Final Project Narrative

Audrey Glaser

12/9/2019

Brief substantive background / goal

Google BigQuery is a cloud data warehouse that enables fast SQL queries using the processing power of Google's infrastructure. It is designed for commercial use, but it is also where Google hosts a number of public datasets that may be of interest to political scientists. One of these is a dataset containing comprehensive data on Google's political advertising clients across the world. This dataset includes ad-level and advertiser-level spending statistics, and geographic and demographic targeting data, for all political campaigns on the platform since the creation of the transparency report in the 2017-2018 U.S. midterms cycle.

The objective of this tutorial is to access this public dataset housed in Google BigQuery by making calls to its API, using the R package 'bigrquery'. The tutorial consists of two parts: setting up API access, and writing and making calls to the API.

Setting up BigQuery API access

There are multiple ways to enable access to the BigQuery API, but the first section of the tutorial will cover the "service account key" method, which is recommended in the BigQuery documentation.

Here is an overview of the steps covered in the tutorial:

1. Accessing the BigQuery web UI
2. Creating a service account key in the web UI
3. Installing 'bigrquery'
4. Connecting 'bigrquery' to the BigQuery API

Note that this process is more complicated than simply setting up an OAuth connection to an existing user account, as we did with 'rtweets' and Twitter. The BigQuery API set-up also requires cross-referencing between several pieces of Google documentation and the 'bigrquery' CRAN guide. My tutorial aims to clearly and efficiently guide newcomers through this multi-step process.

Using 'bigrquery'

The second section of my tutorial begins by explaining the three different "layers" of the bigrquery package: the low-level layer, the DBI interface, and the dplyr interface. The goal of this introduction to 'bigrquery' is to familiarize users with the different types of functions they will be working with. This part of the tutorial also sheds light on how "plug n' play" packages like 'bigrquery' are typically structured in relation to a RESTful API.

Next, the tutorial introduces the dbConnect() function, which is used to establish a live connection to Google's public dataset of political advertising data, The tutorial explains the function's component arguments, and its role in enabling specific calls to the API.

Finally, we use 'bigrquery' dbplyr functions to craft a specific call to the API. The goal of the request is to import a dataframe of U.S. political advertisers on Google who have spent at least $1,000,000 on the platform. This task is used as an opportunity to explore how 'bigrquery' transforms dbplyr code into SQL, and better understand what it means for dbplyr to "lazily" execute queries by default.

Future work

My initial plan was to access Facebook's Political Ads API as well, but Facebook requires users to first verify identity and country of residence. As part of the verification process, Facebook sends you a physical letter in the mail with an authorization code. I did not get the authorization letter in time to include Facebook Ads API in my project, but I hope to apply what I've learned using BigQuery to access the Facebook Ads API in the future!