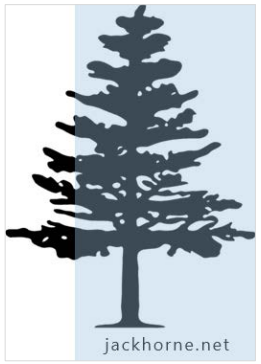
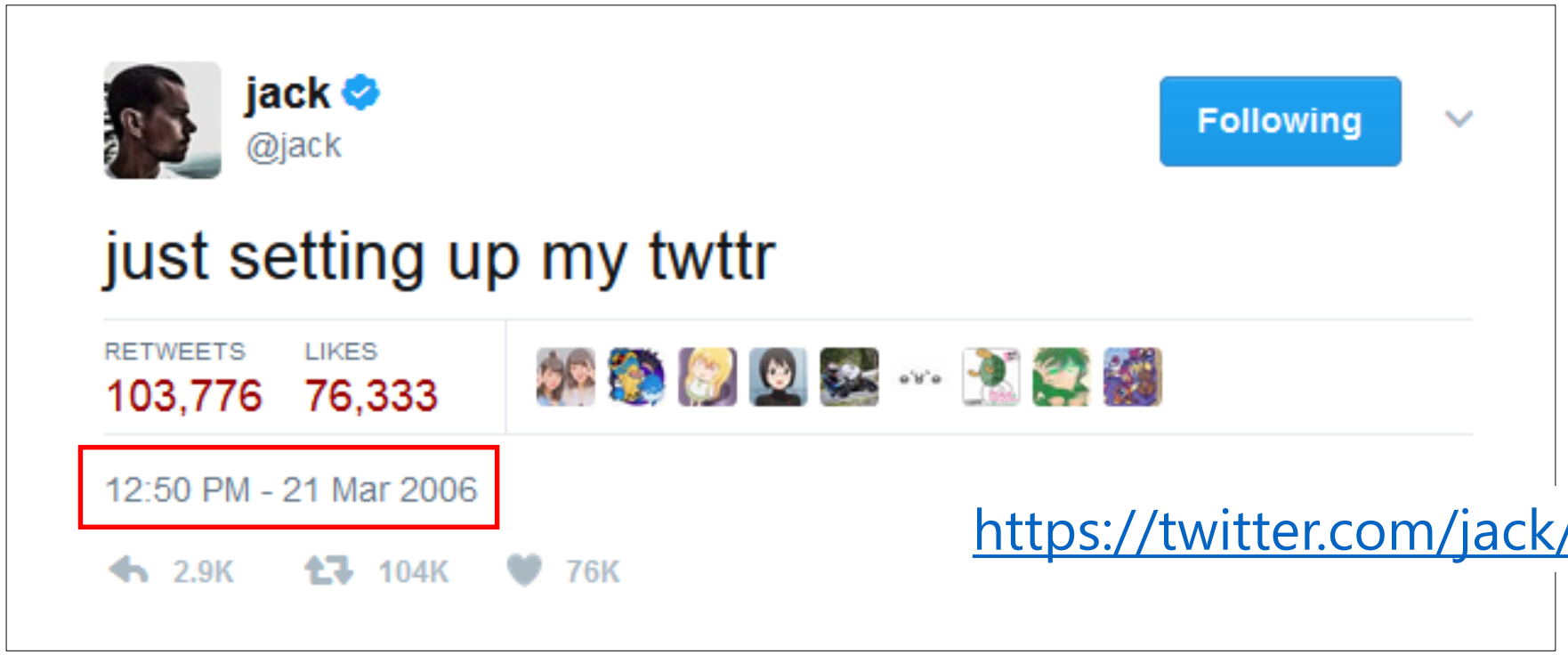


Sampling in the River Twitter: A DIY Tool for Social Media Tracking via the REST API

Jack Horne <jack@jackhorne.net>
Jane Tang <jane.tang@marumatchbox.com>

maru/matchbox

Advanced Research Techniques Forum
June 26. 2017
Seattle, WA



https://twitter.com/jack/status/20

March 2006 → 1 user (@jack)

May 2017 → 320 million average monthly users

The 1 billionth tweet came **31 months** later in November 2008

https://twitter.com/folkhero/status/999999999

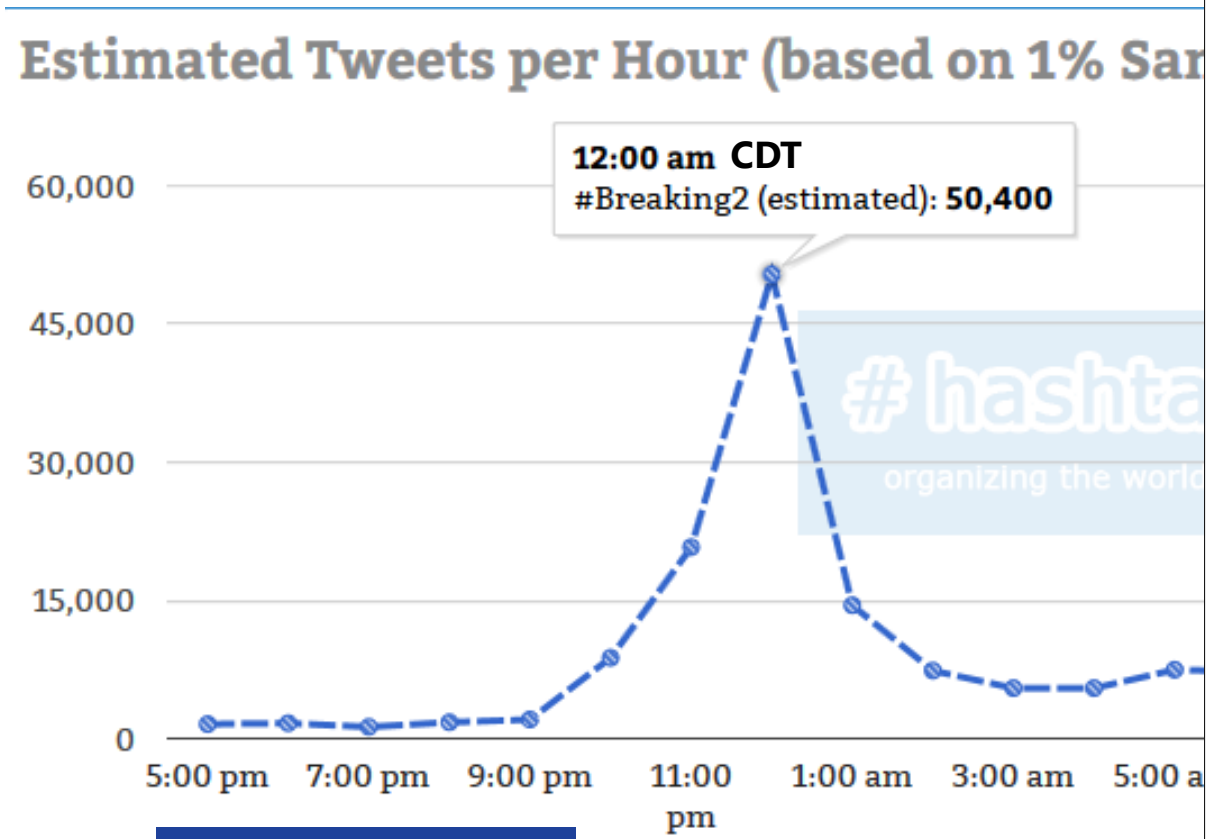
This is the 999,999,999th tweet; The 1 billionth tweet came from a private account, so we don't know what it was

Today there are ~650,000,000 tweets per day, or about 1 billion tweets every **37 hours**

Screenshot from internetlivestats.com at 9:41pm 6 May 2017

Tracking Twitter Content & Hashtags

Lots of tools … Few are free … Limited query capabilities



maru/matchbox

SOCIALBEARING

Kentucky Derby post time: 3:38pm (PDT) 6 May 2017;
Data pulled: 4:00pm

FILTERS

Twitter search & analytics for '#kentuckyderby'

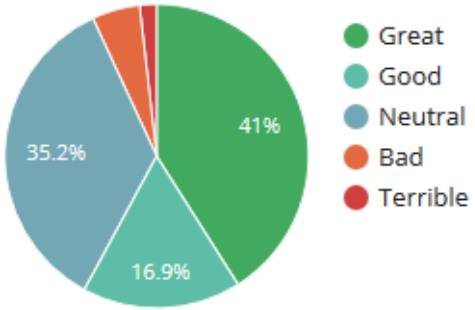
Showing all public tweets that match your query. Tweets are loaded 100 at a time, up to the past **9 days. Tweets returned may be limited**

Tweet | CSV NEW

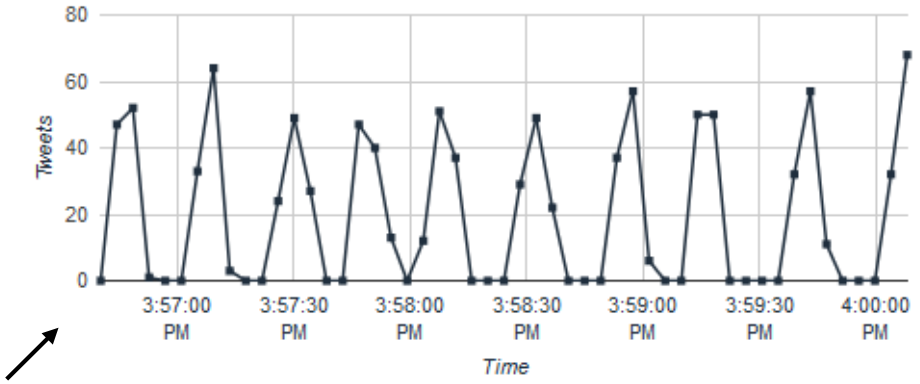
TWEETS	TIMEFRAME	REACH	IMPRESSIONS	TOTAL RT'S	TOTAL FAVES	REPLIES	HIDDEN
1000	3m	9,576,995	9,880,223	24,235	58,448	20	6

LOAD MORE

TWEETS BY SENTIMENT



TWEETS OVER TIME



Intermittent use of the hashtag is usually evidence of bots

WORD CLOUD

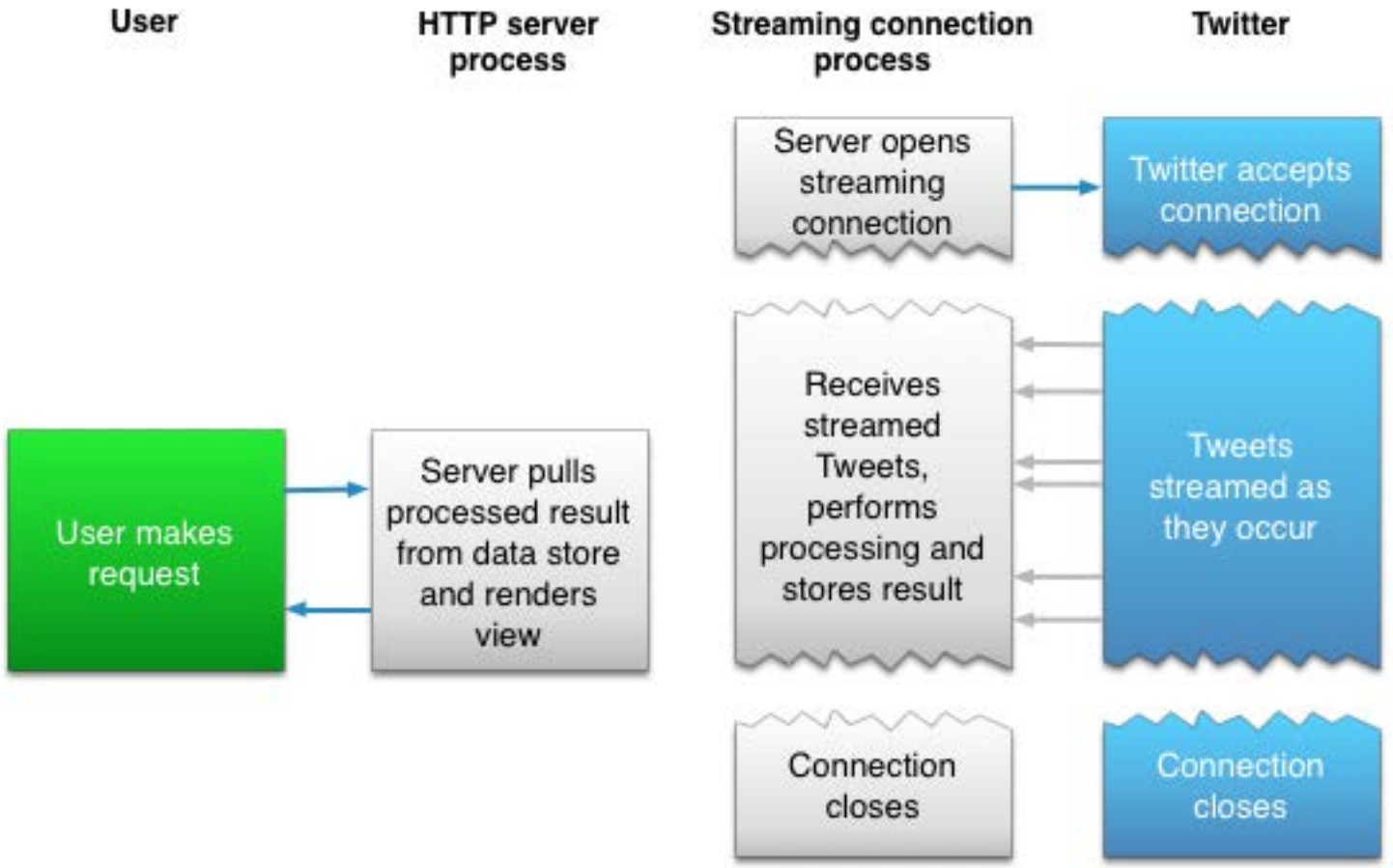
"Always Dreaming" is the name of the winning horse

maru/matchbox

Can we accurately assess high-volume Twitter content using APIs?

DIY Apps using Twitter APIs

Streaming APIs



Real-time (i.e., no fixed endpoint)

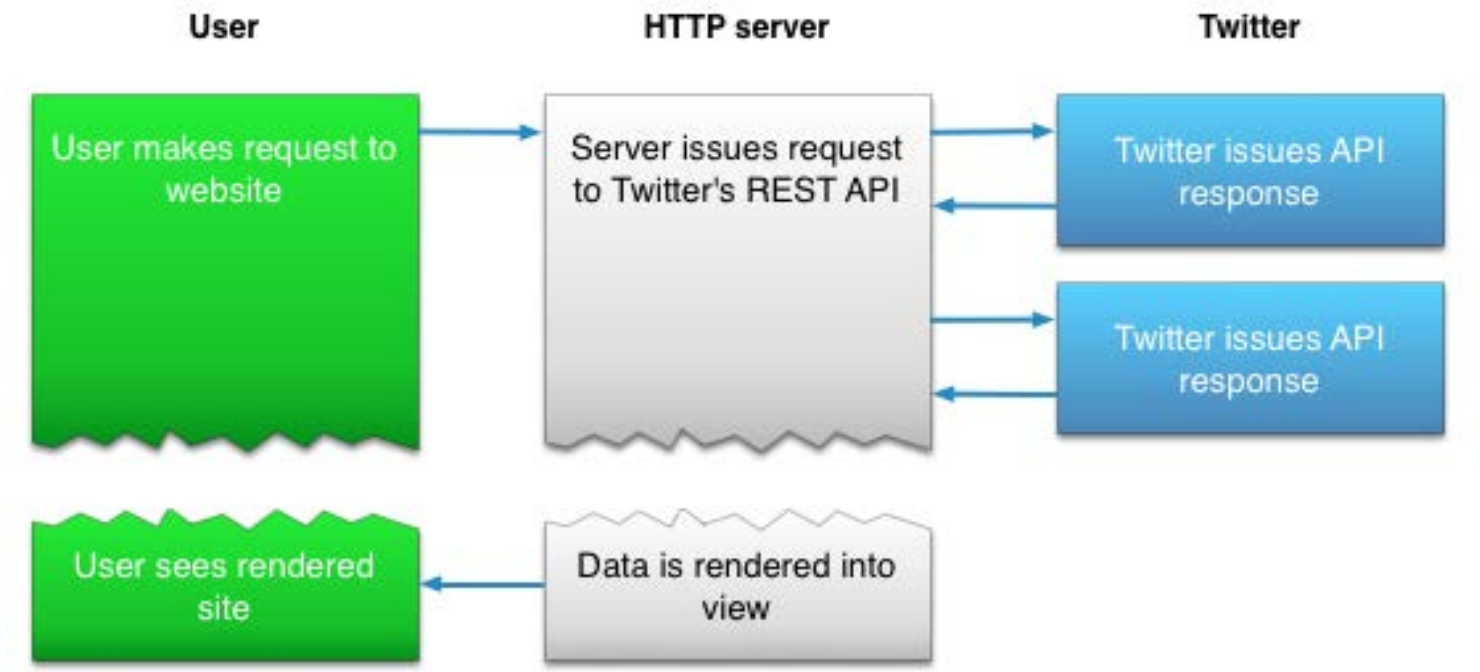
Unable to establish connections to Twitter in response to a user request

Streaming process gets Tweets, parses, filters, and/or aggregates before storing

HTTP process queries the data store in response to user requests

DIY Apps using Twitter APIs

REST APIs



Web application accepts user requests and makes one or more requests to Twitter

App then formats and prints results to user in response to initial request

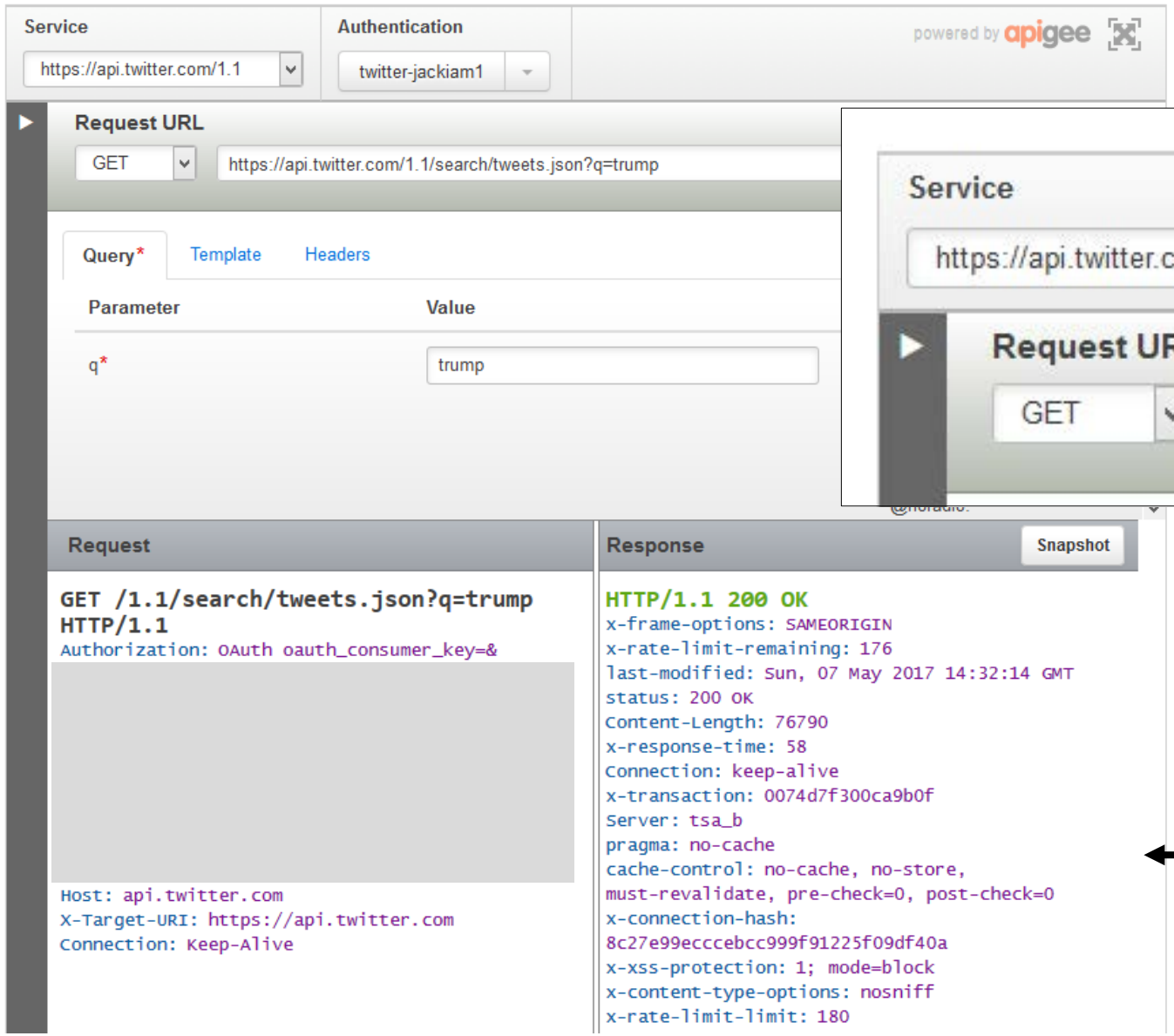
Stores only (up to) previous seven days of Tweets

Rate limited: Only a limited number of requests can be made in a given time period

We will use the REST APIs to develop our DIY approach to sample the Twitter platform

API Console

https://dev.twitter.com/rest/tools/console



signed into service using Twitter credentials

search URL: find any tweet containing the word “trump”

response (15 tweets in < 1 sec)

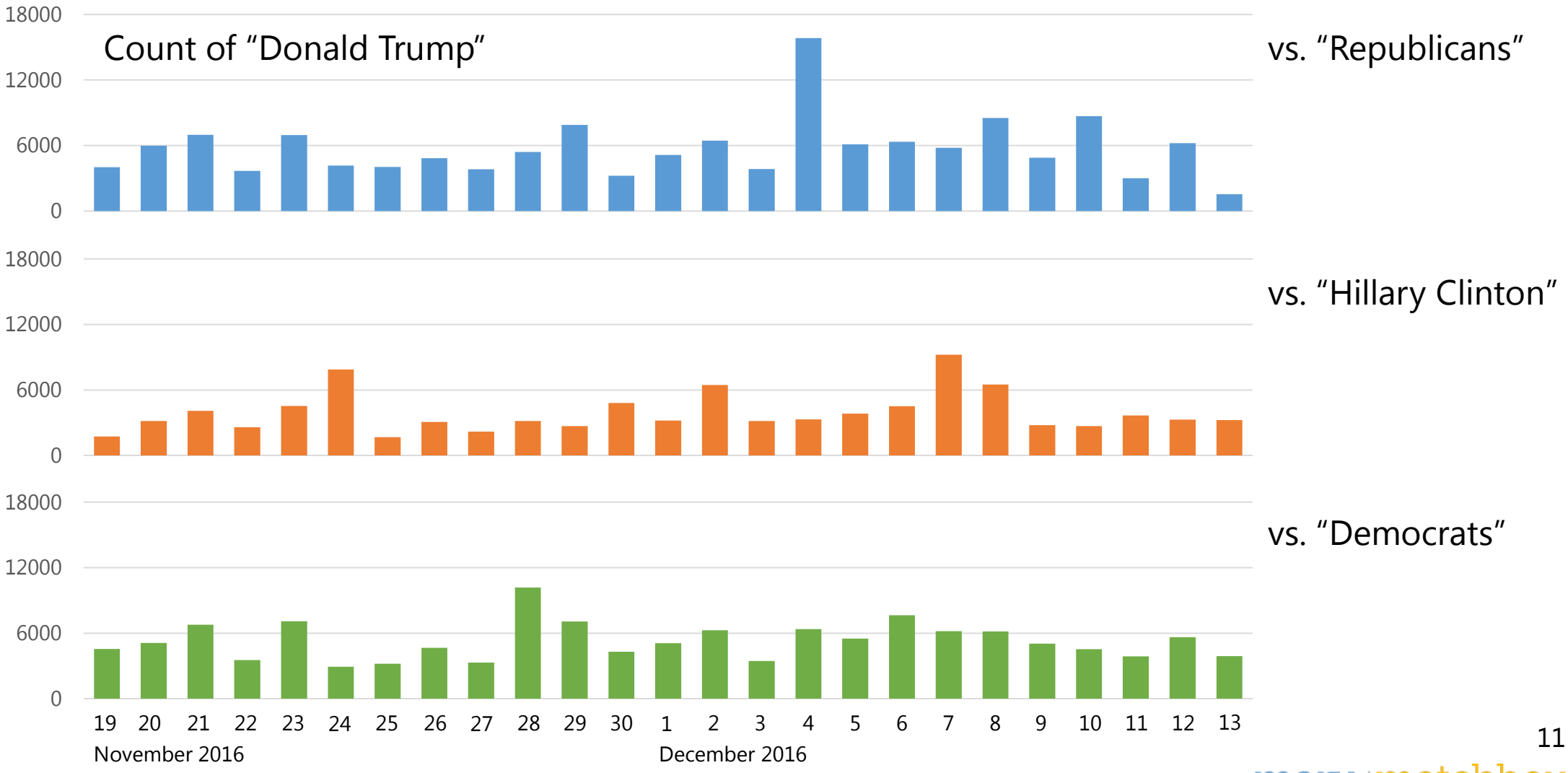
maru/matchbox

One Tweet (or part of one)


```
{"created_at":"Sun May 07 14:18:45 +0000
2017","id":861223784447172609,"id_str":"861223784447172609","text":"Trump
questions whether key funding source for historically black colleges is
constitutional https:\/\/t.co\/1rtA684283","truncated":false,"entities":
{"hashtags":[],"symbols":[],"user_mentions":[],"urls":[{"url":"https:\/\/t
.co\/1rtA684283","expanded_url":"http:\/\/wapo.st\/2qdEbT0","display_url":
"wapo.st\/2qdEbT0","indices":[93,11
en","result_type":"recent"},"source
href=\"https:\/\/about.twitter.com
rel=\"nofollow\"\u003eTweetDeck\u00
ll,"in_reply_to_status_id_str":null
o_user_id_str":null,"in_reply_to_s
"id_str":"299802277","name":"Bradd
Jaffy","screen_name":"BraddJaffy",
News \u2022 Senior news editor and
dashes \u2014 the merrier." ... }
```


https://twitter.com/braddjaffy/status/861223784447172609

Searching in Pairs (e.g., “Donald Trump” vs. “Hillary Clinton”) Does NOT Yield Consistent Results for Different Paired Terms



maru/matchbox

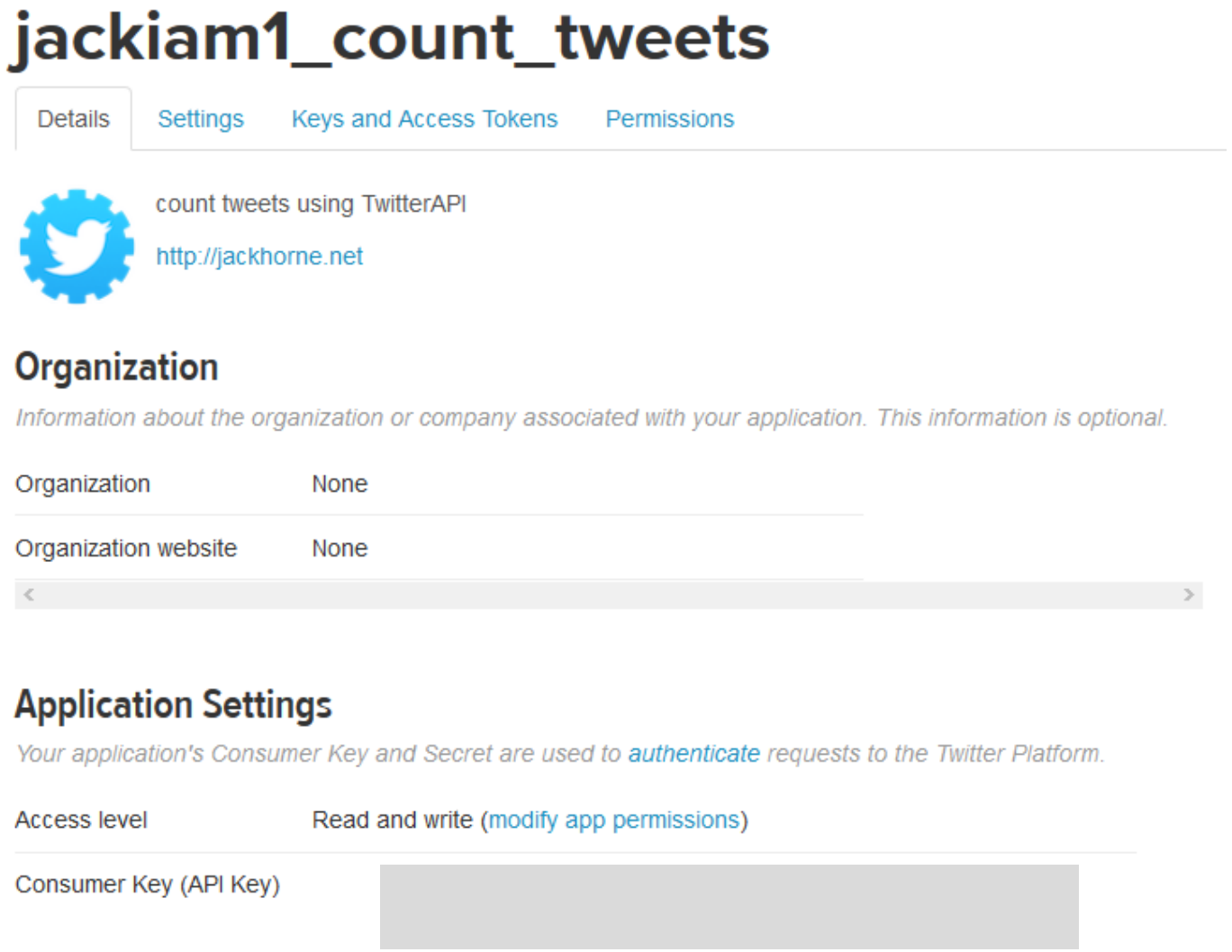
Why Such Inconsistent Results?

Rate Limits → The REST API imposes a rate limit of 180 requests per 15 minute period. Each request returns 15 tweets by default. So, max 2,700 tweets per session before rate limit is reached (a tiny fraction of the total number of tweets)

Terms are a Filter → We are pre-filtering results, and only returning tweets that contain one or more of the terms in the search string. We are not returning a "representative" sample of tweets and then post-filtering to count incidence of terms.

Complex Search Terms → More complex search terms (e.g., "Donald Trump" instead of "trump") can lead to early termination of the session.

A Bit of Python Programming



Set up a web application in a Twitter account that you control

https://apps.twitter.com

This is for authentication into the API

You don't need to ever use Twitter beyond this point, but you do need an account to access the API

A Bit of Python Programming

```
from TwitterAPI import TwitterAPI, TwitterRestPager

WORDS_TO_COUNT = ['trump', 'clinton', 'obama', 'aapl', 'googl', 'amzn', 'e']

API_KEY =
API_SECRET =
ACCESS_TOKEN =
ACCESS_TOKEN_SECRET =


api = TwitterAPI(API_KEY, API_SECRET, ACCESS_TOKEN, ACCESS_TOKEN_SECRET)
words = ' OR '.join(WORDS_TO_COUNT)
counts = dict((word,0) for word in WORDS_TO_COUNT)
tweet = 0
```

Python libraries for accessing Twitter APIs

Search string

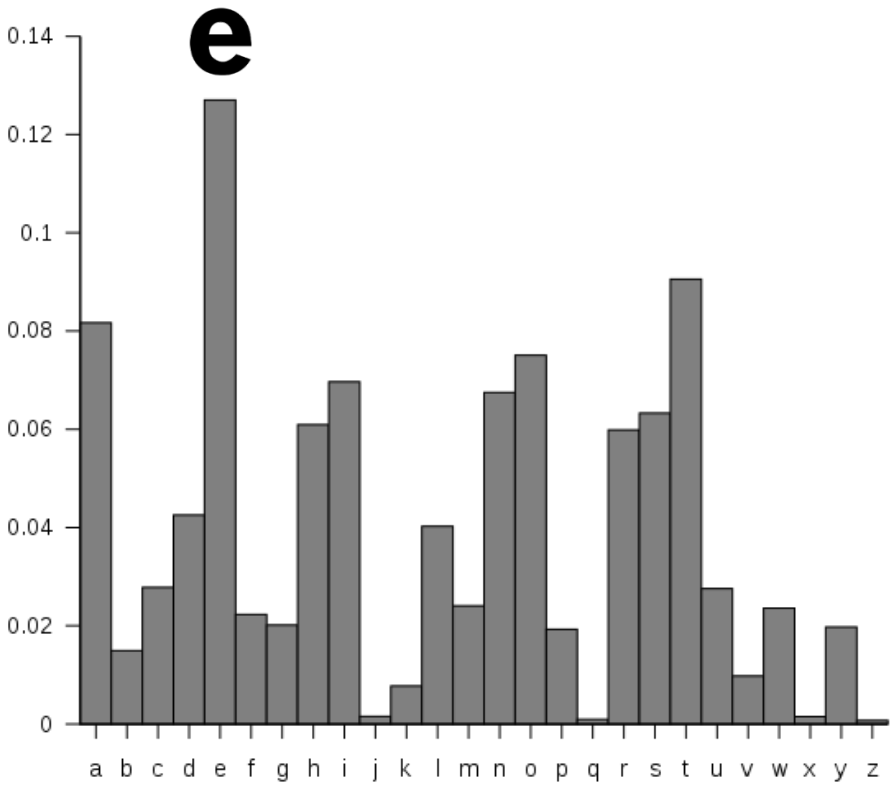
Authentication strings (OAuth2) (from the Twitter web app)

Authenticate, and set up some global variables

A Bit of Python Programming

```
terAPI, TwitterRestPager

'clinton', 'obama', 'aapl', 'googl', 'amzn', 'e']
```



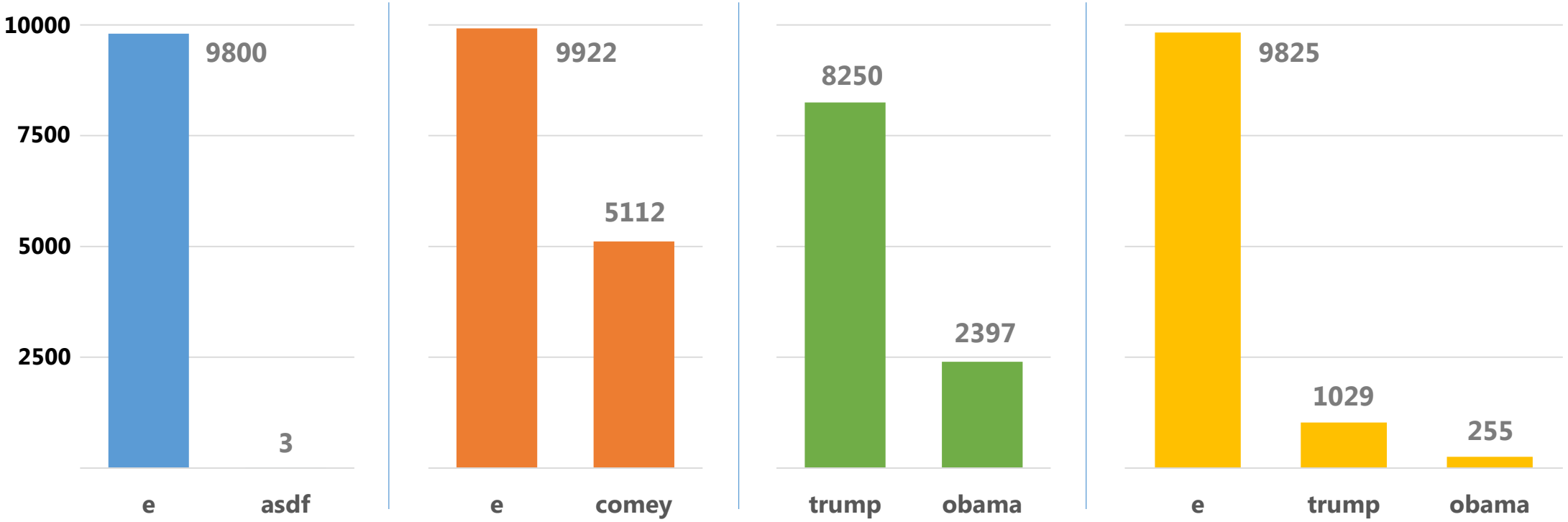
Relative frequency in the English language

Why is the letter “e” in the search string?

The letter “e” is the most common letter in the English language, and is likely to occur in the text of nearly all tweets. Including it in the search string helps ensure a representative sample of tweets, not just those containing a search term of interest.

A Bit of Python Programming

The letter “e” occurs in ~98% of all tweets, regardless of the search terms it is paired with



A Bit of Python Programming

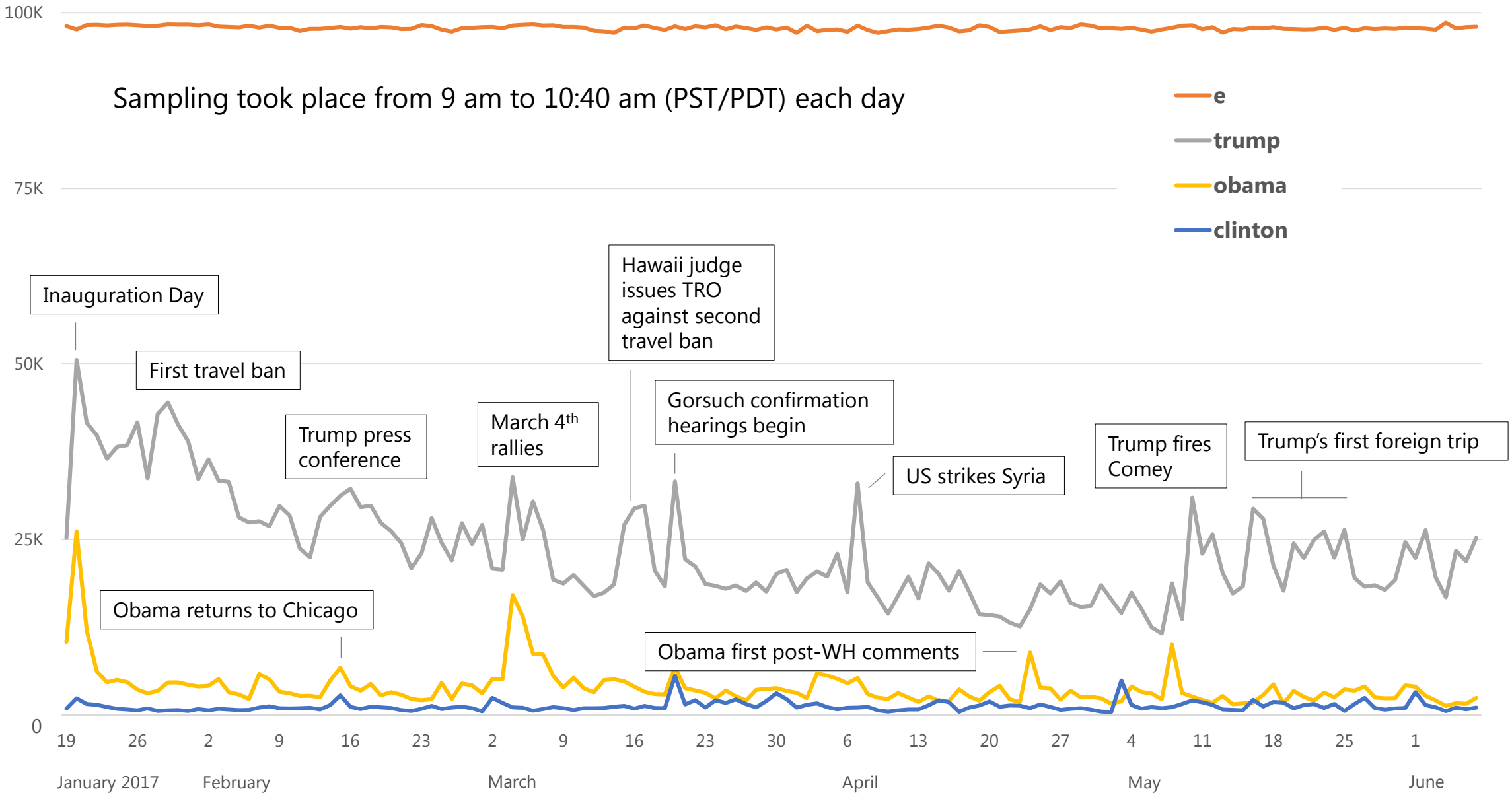
```
def process_tweet(text):
        text = text.lower()
        for word in WORDS_TO_COUNT:
                if word in text:
                        counts[word] += 1
        s = "tweets: " + str(tweet) + " " + str(counts)
        print(s)


r = TwitterRestPager(api, 'search/tweets', {'q':words, 'count':100})
for item in r.get_iterator(wait=6):
        tweet += 1
        if tweet > 100000:
                break
        if 'text' in item:
                process_tweet(item['text'])
        elif 'message' in item and item['code'] == 88:
                print('\n*** SUSPEND, RATE LIMIT EXCEEDED: %s\n' % item['message'])
                break
```

Function to process each tweet and look for the presence of words in the search string

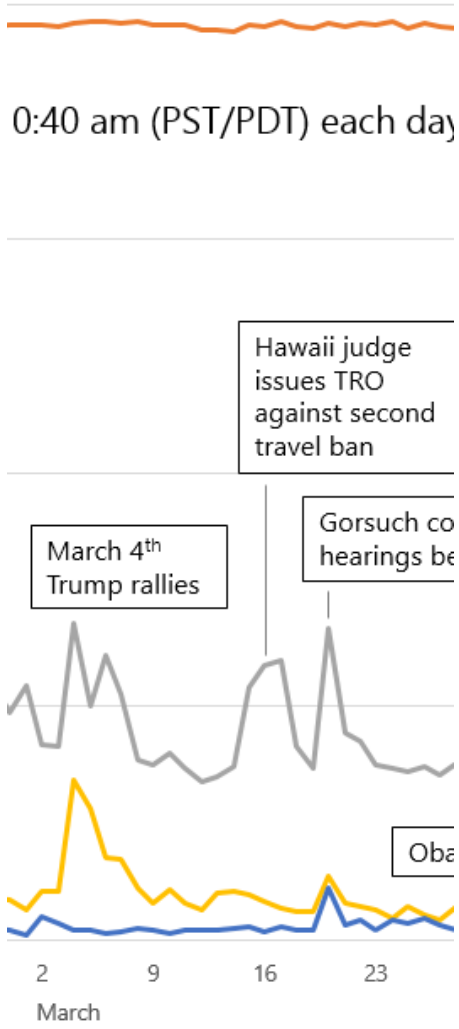
Page through tweets (100 at a time) and process each given the search string

Wait 6 sec between requests to get around rate limits



maru/matchbox

Other Things We Tried



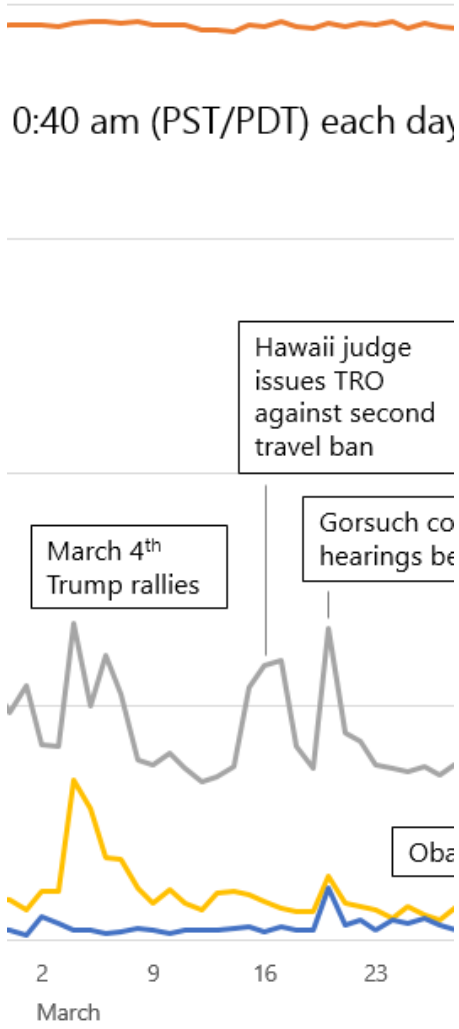
Does (Sampling) Time of Day Matter?

We sampled at random times four to six times per day for 60+ days. Overall trends over time are similar. Spikes are sensitive to breaking news events within a few hours of the actual event.

Does Sampling Size Matter?

We pulled sample sizes ranging from 500 to 100,000. Results stabilized at ~10,000, without much benefit beyond that size. Lower incidence terms may require larger sample sizes.

Other Things We Tried



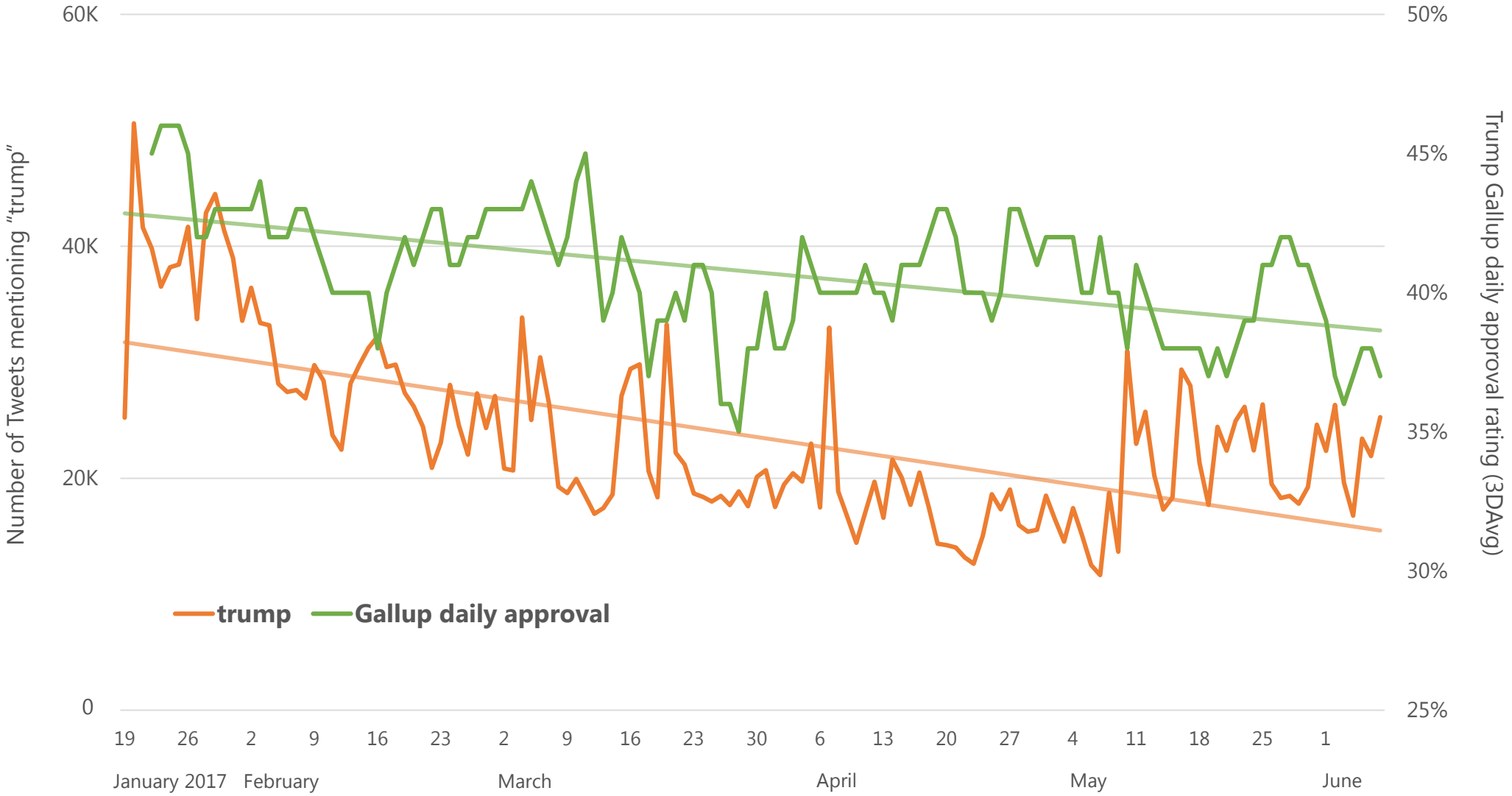
Can we just sample on “e”, Retain all of the text, and sample that text for any term, group of terms, or sentiment?

Yes, … requires more storage.

Potentially an interesting user case – as it allows you to build a mini-version of a Tweet library.

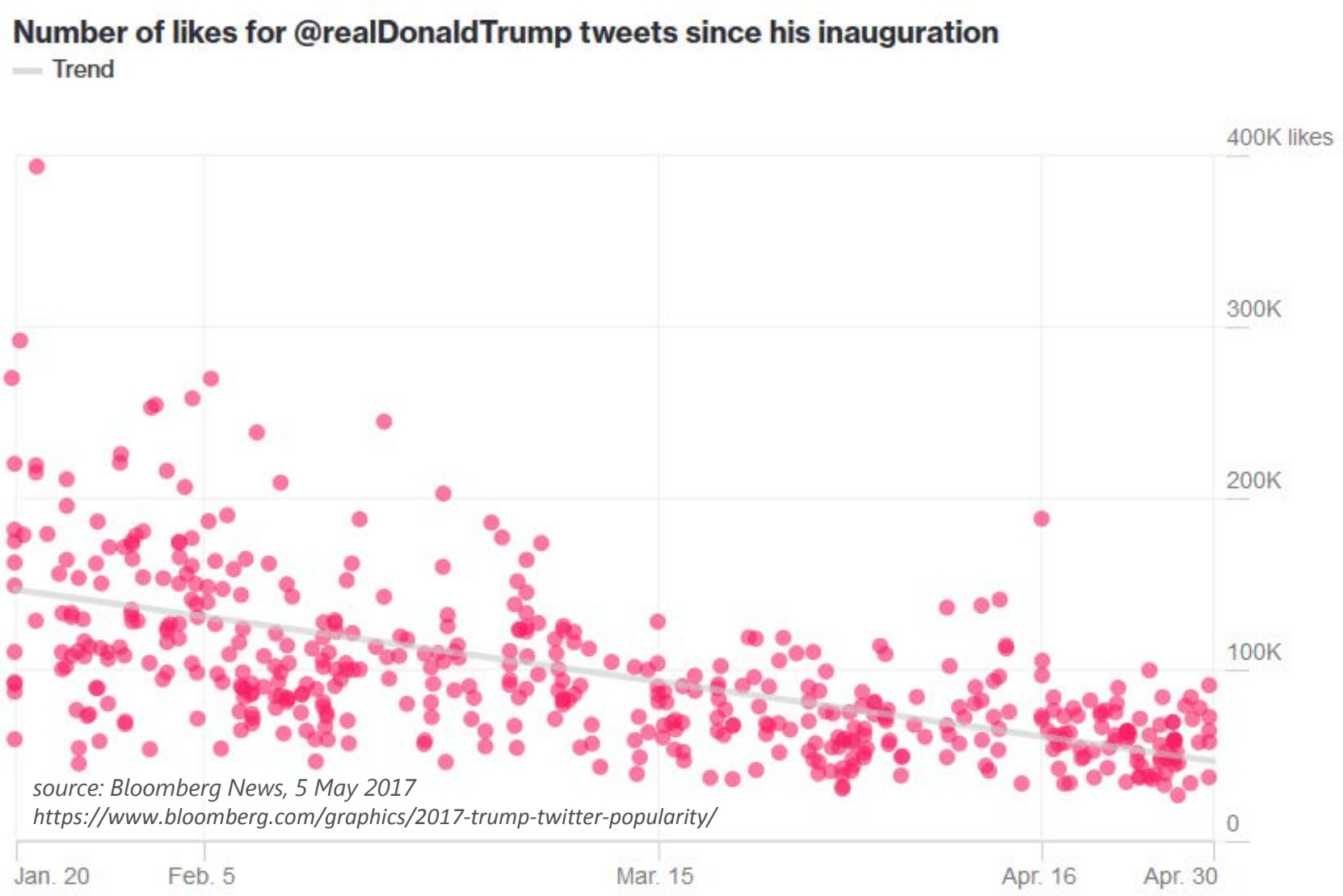
It may violate Twitter TOS to hang on to the data over an extended period of time. Check Twitter user policy: https://dev.twitter.com/overview/terms/policy#updated-policy

Does Number of Tweets Track Other Metrics?



maru/matchbox

Rate of “Likes” on Individual Tweets Appears to Track Number of Mentions



source: Bloomberg News, 5 May 2017
https://www.bloomberg.com/graphics/2017-trump-twitter-popularity/

maru/matchbox

Tweets and Brand Tracking: Kalman Filter

How Kalman Filter Works

estimation at time t | measured value | estimation at time $(t-1)$

$$\hat{X}_t = K_t \cdot X_t + (1 - K_t) \cdot \hat{X}_{t-1}$$

Kalman gain

State/prior equation

$$\tilde{x}_t = \hat{x}_{t-1} + w_t \qquad \rightarrow p(w) \sim N(0, Q)$$

Measurement/update equation

$$\hat{x}_t = x_t + v_t \qquad \rightarrow p(v) \sim N(0, R)$$

http://bilgin.esme.org/BitsBytes/KalmanFilterforDummies.aspx
http://www.cs.unc.edu/~tracker/media/pdf/SIGGRAPH2001_CoursePack_08.pdf

Equations Simplify to a Random Walk

estimation at time t | measured value | estimation at time $(t-1)$

$$\hat{X}_t = K_t \cdot X_t + (1 - K_t) \cdot \hat{X}_{t-1}$$

Kalman gain

random walk

Time update
"prediction"

(1) Prior for subsequent state

$$\tilde{x}_t = \hat{x}_{t-1}$$

external shock

(2) Prior error covariance

$$\tilde{P}_t = P_{t-1} + Q$$

random walk

Measurement update
"correction"

random measurement error

(1) Compute the Kalman gain

$$K_t = \tilde{P}_t\left(\tilde{P}_t + R\right)^{-1}$$

(2) Update estimate

$$\hat{x}_t = \tilde{x}_t + K_t(x_t - \tilde{x}_t)$$

(3) Update error covariance

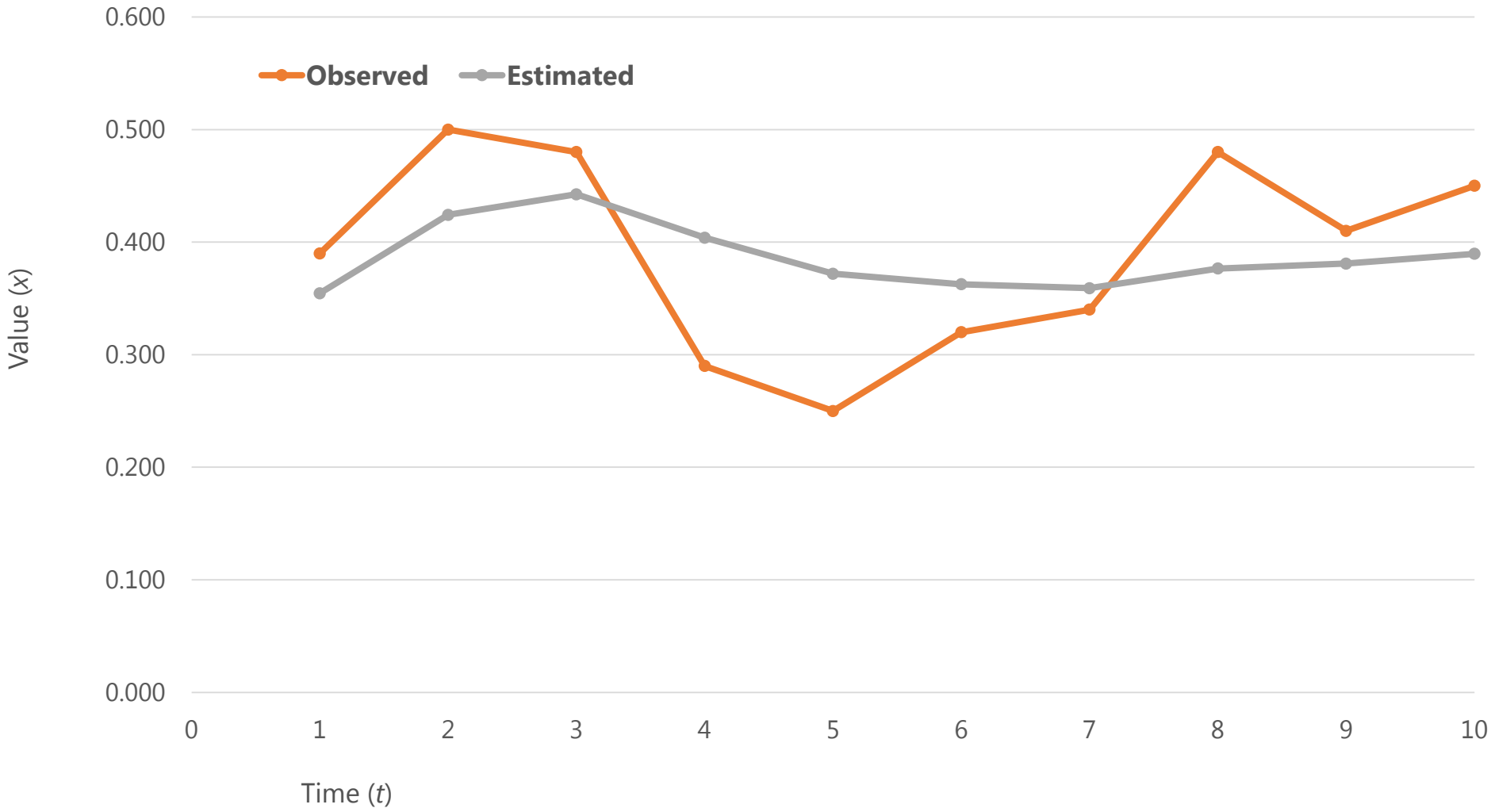
$$P_t = \tilde{P}_t(1 - K_t)$$

A Simple Example

$R = 0.1$ variance around observations (best guess) $Q = 0.001$ external shock variance

t	x_t	$\hat{x}_{t-1}$	$\tilde{P}_t$	Time update	Measurement update	$\hat{x}_t$	P_t
1	0.390	0.000	1.000	$\tilde{x}_t = \hat{x}_{t-1} = 0$ $\tilde{P}_t = P_{t-1} = 1$	$K_t = 1(1+0.1)^{-1} = 0.909$ $\hat{x}_t = 0 + 0.909(0.390 - 0) = 0.355$ $P_t = 1(1 - 0.909) = 0.091$	0.355	0.091
2	0.500	0.355	0.092	$\tilde{x}_t = 0.355$ $\tilde{P}_t = P_{t-1} + Q = 0.092$	$K_t = 0.092(0.092 + 0.1)^{-1} = 0.479$ $\hat{x}_t = 0.355 + 0.479(0.5 - 0.355) = 0.424$ $P_t = 0.092(1 - 0.479) = 0.048$	0.424	0.048
3	0.480	0.424	0.048			0.443	0.033
4	0.290	0.443	0.033			0.404	0.025
5	0.250	0.404	0.025			0.372	0.021
6	0.320	0.372	0.021			0.363	0.018

A Simple Example



Incorporating Known System Instability

$R = 0.1$ variance around observations (best guess) $Q = variable$

t	x_t	$\hat{x}_{t-1}$	$\tilde{P}_t$	q_t	K_t	$\hat{x}_t$	P_t
1	0.390	0.000	1.000	0.001	0.909	0.355	0.091
2	0.500	0.355	0.092	0.001	0.479 ↓	0.424	0.048 ↓
3	0.480	0.424	0.048	0.001	0.328 ↓	0.443	0.033 ↓
4	0.290	0.443	0.033	0.001	0.253 ↓	0.404	0.025 ↓
5	0.250	0.404	0.025	**0.020** ↑	**0.312** ↑	0.356	**0.031** ↑
6	0.320	0.356	0.031	**0.050** ↑	**0.448** ↑	0.340	**0.045** ↑
7	0.340	0.340	0.045	**0.100** ↑	**0.592** ↑	0.340	**0.059** ↑
8	0.480	0.340	0.059	**0.010** ↓	**0.409** ↓	0.397	**0.041** ↓
9	0.410	0.397	0.041	0.001 ↓	0.295 ↓	0.401	0.030 ↓

external shock

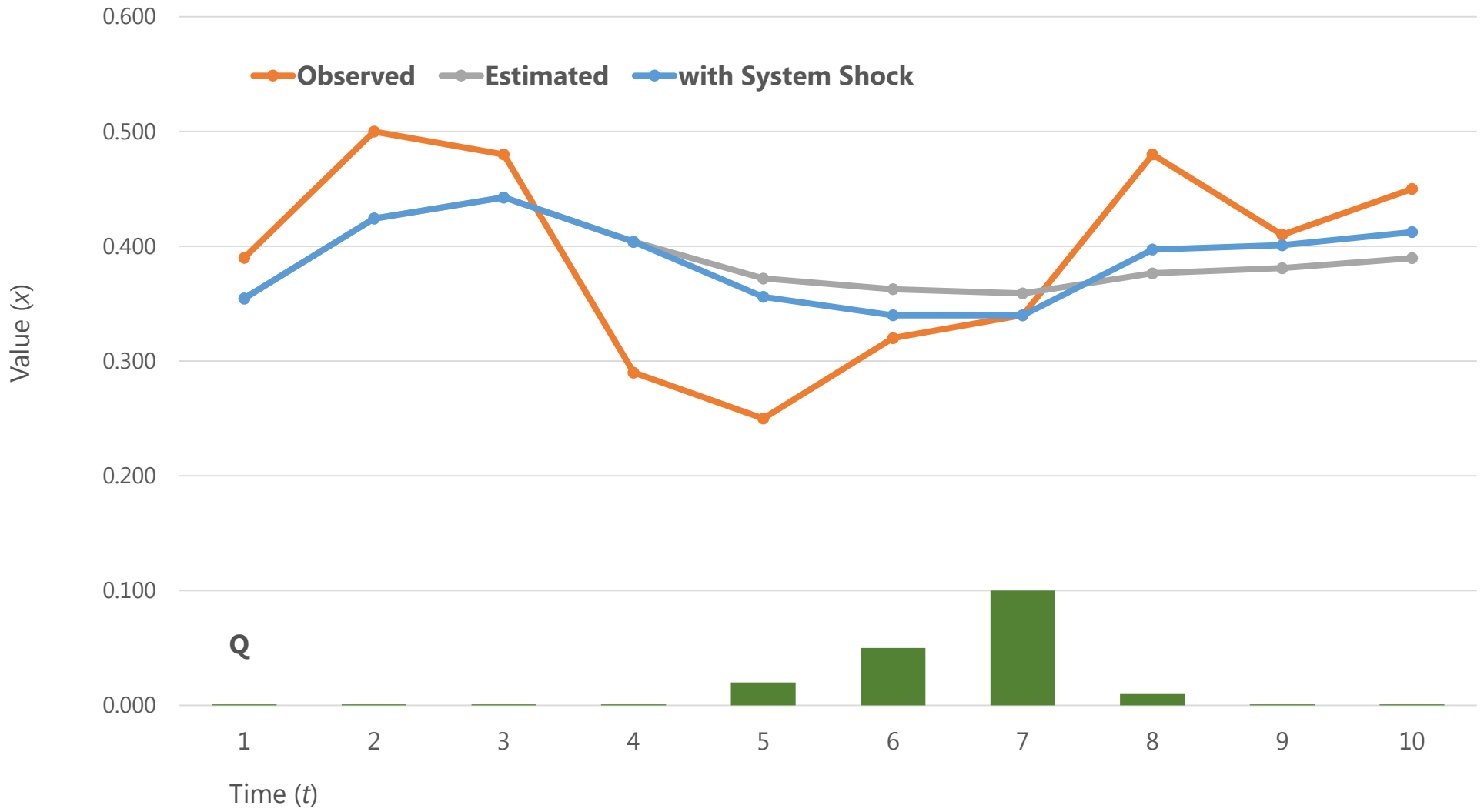
$$\tilde{P}_t = P_{t-1} + Q$$

$$K_t = \tilde{P}_t\left(\tilde{P}_t + R\right)^{-1}$$

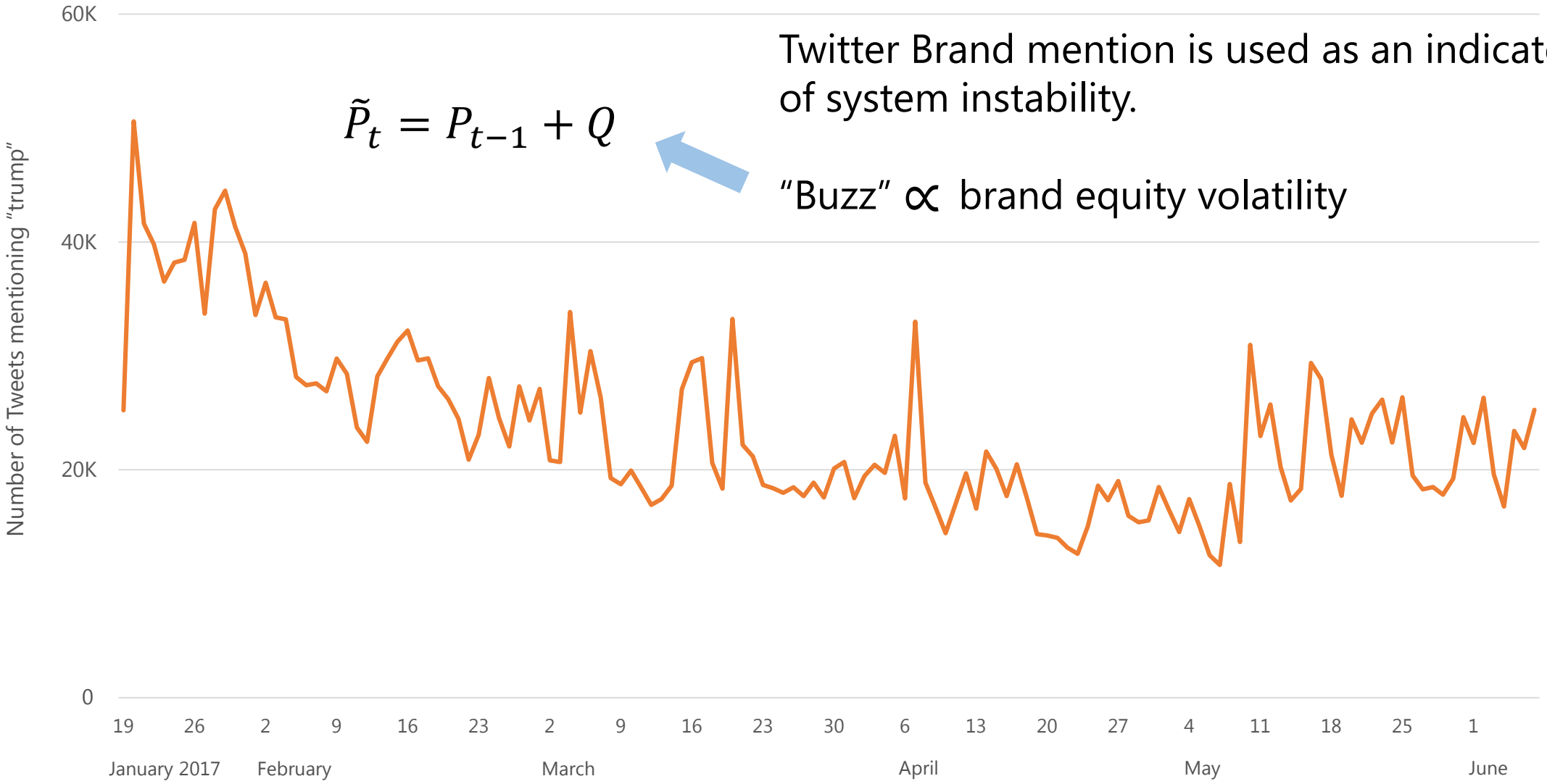
random measurement error

$$P_t = \tilde{P}_t(1 - K_t)$$

Incorporating Known System Instability



Incorporating Twitter Brand Mention in Brand Tracking



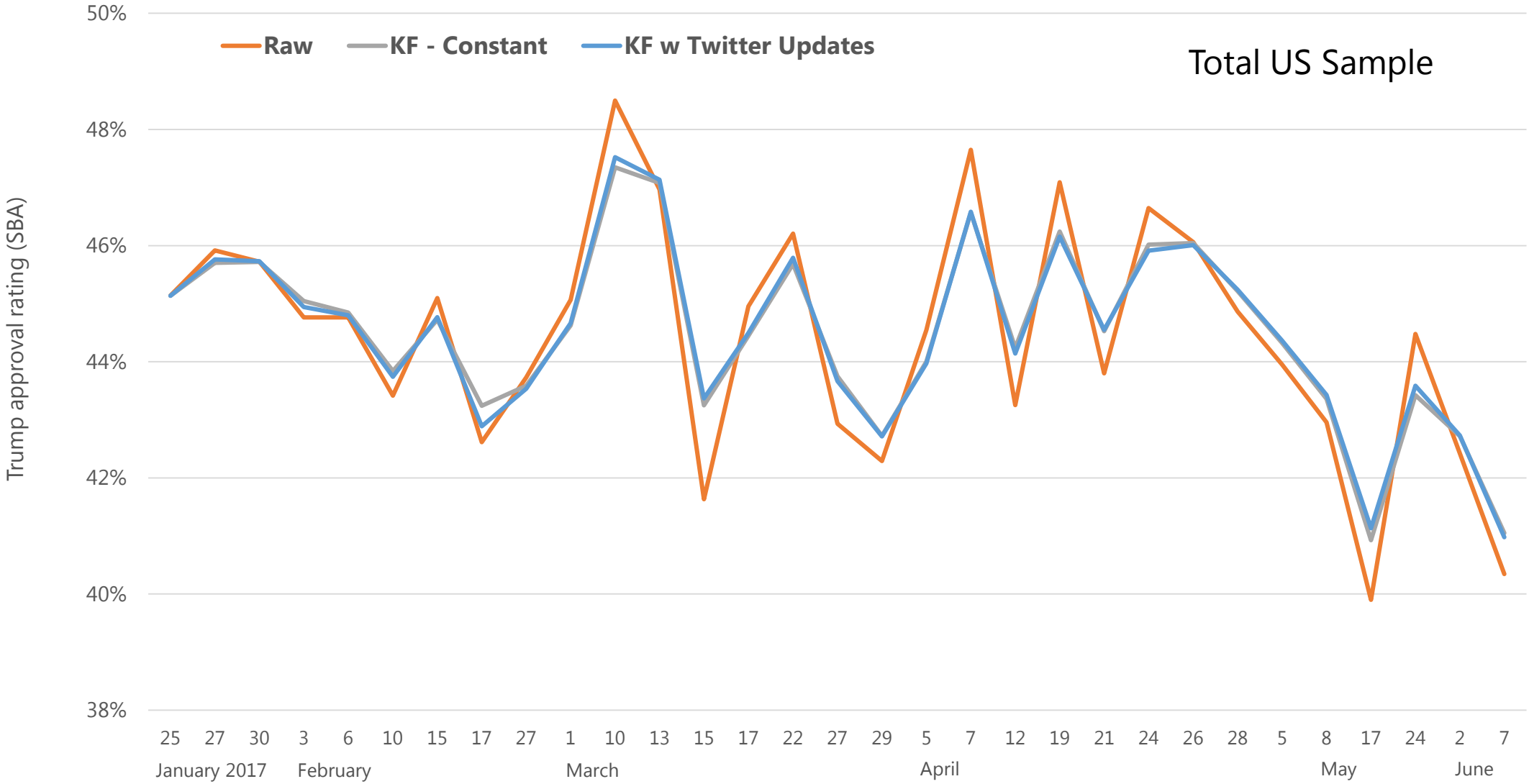
$$\tilde{P}_t = P_{t-1} + Q$$

Twitter Brand mention is used as an indicator of system instability.

"Buzz" $\propto$ brand equity volatility

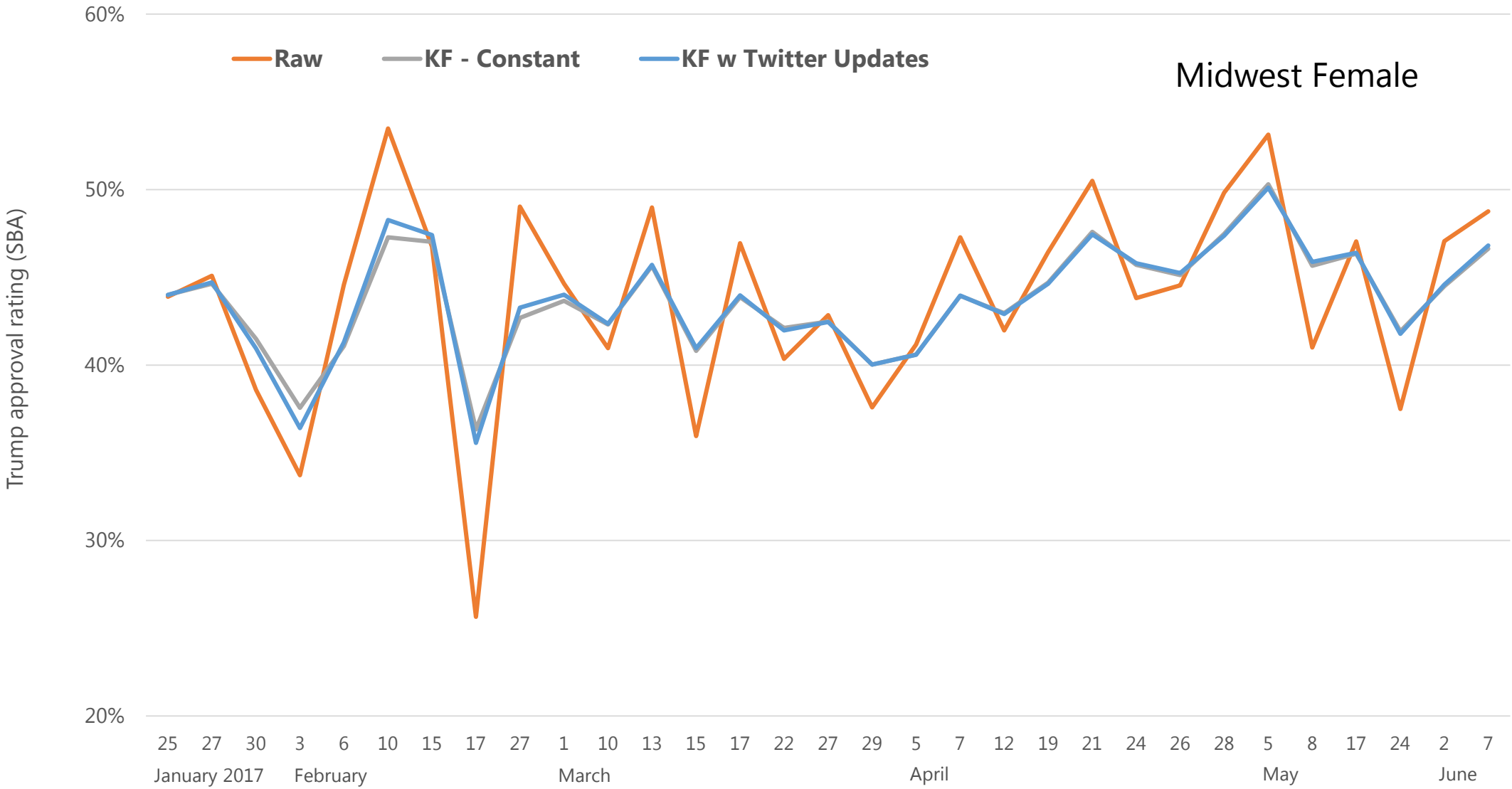
maru/matchbox

Tracking the "Trump" Brand

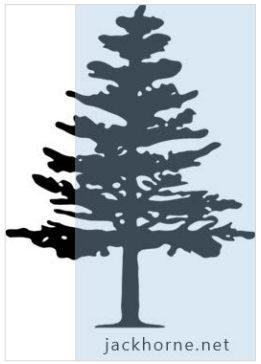


maru/matchbox

“Trump” Brand Tracking in Subpopulation



maru/matchbox



THANK YOU!

Jack Horne <jack@jackhorne.net>
Jane Tang <jane.tang@marumatchbox.com>

maru / matchbox