(function(x) {
      ifelse(length(unique(x)) > 1,
             list(unique(x)), unique(x))
    })

  r_df <-
    as_data_frame(content(r, as = "parsed")$data) %>%
    select(users)

  bind_cols(ts_df, r_df)
}
```

```
bind_rows(
  process_fb_response(ts1, r1),
  process_fb_response(ts2, r2)
)
```

genders	age_min	age_max	key	name	users
1	20	30	3890	Washington	830000
2	20	30	3890	Washington	760000

Example 3: A more complex query

```
ts3 <- read_file("targeting_specs/targeting_spec_03.json")
cat(ts3)

## {
##     "geo_locations": {
##         "countries": ["US", "GB"]
##     },
##     "genders": [2],
##     "age_min": 25,
##     "age_max": 55,
##     "relationship_statuses": [2, 3],
##     "education_statuses": [2, 3]
## }
```

How many women in the US *or* Great Britain, between ages 25-55, are

- either in a relationship *or* married

AND

- either an undergrad *or* an alum

```
https://developers.facebook.com/docs/marketing-api/
targeting-specs
```

```
r3 <- make_fb_ads_request(ts3)
prettify(content(r3, as = "text"))

## {
##     "data": {
##         "users": 20000000,
##         "estimate_ready": true
##     }
## }
##
```

Try out `process_fb_response(ts3, r3)` too!

Exercises

Exercise 1: Pick 3 other countries and report the number of Facebook users in each. Compare these numbers to the actual populations from the World Bank or some other source.

Use two-digit country codes for countries:

https://en.wikipedia.org/wiki/ISO_3166-1_alpha-2

Exercise 2: Pick another US state and age range, and compare the numbers of men and women. You can look up the key that Facebook uses for each state in the file provided, targeting_spec_us_states.json.

You will need to create your own json targeting spec, using the examples provided as a template. It is recommended that you edit json files in RStudio, unless you are using a text editor designed for writing code.

Challenge exercise: Get the Facebook user population for each US state. Compare these estimates to population estimates from the Census ACS. Note that you'll need to make a separate call to the API for each state, which will take several minutes.