databricks Module 4 assignment 1

(https://databricks.com)

1. Which client/dataset did you select and why?

- I am using the Lobbyists4America dataset because I wanted to conduct a sentiment analysis of the tweet text.

Potential analyses:

- Who are the top tweeters and what do they tweet?
- Who do the top tweeters reply to?
- There are three tweets with >50 replies. What do the tweets say, who replied, and was it positively recieved or negatively recieved in the replies? (status_id: 590983916758163457, 857567477932462080, 841384463745589248)
- Sentiment analysis of tweets. What language is used and how does the sentiment in the tweet relate to sentiments in the replies?
- Network analysis of who replies to whom and how often

2. Describe the steps you took to import and clean the data.

- Data was uploaded to databricks and table was created from the DBFS using the built-in table creator. The variable created_at in the tweets table was interpreted as a bigint type, so was converted to date with to_timestamp(); created_at in the users table was read in as a string, but the data only show when the user's account was made, so the data were ignored for now.
- In 2008 and 2017, the data are not complete, with 2017 specifically only collecting data from half the year
- Several columns of data did not seem relevant to the analysis in both tables, so a truncated version was generated. of each table: tweets_clean and users_clean.
- The "retweeted" variable does not seem to represent whether the tweet was actually retweeted. In the "retweet_count" column, there are values even when "retweeted" is FALSE. Removing "retweeted" and keeping "retweet_count" as the measure for whether the tweet was retweeted.
- The users table isn't super interesting. It's mostly to just provide a little more info on the tweeter, so not much to be cleaned. Just truncated some of the data.

3. Perform initial exploration of data and provide some screenshots or display some stats of the data you are looking at.

- See output below for data exploration

4. Create an ERD or proposed ERD to show the relationships of the data you are exploring.

- I considered splitting the data set up into smaller chunks for the ERD, but I think it makes sense to just keep the tweets and users tables seperate and join by user_id to merge. There are some optimizations that can make the database a little cleaner, such as pulling out the columns that track quoted tweets and replies, but it's a relatively small dataset (~1.5 Gb) so the optimization seems unnecessary.
- Thus, the ERD for these tables is just the association between the tweets table and the users table, which are linked by user_id

```
DESCRIBE tweets
```

Table

	col_name	data_type
1	contributors	string
2	coordinates	struct<coordinates:array<double>,type:string>
3	created_at	bigint
4	display_text_range	array<bigint>
5	entities	struct<hashtags:array<struct<indices:array<bigint>,text:string>>,media:array<struct<display_url:string,expanded_url:string,id:int,id_str:string,indices:array<bigint>,media_url:string,media_url_https:string,sizes:struct<large:struct<h:bigint,resize:string,w:bigt>,medium:struct<h:bigint,resize:string,w:bigint>,small:struct<h:bigint,resize:string,w:bigint>,thumb:struct<h:bigint,resize:strin:bigint>>,source_status_id:bigint,source_status_id_str:string,source_user_id:bigint,source_user_id_str...
6	extended_entities	struct<media:array<struct<additional_media_info:struct<call_to_actions:struct<visit_site:struct<url:string>,watch_now:struct<ustring>>,description:string,embeddable:boolean,monetizable:boolean,source_user:struct<contributors_enabled:boolean,created:string,default_profile:boolean,default_profile_image:boolean,description:string,entities:struct<description:struct<urls:array<st<display_url:string,expanded_url:string,indices:array<bigint>,url:string>>>,url:struct<urls:array<struct<display_...
7	favorite_count	bigint

8	favorited	boolean

32 rows

```
DESCRIBE users
```

Table

	col_name	data_type
1	contributors_enabled	boolean
2	created_at	string
3	default_profile	boolean
4	default_profile_image	boolean
5	description	string
6	entities	struct<description:struct<urls:array<struct<display_url:string,expanded_url:string,indices:array<bigint>,url:string> rls:array<struct<display_url:string,expanded_url:string,indices:array<bigint>,url:string>>>>
7	favourites_count	bigint

42 rows

Note that created_at in both datasets are not dates (tweets inferred timestamps users inferred string?) and need to be converted using to_timestamp()

This code should work but databricks doesn't seem to support adding columns to tables, so I'll just hard convert to a new table later

```
ALTER TABLE tweets ADD created_at_date AS
    (cast(to_timestamp(created_at) AS DATE));
```

```
ParseException:
[PARSE_SYNTAX_ERROR] Syntax error at or near 'created_at_date'.(line 1, pos 23)

== SQL ==
ALTER TABLE tweets ADD created_at_date AS
-----------------------^^^
    (cast(to_timestamp(created_at) AS DATE))
```

```
SELECT *
FROM tweets
LIMIT 10
```

Table

	contributors	coordinates	created_at	display_text_range	entities
1	null	null	1217870931	[0, 74]	{"hashtags": [], "media": null, "symbols": [], "urls": [], "user_mentions": []}
2	null	null	1218049485	[0, 25]	{"hashtags": [], "media": null, "symbols": [], "urls": [], "user_mentions": []}
3	null	null	1218054936	[0, 65]	{"hashtags": [], "media": null, "symbols": [], "urls": [], "user_mentions": []}
4	null	null	1218117172	[0, 37]	{"hashtags": [], "media": null, "symbols": [], "urls": [], "user_mentions": []}

10 rows

Let's take a look at the "entities" entry

```
SELECT entities, COUNT(*)
FROM tweets
GROUP BY entities
ORDER BY (COUNT(*)) DESC
LIMIT 10
```

Table

	entities	count(1)
1	▸ {"hashtags": [], "media": null, "symbols": [], "urls": [], "user_mentions": []}	95822
2	▸ {"hashtags": [], "media": null, "symbols": [], "urls": [], "user_mentions": [{"id": 301549400, "id_str": "301549400", "indices": [3, 19], "name": "House Appropriations", "screen_name": "HouseAppropsGOP"}, {"id": 550401754, "id_str": "550401754", "indices": [22, 35], "name": "Hal Rogers", "screen_name": "RepHalRogers"}]}	458
3	▸ {"hashtags": [], "media": null, "symbols": [], "urls": [], "user_mentions": [{"id": 31122582, "id_str": "31122582", "indices": [3, 17], "name": "House OversightDems", "screen_name": "OversightDems"}, {"id": 787373558, "id_str": "787373558", "indices": [20, 32], "name": "Elijah E. Cummings", "screen_name": "RepCummings"}]}	435
4	▸ {"hashtags": [], "media": null, "symbols": [], "urls": [], "user_mentions": [{"id": 18916432, "id_str": "18916432", "indices": [3, 15], "name": "Paul Ryan", "screen_name": "SpeakerRyan"}]}	387
5	▸ {"hashtags": [], "media": null, "symbols": [], "urls": [], "user_mentions": [{"id": 29450962, "id_str": "29450962", "indices": [3, 16], "name": "John Lewis", "screen_name": "repjohnlewis"}]}	355
6	▸ {"hashtags": [], "media": null, "symbols": [], "urls": [], "user_mentions": [{"id": 12788332, "id_str": "12788332", "indices": [3, 19], "name": "Ways and Means", "screen_name": "WaysandMeansGOP"}]}	350
	▸ {"hashtags": [], "media": null, "symbols": [], "urls": [], "user_mentions": [{"id": 19739126, "id_str": "19739126", "indices": [3, 13],	270

10 rows

Looks like this is nested JSON data for each tweet, summarizing metadata related to the tweet, such as the use of hashtags, links, and user mentions.

It looks like this data is stored elsewhere in the table, so I'm going to ignore this for now.

Let's explore some of the other variables

```
SELECT favorited,
COUNT(*)
FROM tweets
GROUP BY favorited
ORDER BY COUNT(*) DESC
```

Table

	favorited	count(1)
1	false	1243352
2	true	18

2 rows

```
SELECT in_reply_to_screen_name,
COUNT(*)
FROM tweets
GROUP BY in_reply_to_screen_name
ORDER BY COUNT(*) DESC
```

Table

	in_reply_to_screen_name	count(1)
1	null	1177959
2	SenatorDurbin	1309
3	SenGillibrand	1011
4	SenFeinstein	798
5	RepMcGovern	753
6	SenBobCasey	736
7	SenJeffMerkley	612

10,000 rows | Truncated data

```
SELECT in_reply_to_status_id,
COUNT(*)
FROM tweets
GROUP BY in_reply_to_status_id
ORDER BY COUNT(*) DESC
```

Table

	in_reply_to_status_id	count(1)
1	null	1189224
2	590983916758163500	150
3	857567477932462100	100
4	841384463745589200	52
5	846835237099388900	12
6	441258035155972100	11
7	858036241278935000	11

10,000 rows | Truncated data

```
SELECT in_reply_to_user_id,
  COUNT(*)
FROM tweets
GROUP BY in_reply_to_user_id
ORDER BY COUNT(*) DESC
```

Table

	in_reply_to_user_id	count(1)
1	null	1177959
2	247334603	1309
3	72198806	1011
4	476256944	798
5	242426145	753
6	171598736	736
7	29201047	612

10,000 rows | Truncated data

```
SELECT is_quote_status,
COUNT(*)
FROM tweets
GROUP BY is_quote_status
ORDER BY COUNT(*) DESC
```

Table

	is_quote_status	count(1)
1	false	1186059
2	true	57311

2 rows

```
SELECT
  screen_name,
  SUM(favorite_count)
FROM tweets
GROUP BY screen_name
ORDER BY SUM(favorite_count) DESC
```

Table

	screen_name	sum(favorite_count)
1	realDonaldTrump	137775878
2	SenSanders	33039433
3	POTUS	9559858
4	SenWarren	8442047
5	RepAdamSchiff	3917791
6	ChrisMurphyCT	3359728
7	timkaine	3304474

545 rows

```
SELECT
  retweeted,
  retweet_count
FROM tweets
```

Table

	retweeted	retweet_count
1	false	0
2	false	0
3	false	0
4	false	0
5	false	0
6	false	0
7	false	0

10,000 rows | Truncated data

```
SELECT
  to_timestamp(created_at)
FROM tweets
LIMIT 10
```

Table

	to_timestamp(created_at)
1	2008-08-04T17:28:51.000+0000
2	2008-08-06T19:04:45.000+0000
3	2008-08-06T20:35:36.000+0000
4	2008-08-07T13:52:52.000+0000
5	2008-08-07T15:12:05.000+0000
6	2008-08-07T18:35:25.000+0000
7	2008-08-18T14:07:35.000+0000

10 rows

```
SELECT *
FROM tweets
WHERE is_quote_status = true
LIMIT 10
```

Table

	contributors	coordinates	created_at	display_text_range	entities
1	null	null	1311283147	▶ [0, 23]	▶ {"hashtags": [], "media": null, "symbols": [], "urls": [{"display_url": "twitte "https://twitter.com/#!/RepublicanStudy/status/94152153911930882", "in "user_mentions": []}
2	null	null	1335454907	▶ [0, 105]	▶ {"hashtags": [{"indices": [60, 69], "text": "DeathTax"}], "media": null, "sym "twitter.com/#!/AustinScott...", "expanded_url": "https://twitter.com/#!/Au [84, 105], "url": "https://t.co/VoMifTkP"}], "user_mentions": [{"id": 2095719 "American Farm Bureau", "screen_name": "FarmBureau"}]}
3	null	null	1335454978	▶ [0, 123]	▶ {"hashtags": [{"indices": [64, 73], "text": "DeathTax"}], "media": null, "sym "twitter.com/#!/AustinScott...", "expanded_url": "https://twitter.com/#!/Au [102, 123], "url": "https://t.co/VoMifTkP"}], "user_mentions": [{"id": 439708 TV", "screen_name": "OfficialRFDTV"}]}
4	null	null	1336143859	▶ [0, 140]	▶ {"hashtags": [], "media": null, "symbols": [], "urls": [], "user_mentions": [{" "name": "ClotureClub.com", "screen_name": "ClotureClub"}, {"id": 298219("Селиверст Тетерин", "screen_name": "RepLankford"}]}
5	null	null	1341527805	▶ [0, 140]	▶ {"hashtags": [], "media": null, "symbols": [], "urls": [{"display_url": "twitte "https://twitter.com/rorycooper/status/220983831606464514", "indices": "user_mentions": [{"id": 640893, "id_str": "640893", "indices": [3, 14], "nam
	null	null	1371155147	▶ [0, 140]	▶ {"hashtags": [{"indices": [123, 126], "text": "VA"}, {"indices": [127, 133], "t [{"display_url": "twitter.com/RobWittman/sta...", "expanded_url": "https://t

10 rows

```
SELECT lang,
COUNT(*)
FROM tweets
GROUP BY lang
ORDER BY COUNT(*) DESC
```

Table

	lang	count(1)
1	en	1226949
2	und	8137
3	es	5108
4	fr	674
5	in	346
6	ro	261
7	tl	245

37 rows

```
SELECT YEAR(to_timestamp(created_at)),
  COUNT(*)
FROM tweets
GROUP BY YEAR(to_timestamp(created_at))
ORDER BY YEAR(to_timestamp(created_at)) DESC
```

Table Visualization 1

	year(to_timestamp(created_at))	count(1)
1	2017	229362
2	2016	354942
3	2015	258256
4	2014	168308
5	2013	124439
6	2012	50791
7	2011	35163

10 rows

```
SELECT DATE(to_timestamp(created_at)),
  COUNT(*)
FROM tweets
GROUP BY DATE(to_timestamp(created_at))
ORDER BY DATE(to_timestamp(created_at)) DESC
```

The tweet frequency has been increasing since 2008. The sharp decrease in tweets in 2017 is due to the data in 2017 truncating in June (so halfway through the year).

Based on the exploration above, the following features should be pulled from the tweets.json:

- id
- user_id
- created_at
- text
- favorited
- favorite_count
- in_reply_to_status_id
- in_reply_to_user_id
- is_quote_status
- quoted_status_id
- retweet_count

Let's take a look at the users tables now

```
SELECT *
FROM users
LIMIT 10
```

Table

	contributors_enabled	created_at	default_profile	default_profile_image	description
1	false	1417384037	true	false	Official Twitter page of Alaska Governor Bill Walker; hon

2	false	1240239576	false	false	U.S. Senator from Minnesota. Follows, Retweets, Replies
3	false	1366837593	false	false	Congressman for Maryland's 4th Congressional District,
4	false	1300739574	false	false	Husband of 43 yrs, Dad of 4, Papaw of 6. Lifelong Arkans Sec. of Homeland Security.
5	false	1294329706	false	false	I am proud to represent the 8th Congressional District o
6	false	1255553223	true	false	
	false	1247263375	false	false	U.S. Senator for Louisiana

10 rows

```
SELECT lang,
COUNT(*)
FROM users
GROUP BY lang
ORDER BY COUNT(*) DESC
```

Table

	lang	count(1)
1	en	548

1 row

From the users.json table, we want the following features:

- description
- favourites_count
- followers_count
- following
- id
- location
- name
- screen_name
- statuses_count
- time_zone
- verified

Let's make the tables

```
CREATE TABLE tweets_clean AS
  SELECT
    id,
    user_id,
    to_timestamp(created_at) AS created_at,
    text,
    favorited,
    favorite_count,
    in_reply_to_status_id,
    in_reply_to_user_id,
    is_quote_status,
    quoted_status_id,
    retweet_count
  FROM tweets
```

Query returned no results

Query returned no results