Analysing Data with Python and BigQuery

Tom Clark
Runaway Play

What is BigQuery?

- Cloud based tool offered by Google for quickly querying large data sets
- Queryable using SQL
- Accessible through a web console, CLI tools, and a remote API
- Libraries available for many languages, including Python
- Integrates with other tools, particularly Google ones, for data input, analysis, and display of results

The plan: work through an example problem

- Create a project
- Create a dataset and table
- Prepare and upload our data
- Run queries
- Tidy up

Example problem: Twitter ratios*

$$\frac{\text{1 reply}}{\text{1 retweet}} = 1$$

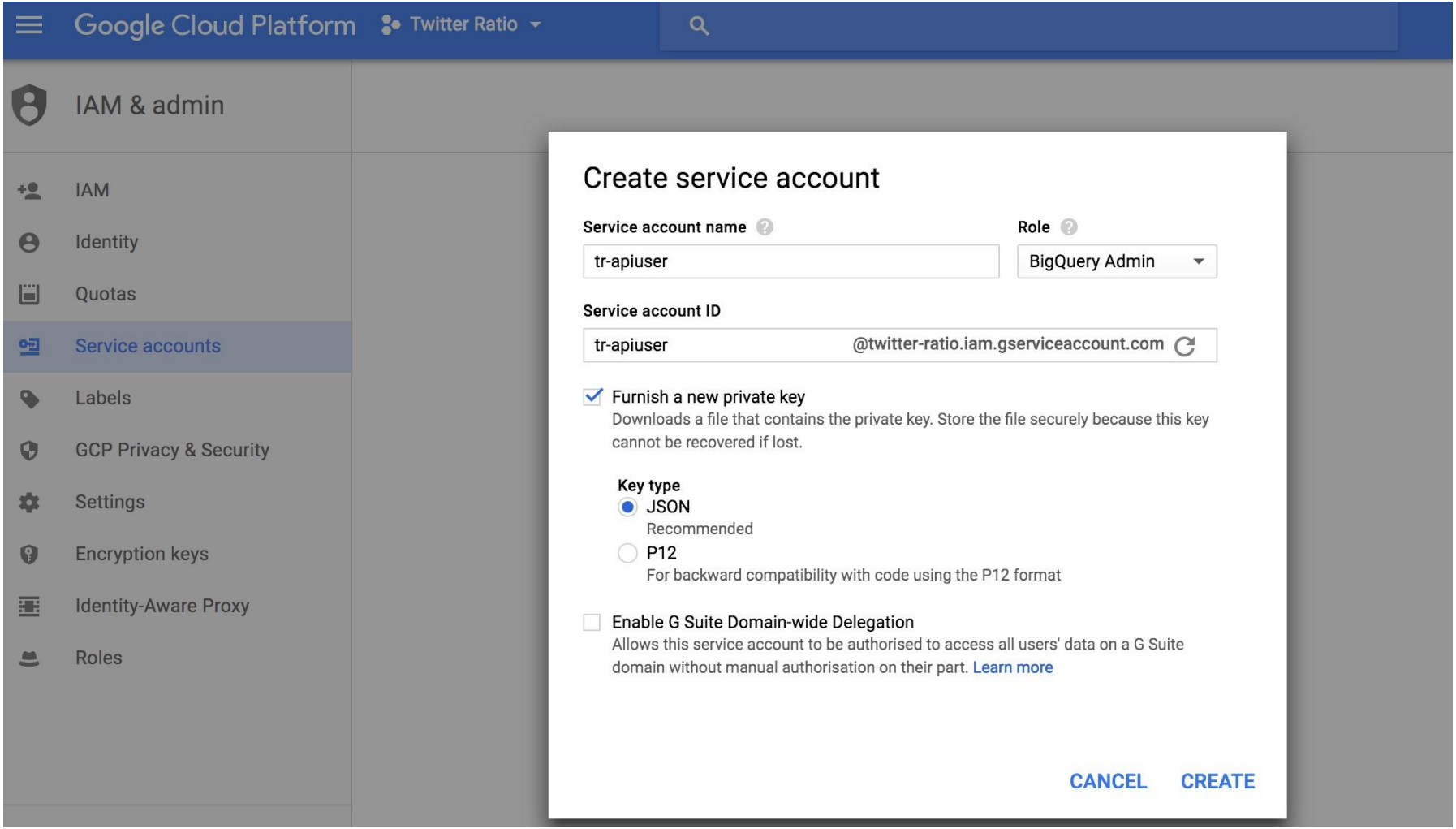
*http://www.esquire.com/news-politics/news/a54440/twitter-ratio-reply/

Our data

- About 31,000 tweets
- March 2009 to September 2017
- Excluded tweets with no text, i.e., only images or videos
- For each tweet I collected its
  - ID
  - Timestamp
  - Text
  - Number of retweets
  - Number of replies

Create a GCP project

1. Log in with a Google account and go to the GCP web console at https://console.cloud.google.com. Pull down the project menu at the top and create a new project.
2. From the main menu, go to “IAM & admin” and create a *service account*. Give this account privileges to access BigQuery. You will download a json file with the associated credentials.


Google Cloud Platform
Twitter Ratio
IAM & admin
IAM
Identity
Quotas
Service accounts
Labels
GCP Privacy & Security
Settings
Encryption keys
Identity-Aware Proxy
Roles
Create service account
Service account name
tr-apiuser
Role
BigQuery Admin
Service account ID
tr-apiuser
@twitter-ratio.iam.gserviceaccount.com
Furnish a new private key
Downloads a file that contains the private key. Store the file securely because this key cannot be recovered if lost.
Key type
JSON
Recommended
P12
For backward compatibility with code using the P12 format
Enable G Suite Domain-wide Delegation
Allows this service account to be authorised to access all users' data on a G Suite domain without manual authorisation on their part. Learn more
CANCEL
CREATE

Connect to BQ and create a dataset

```
from google.cloud import bigquery
bq_client = bigquery.Client.from_service_account_json('filename')
dataset_ref = bq_client.dataset('trump_tweets')
dataset = bigquery.Dataset(dataset_ref)
dataset = bq_client.create_dataset(dataset)
```

Create a table (option 1)

```
table_ref = dataset.table('tweet_data')
table = bigquery.Table(table_ref)
table.schema = [
    bigquery.SchemaField('id', 'STRING', mode='REQUIRED'),
    bigquery.SchemaField('text', 'STRING', mode='REQUIRED'),
    bigquery.SchemaField('date', 'TIMESTAMP', mode='REQUIRED'),
    bigquery.SchemaField('retweets', 'INTEGER', mode='REQUIRED'),
    bigquery.SchemaField('replies', 'INTEGER', mode='REQUIRED')
        ]
bq_client.create_table(table)
```

Sample json data

{"date": "2017-09-14T12:31:00+12:00", "id": "908413019050463232", "text": "Bernie Sanders is pushing hard for a single payer healthcare plan - a curse on the U.S. & its people...", "replies": 23405, "retweets": 16846}

{"date": "2017-09-14T12:18:00+12:00", "id": "908409572943126528", "text": "Spoke to President of Mexico to give condolences on terrible earthquake. Unable to reach for 3 days b/c of his cell phone reception at site.", "replies": 16970, "retweets": 11055}

Create a table (option 2)

```
table_ref = dataset.table('tweet_data')
job_config = bigquery.LoadJobConfig()
job_config.source_format = 'NEWLINE_DELIMITED_JSON'
job_config.autodetect = True
with open('twitter_data_set.json', mode='rb') as datafile:
    job = bq_client.load_table_from_file(datafile, table_ref,
            job_config=job_config)
```

Run a query

```
from google.cloud import bigquery
bq_client = bigquery.Client.from_service_account_json('filename')

query_string = 'SELECT id, replies,retweets, replies/retweets AS
ratio, date, text FROM `twitter-ratio.trump_tweets.tweet_data`
ORDER BY ratio DESC LIMIT 1'

query_job = bq_client.query(query_string)
print(query_job.state)
result = list(query_job.result())
```

I was going to show the tweet here and we'd get a good laugh out of it because it would be something ridiculous. Instead, yesterday I noticed that something was wrong with my data set. Sorry.

Stick with me though, because we're going to do some more queries that get us past this problem.

```
SELECT COUNT(*) FROM `twitter-ratio.trump_tweets.tweet_data`
WHERE retweets = 0
```

```
SELECT COUNT(*) FROM `twitter-ratio.trump_tweets.tweet_data`
WHERE retweets = 0

> 0
```

Another problem
(not the one that tripped me up earlier)

- If the twitter ratio really is a thing, it probably only holds for tweets with a certain threshold number of replies and retweets.
- But what is a good value to use?
- We can look at this using a *named parameter.*

Query with a named parameter

```
query_string = 'SELECT id, replies,retweets, replies/retweets AS
ratio, date,text FROM `twitter-ratio.trump_tweets.tweet_data`
WHERE replies > @threshold AND retweets > @threshold ORDER BY
ratio DESC LIMIT 1'
```

Queries with a named parameter

```
jobs = []
for t in (5, 10, 50, 100, 500, 1000):
    param = bigquery.ScalarQueryParameter('threshold', 'INTEGER', t)
    job_config = bigquery.QueryJobConfig()
    job_config.query_parameters = [param]
    jobs.append(bq_client.query( query_string, job_config=job_config))

results = [list(job.result()) for job in jobs]
```

Donald J. Trump

@realDonaldTrump

Follow

Who is winning the debate so far (just last name)? #DemDebate

2:33 PM - 14 Oct 2015

1,026 Retweets **3,184** Likes

9.4K 1.0K 3.2K

Donald J. Trump
@realDonaldTrump

Follow

Made additional remarks on Charlottesville and realize once again that the #Fake News Media will never be satisfied...truly bad people!

10:29 AM - 15 Aug 2017

33,546 Retweets **127,021** Likes

79K 34K 127K

But how does Trump do on average?

```
query_string = 'SELECT AVG(replies/retweets) AS average FROM
`twitter-ratio.trump_tweets.tweet_data` WHERE replies > 500 AND
retweets > 500'
query_job = bq_client.query(query_string)
result = list(query_job.result())[0]
print(result.average)
```

But how does Trump do on average?

```
query_string = 'SELECT AVG(replies/retweets) AS average FROM
`twitter-ratio.trump_tweets.tweet_data` WHERE replies > 500 AND
retweets > 500'
query_job = bq_client.query(query_string)
result = list(query_job.result())[0]
print(result.average)

0.5571272849766907
```

Broken down year by year

2009: N/A
2010: N/A
2011: 0.44
2012: 0.63
2013: 0.47
2014: 0.34
2015: 0.46
2016: 0.36
2017: 1.04

High ratio tweets (i.e., ratio > 2) year by year

2009: 0
2010: 0
2011: 0
2012: 6
2013: 1
2014: 0
2015: 7
2016: 4
2017: 112
Of a total of 6380 tweets considered, so about 2%

Cleaning up

Once we've finished working with our data we may want to get rid of our dataset to avoid incurring storage charges.

```
from google.cloud import bigquery
bq_client = bigquery.Client.from_service_account_json('filename')
dataset_ref = bq_client.dataset('trump_tweets')
dataset = bigquery.Dataset(dataset_ref)
bq_client.delete_dataset(dataset)
```

Sample code and data

https://github.com/tclark/kiwi-pycon-17