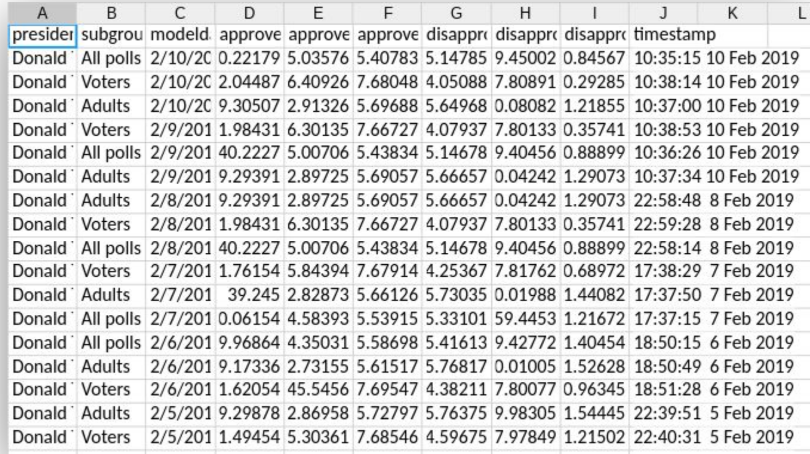
Data Visualization

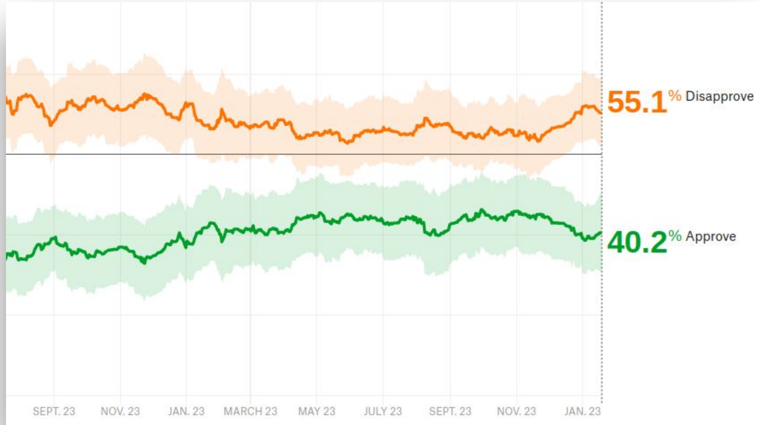
Weiai Xu (Wayne), PhD
Assistant Professor
Department of Communication, UMass-Amherst
Email: weiaixu@umass.edu
curiositybits.cc

What is *data visualization*?

Display statistics or any abstract information using graphs.

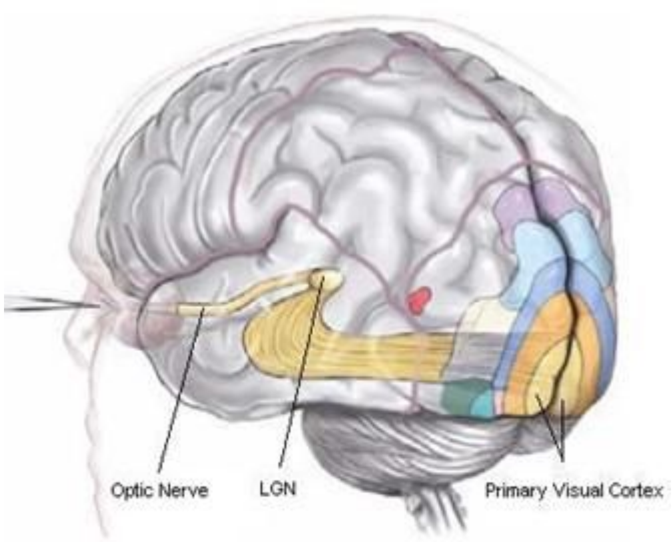


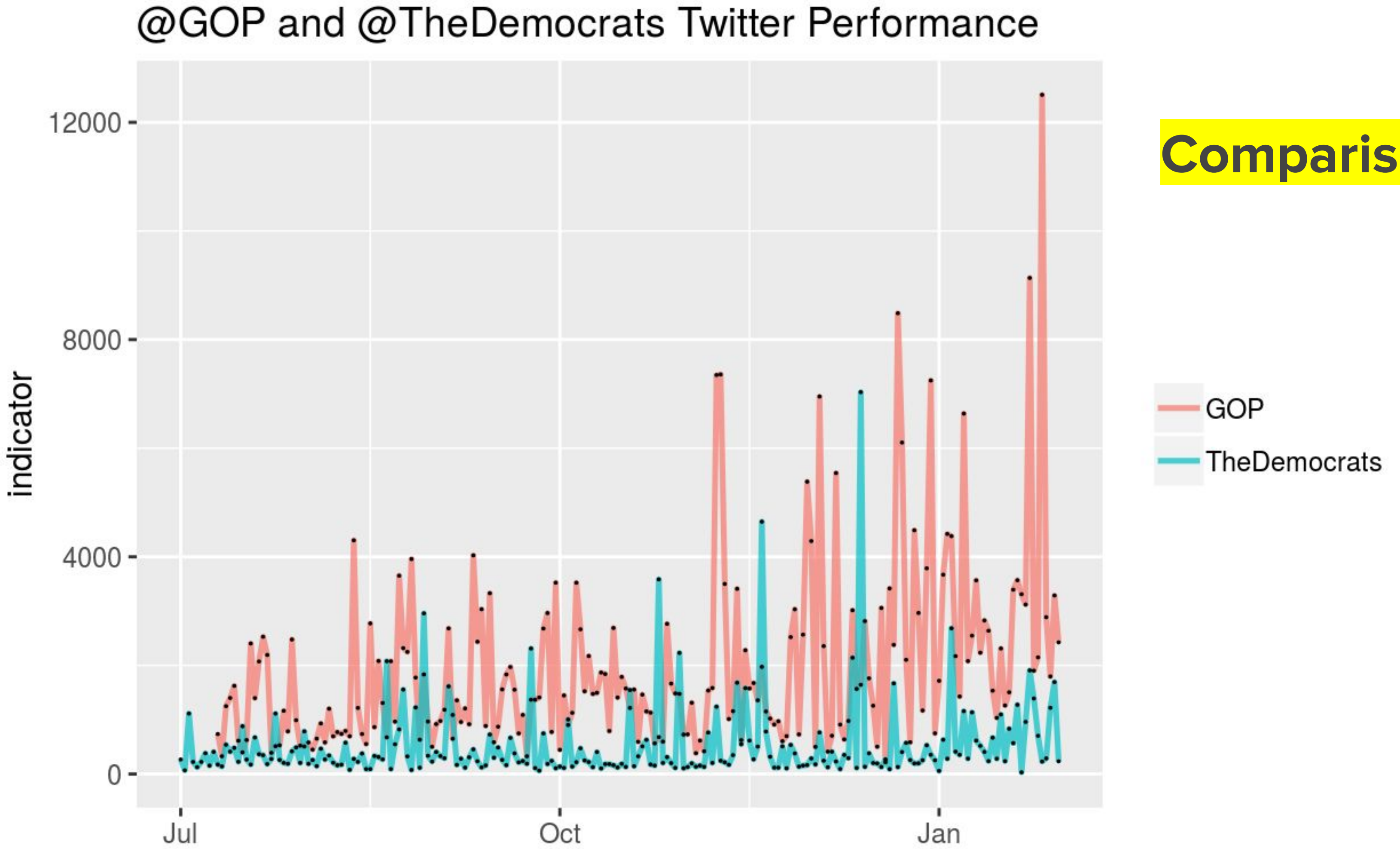
A	B	C	D	E	F	G	H	I	J	K	L
presider	subgrou	modeld.	approve	approve	approve	disappr	disappr	disappr	timestamp		
Donald	All polls	2/10/2C	0.22179	5.03576	5.40783	5.14785	9.45002	0.84567	10:35:15	10 Feb 2019	
Donald	Voters	2/10/2C	2.04487	6.40926	7.68048	4.05088	7.80891	0.29285	10:38:14	10 Feb 2019	
Donald	Adults	2/10/2C	9.30507	2.91326	5.69688	5.64968	0.08082	1.21855	10:37:00	10 Feb 2019	
Donald	Voters	2/9/201	1.98431	6.30135	7.66727	4.07937	7.80133	0.35741	10:38:53	10 Feb 2019	
Donald	All polls	2/9/201	40.2227	5.00706	5.43834	5.14678	9.40456	0.88899	10:36:26	10 Feb 2019	
Donald	Adults	2/9/201	9.29391	2.89725	5.69057	5.66657	0.04242	1.29073	10:37:34	10 Feb 2019	
Donald	Adults	2/8/201	9.29391	2.89725	5.69057	5.66657	0.04242	1.29073	22:58:48	8 Feb 2019	
Donald	Voters	2/8/201	1.98431	6.30135	7.66727	4.07937	7.80133	0.35741	22:59:28	8 Feb 2019	
Donald	All polls	2/8/201	40.2227	5.00706	5.43834	5.14678	9.40456	0.88899	22:58:14	8 Feb 2019	
Donald	Voters	2/7/201	1.76154	5.84394	7.67914	4.25367	7.81762	0.68972	17:38:29	7 Feb 2019	
Donald	Adults	2/7/201	39.245	2.82873	5.66126	5.73035	0.01988	1.44082	17:37:50	7 Feb 2019	
Donald	All polls	2/7/201	0.06154	4.58393	5.53915	5.33101	59.4453	1.21672	17:37:15	7 Feb 2019	
Donald	All polls	2/6/201	9.96864	4.35031	5.58698	5.41613	9.42772	1.40454	18:50:15	6 Feb 2019	
Donald	Adults	2/6/201	9.17336	2.73155	5.61517	5.76817	0.01005	1.52628	18:50:49	6 Feb 2019	
Donald	Voters	2/6/201	1.62054	45.5456	7.69547	4.38211	7.80077	0.96345	18:51:28	6 Feb 2019	
Donald	Adults	2/5/201	9.29878	2.86958	5.72797	5.76375	9.98305	1.54445	22:39:51	5 Feb 2019	
Donald	Voters	2/5/201	1.49454	5.30361	7.68546	4.59675	7.97849	1.21502	22:40:31	5 Feb 2019	



What is the point of visualizing data?

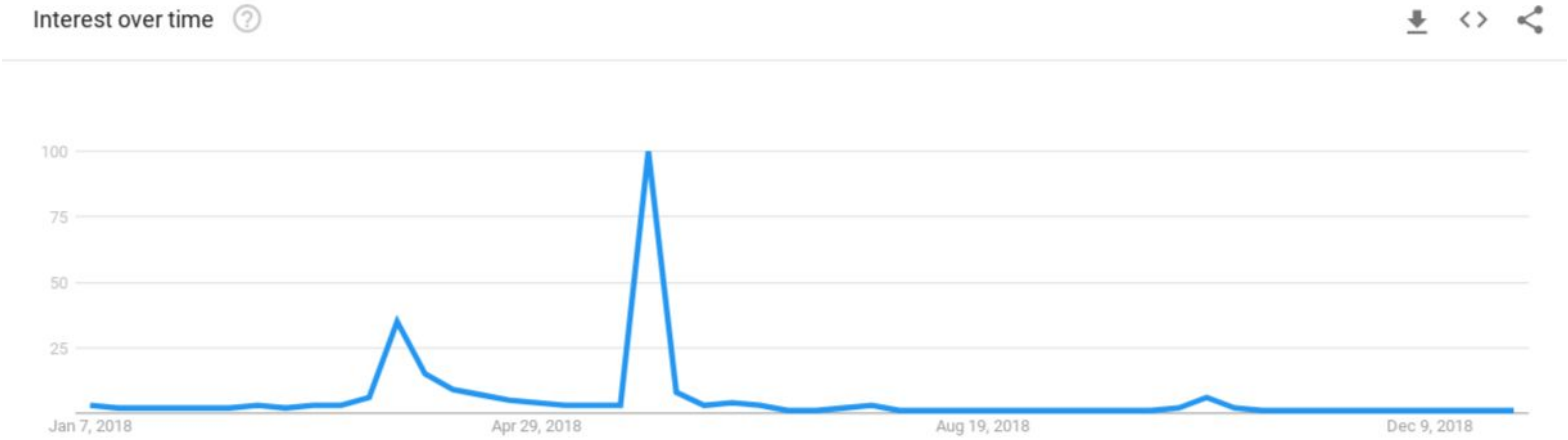
- *Seeing* is “better” than *thinking;*
- A great tool of storytelling and data exploration




@GOP and @TheDemocrats Twitter Performance
indicator
12000
8000
4000
0
Jul
Oct
Jan
GOP
TheDemocrats

Comparison

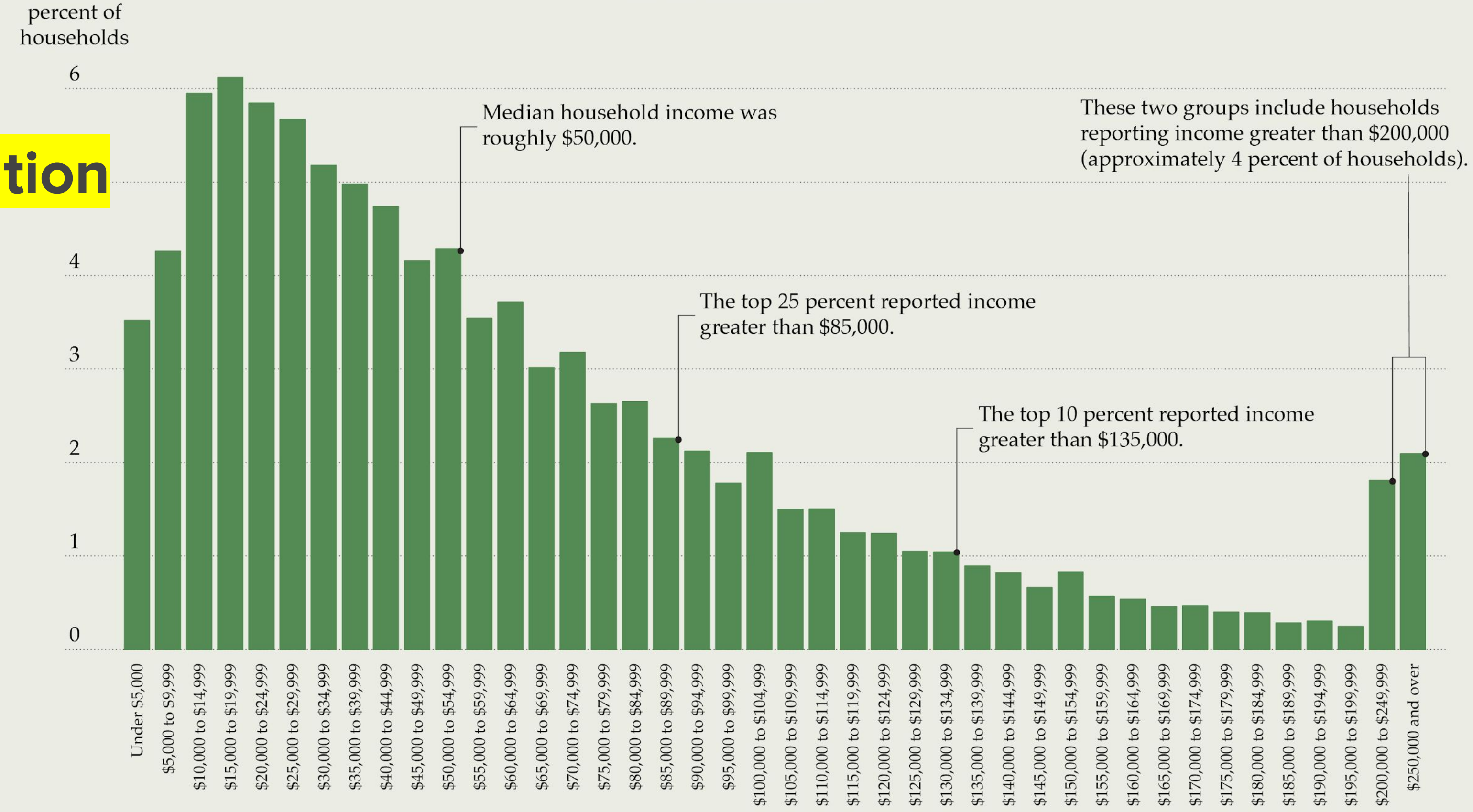
Trend


Interest over time
100
75
50
25
Jan 7, 2018
Apr 29, 2018
Aug 19, 2018
Dec 9, 2018

Distribution

Distribution of annual household income in the United States

2010 estimate



Median household income was roughly $50,000.

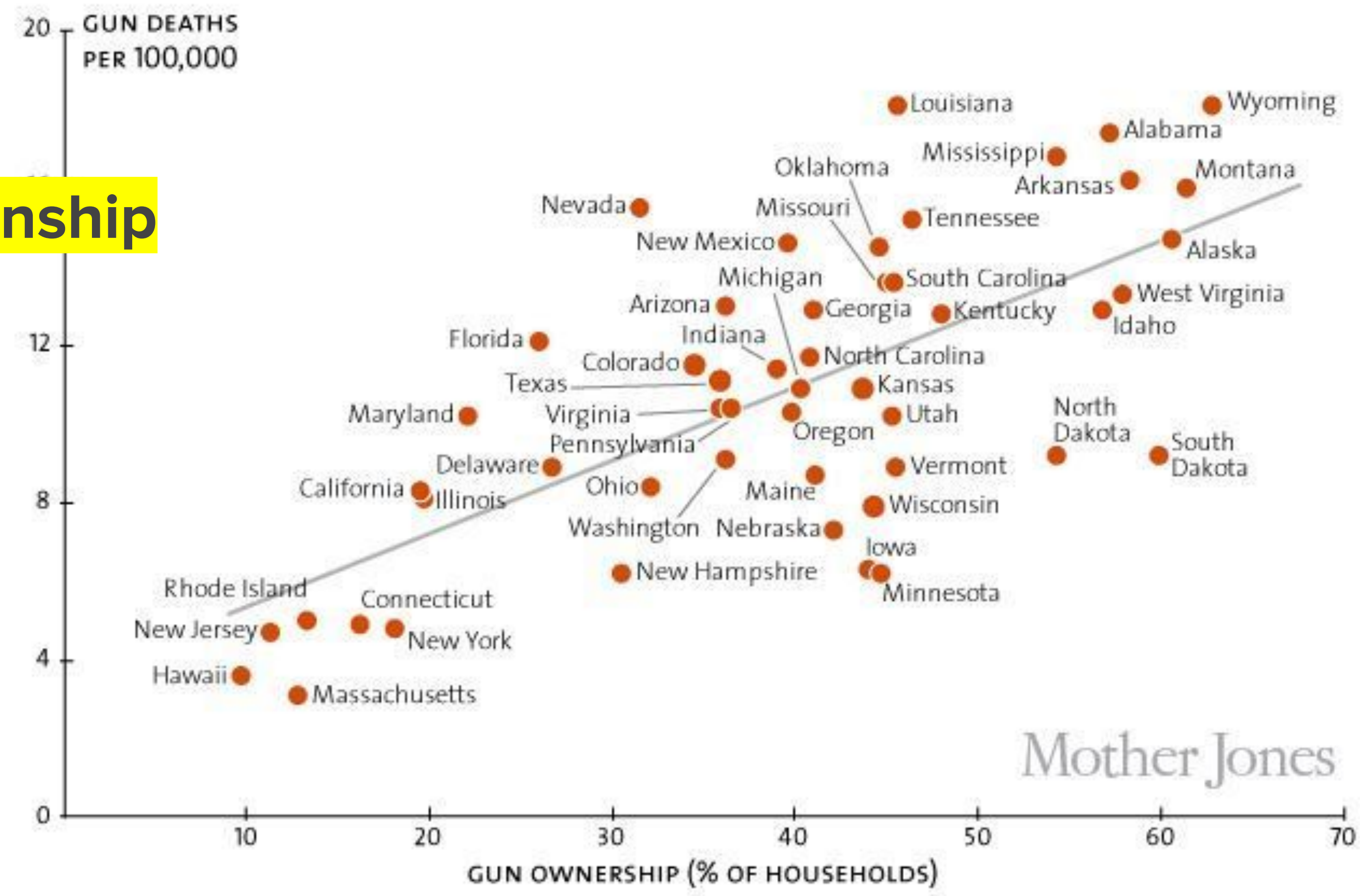
The top 25 percent reported income greater than $85,000.

The top 10 percent reported income greater than $135,000.

These two groups include households reporting income greater than $200,000 (approximately 4 percent of households).

Source: U.S. Census Bureau, Current Population Survey, 2011 Annual Social and Economic Supplement

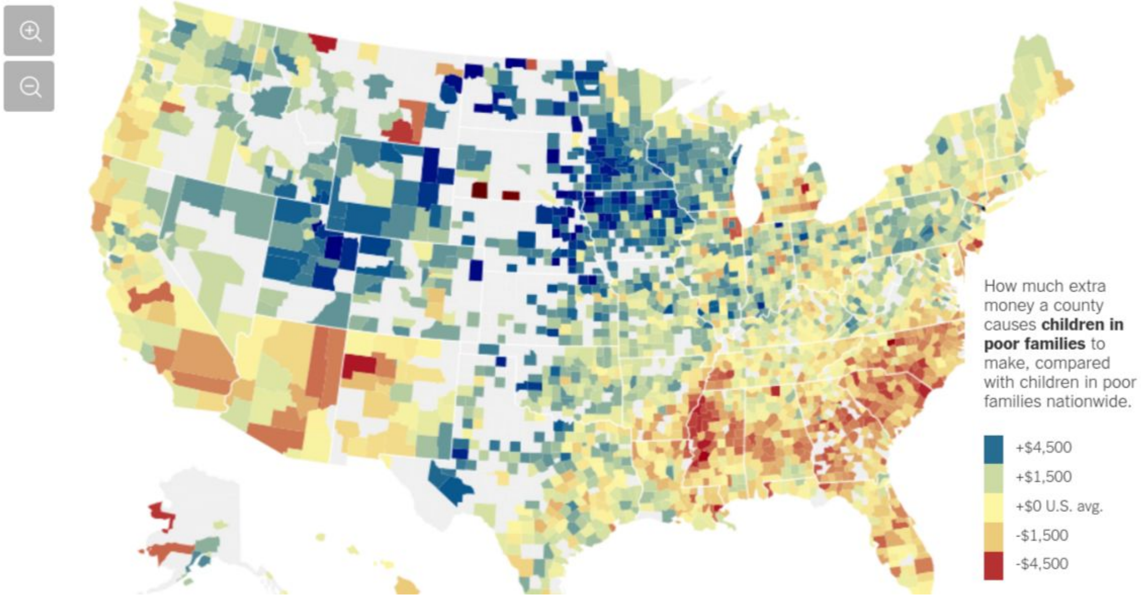
Relationship

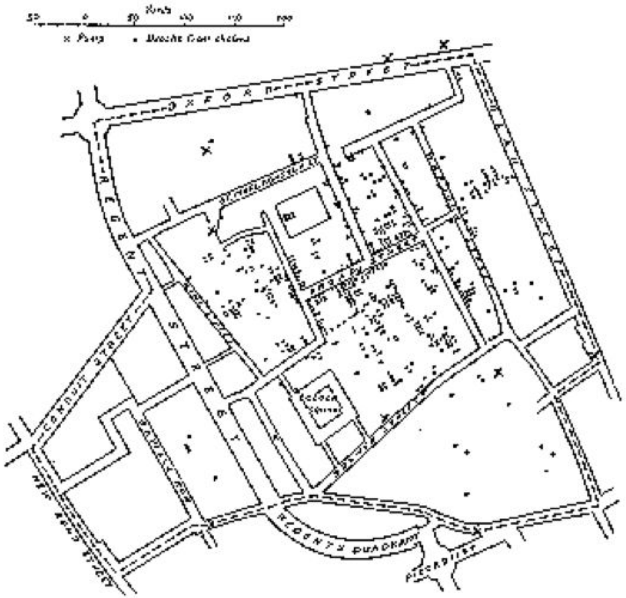


Gun ownership vs. gun deaths, by state

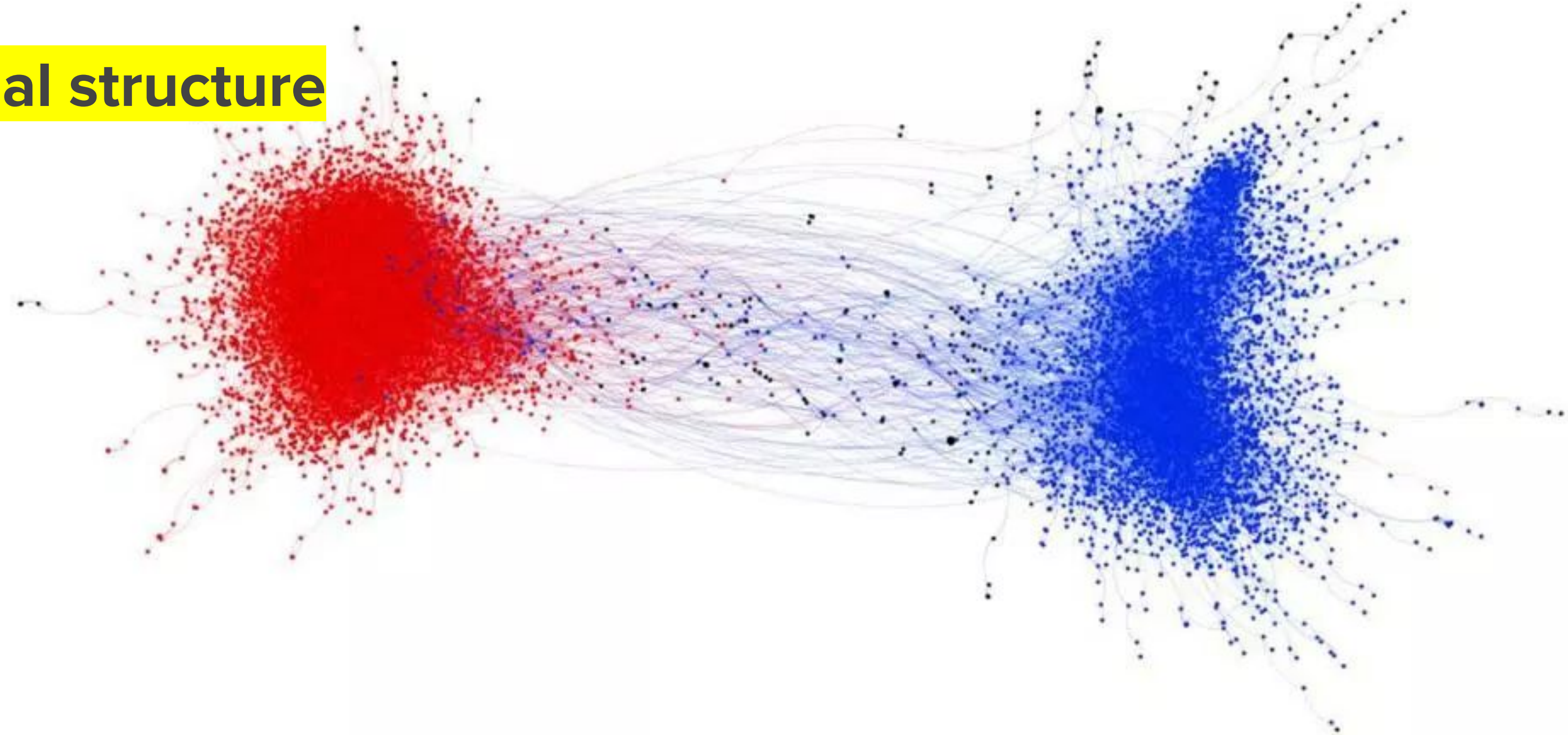
Sources: Pediatrics, Centers for Disease Control and Prevention

Locations

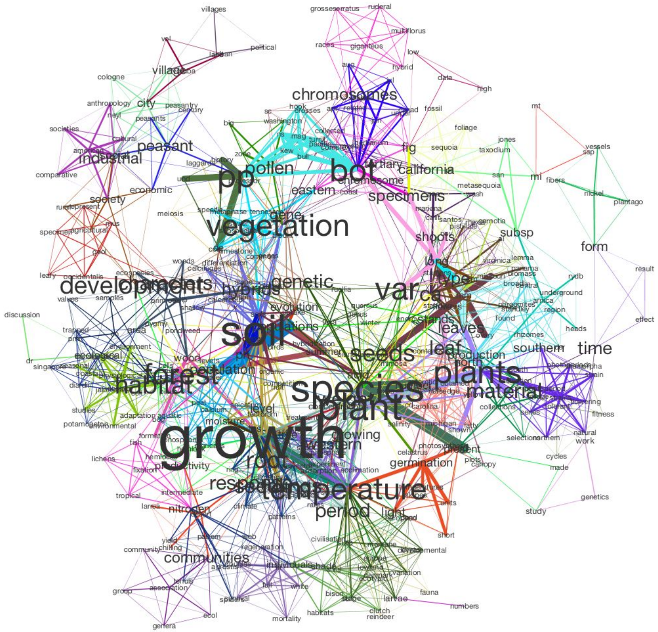

How much extra money a county causes **children in poor families** to make, compared with children in poor families nationwide.
+$4,500
+$1,500
+$0 U.S. avg.
-$1,500
-$4,500


Yards
50
0
50
100
150
200
× Pump
• Deaths from cholera
OXFORD STREET
REGENT STREET
CONDUIT STREET
REGENT'S QUADRANT
PICCADILLY

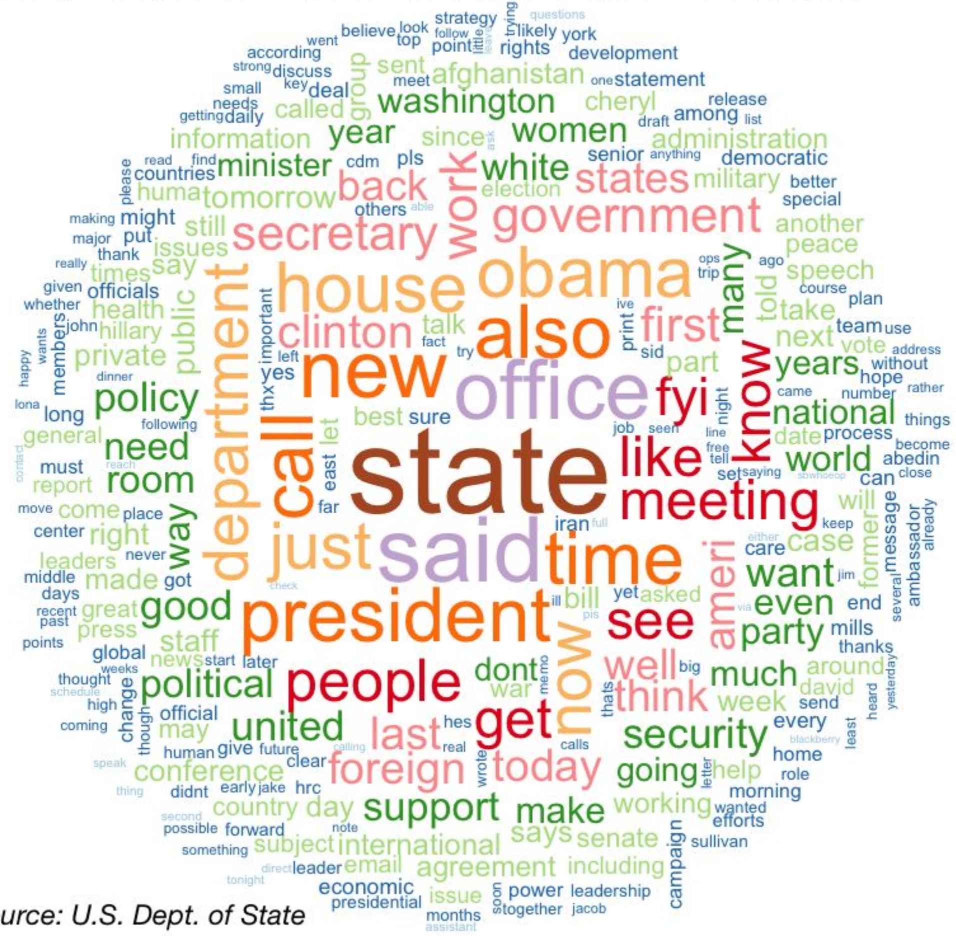
Social structure



Text



Word Cloud: *Most common words in email bodies*

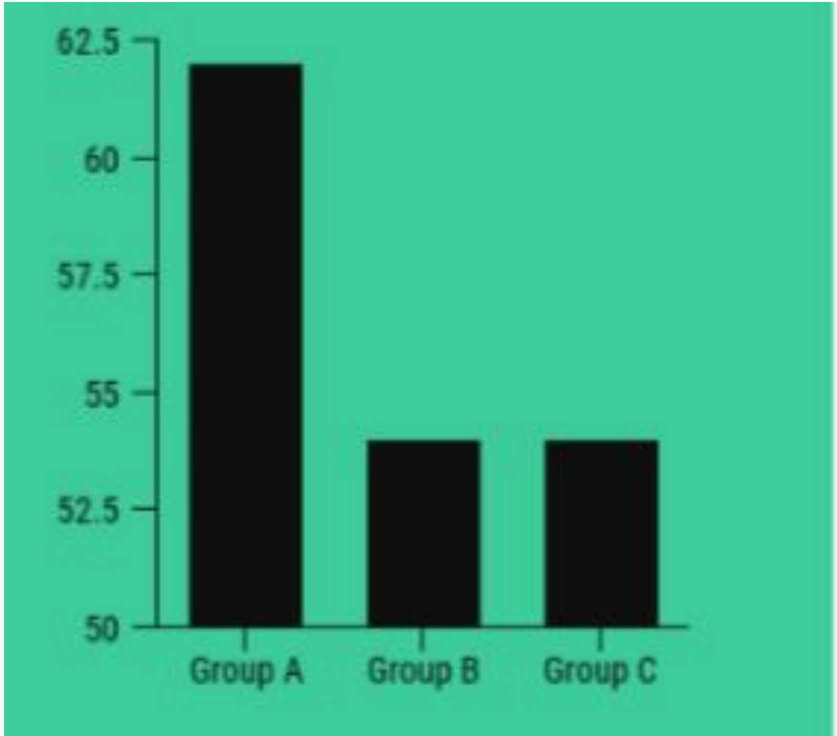


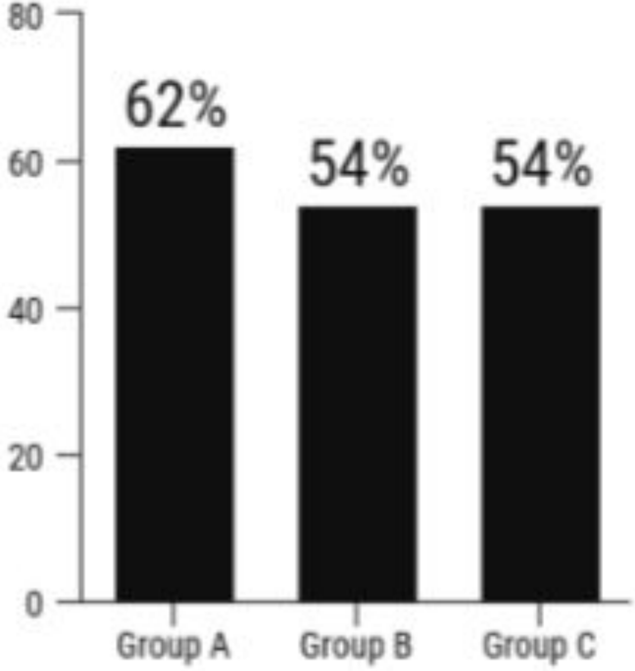
Source: U.S. Dept. of State

Analysis: Eugene Kwak & Prasant Sudhakaran, NYU Stern

Caveats in visualization

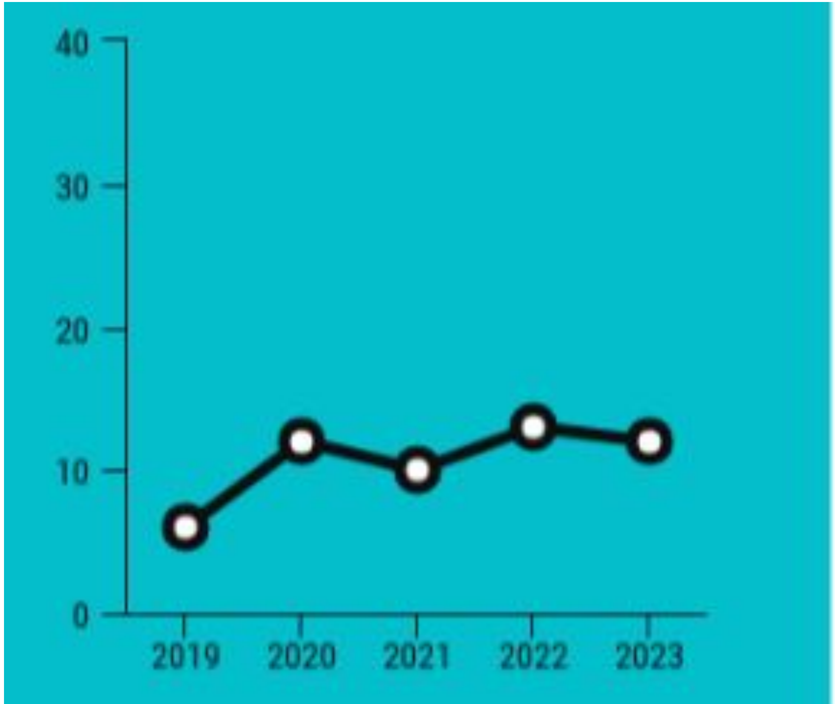
Examples of bad visualizations

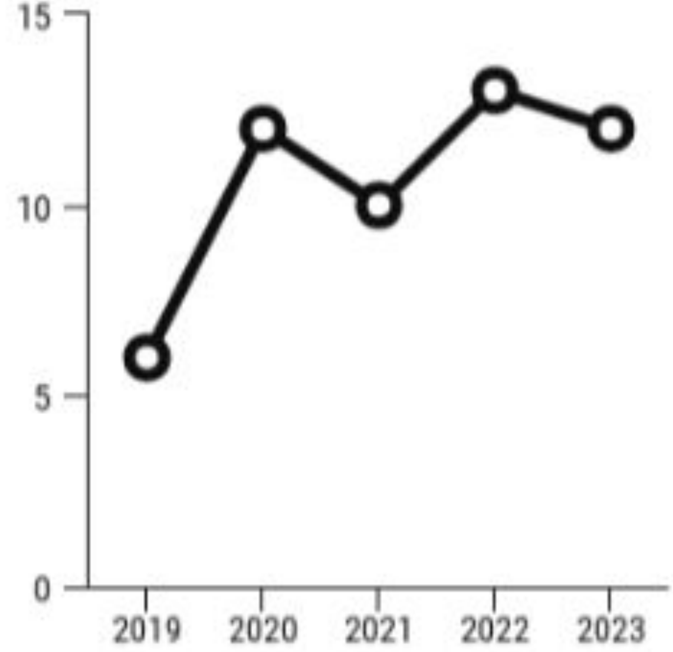


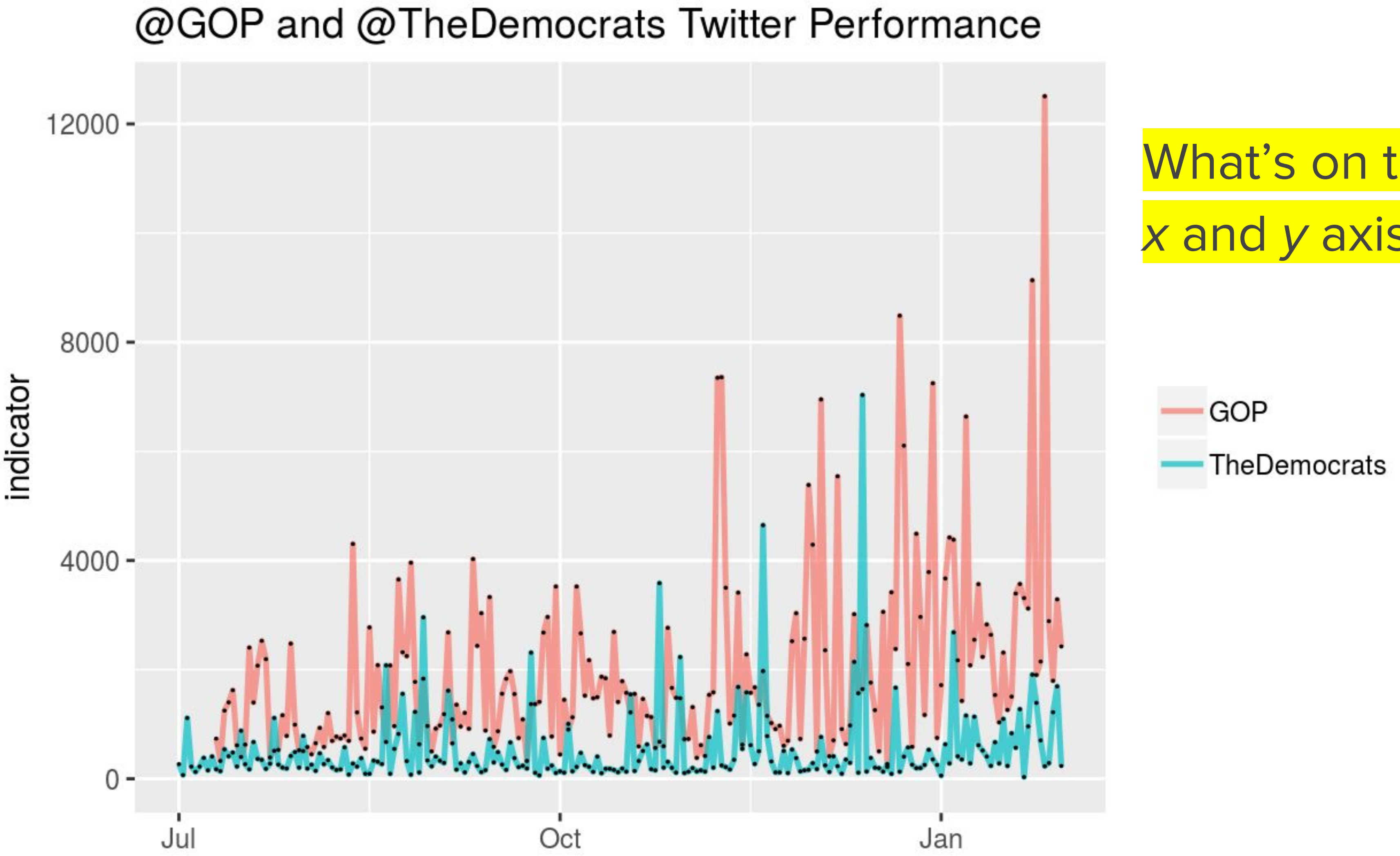


Caveats in visualization

Examples of bad visualizations







What's on the *x* and *y* axis?

Practice: Visualizing virality of tweets

Workflow

Install and load necessary libraries

library()

Download data

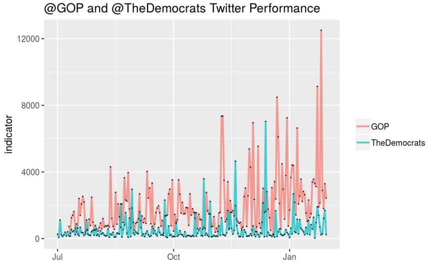
read.csv()

Standardize timestamps

ymd_hms(),
with_tz(),
as.Date()

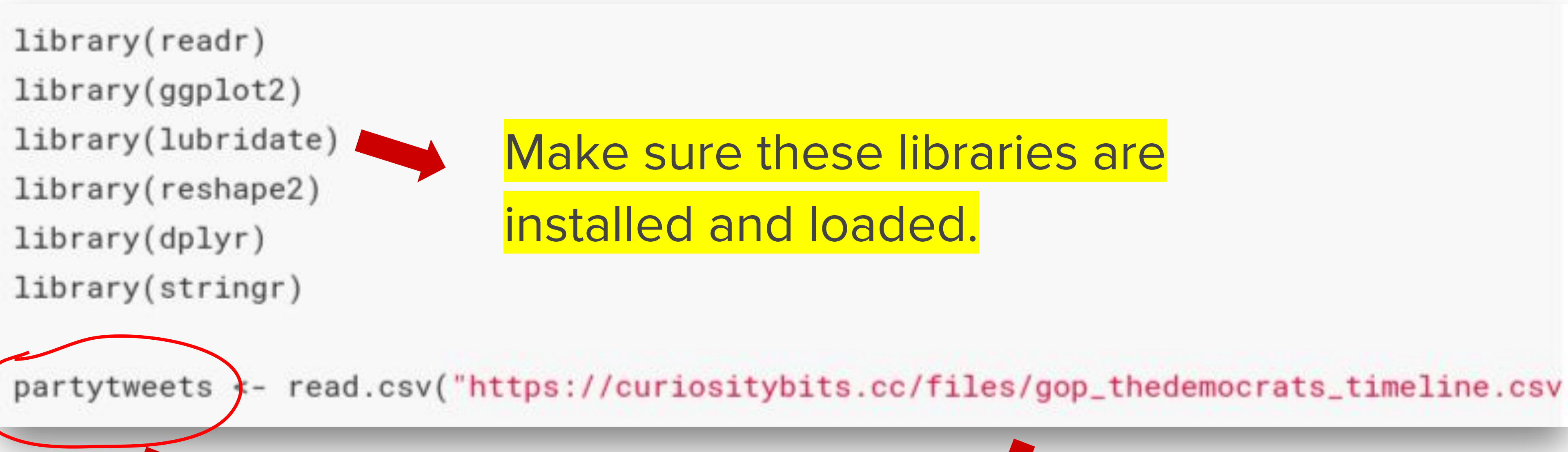
Summarise data by date and Twitter handles

group_by(),
summarise()



Visualize!

ggplot()

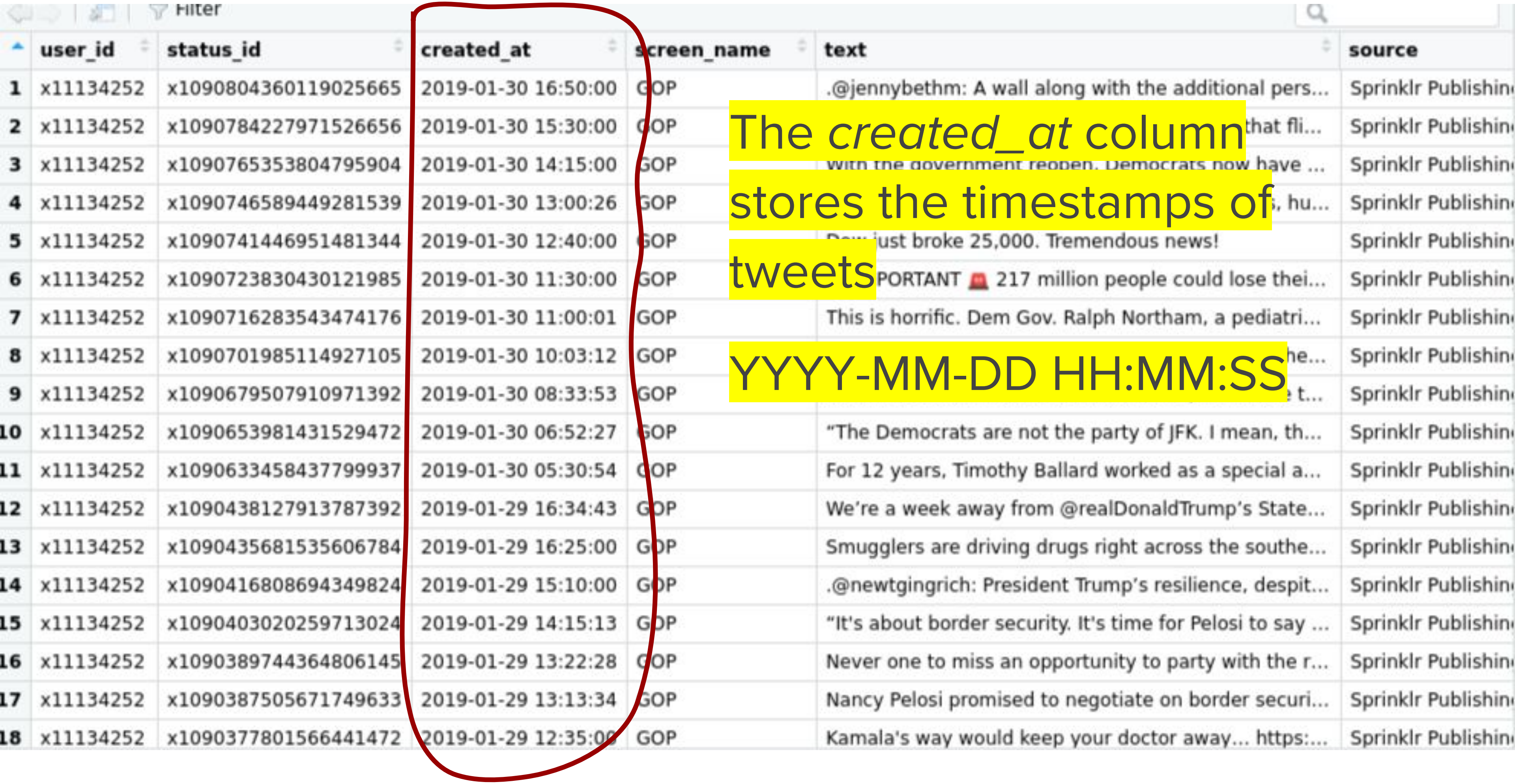


```
library(readr)
library(ggplot2)
library(lubridate)
library(reshape2)
library(dplyr)
library(stringr)

partytweets <- read.csv("https://curiositybits.cc/files/gop_thedemocrats_timeline.csv
```

Make sure these libraries are installed and loaded.

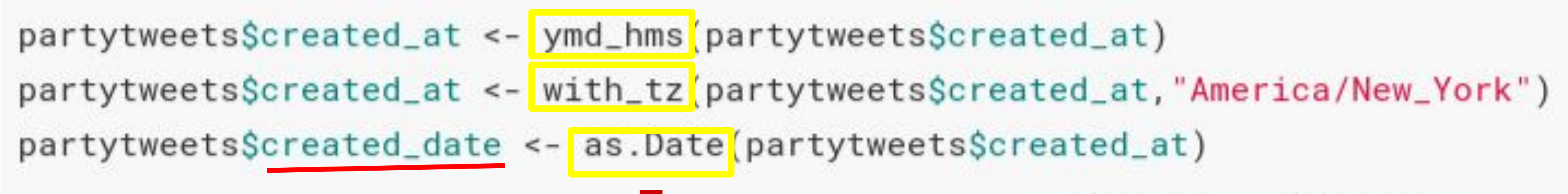
Download the .csv file from my cloud server



	user_id	status_id	created_at	screen_name	text	source
1	x11134252	x1090804360119025665	2019-01-30 16:50:00	GOP	.@jennybethm: A wall along with the additional pers...	Sprinklr Publishin
2	x11134252	x1090784227971526656	2019-01-30 15:30:00	GOP	that fli...	Sprinklr Publishin
3	x11134252	x1090765353804795904	2019-01-30 14:15:00	GOP	with the government reopen. Democrats now have ...	Sprinklr Publishin
4	x11134252	x1090746589449281539	2019-01-30 13:00:26	GOP	, hu...	Sprinklr Publishin
5	x11134252	x1090741446951481344	2019-01-30 12:40:00	GOP	Dow just broke 25,000. Tremendous news!	Sprinklr Publishin
6	x11134252	x1090723830430121985	2019-01-30 11:30:00	GOP	PORTANT 🚨 217 million people could lose thei...	Sprinklr Publishin
7	x11134252	x1090716283543474176	2019-01-30 11:00:01	GOP	This is horrific. Dem Gov. Ralph Northam, a pediatri...	Sprinklr Publishin
8	x11134252	x1090701985114927105	2019-01-30 10:03:12	GOP	he...	Sprinklr Publishin
9	x11134252	x1090679507910971392	2019-01-30 08:33:53	GOP	t...	Sprinklr Publishin
10	x11134252	x1090653981431529472	2019-01-30 06:52:27	GOP	"The Democrats are not the party of JFK. I mean, th...	Sprinklr Publishin
11	x11134252	x1090633458437799937	2019-01-30 05:30:54	GOP	For 12 years, Timothy Ballard worked as a special a...	Sprinklr Publishin
12	x11134252	x1090438127913787392	2019-01-29 16:34:43	GOP	We're a week away from @realDonaldTrump's State...	Sprinklr Publishin
13	x11134252	x1090435681535606784	2019-01-29 16:25:00	GOP	Smugglers are driving drugs right across the southe...	Sprinklr Publishin
14	x11134252	x1090416808694349824	2019-01-29 15:10:00	GOP	.@newtgingrich: President Trump's resilience, despit...	Sprinklr Publishin
15	x11134252	x1090403020259713024	2019-01-29 14:15:13	GOP	"It's about border security. It's time for Pelosi to say ...	Sprinklr Publishin
16	x11134252	x1090389744364806145	2019-01-29 13:22:28	GOP	Never one to miss an opportunity to party with the r...	Sprinklr Publishin
17	x11134252	x1090387505671749633	2019-01-29 13:13:34	GOP	Nancy Pelosi promised to negotiate on border securi...	Sprinklr Publishin
18	x11134252	x1090377801566441472	2019-01-29 12:35:00	GOP	Kamala's way would keep your doctor away... https:...	Sprinklr Publishin

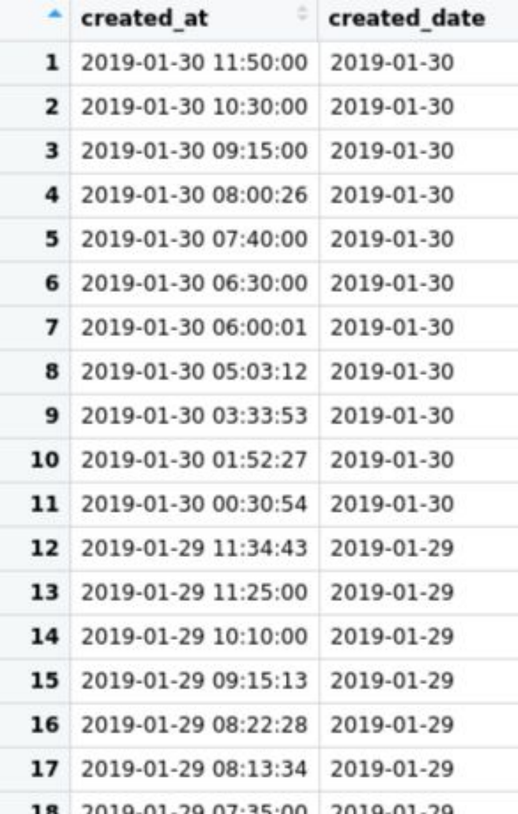
The *created_at* column stores the timestamps of tweets

YYYY-MM-DD HH:MM:SS

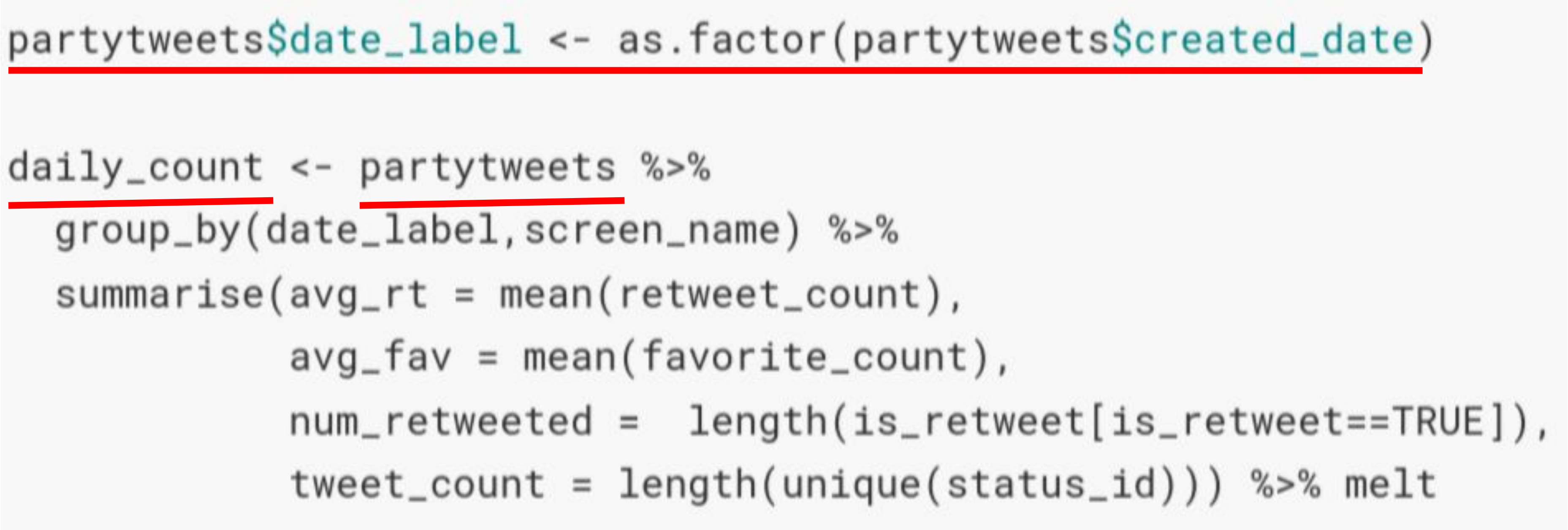


```
partytweets$created_at <- ymd_hms(partytweets$created_at)
partytweets$created_at <- with_tz(partytweets$created_at,"America/New_York")
partytweets$created_date <- as.Date(partytweets$created_at)
```

- Standardize timestamps based on the *YYYY-MM-DD HH:MM:SS* format;
- Convert to the same time zone
- Extract dates and put the dates in a new column named *created_date*.



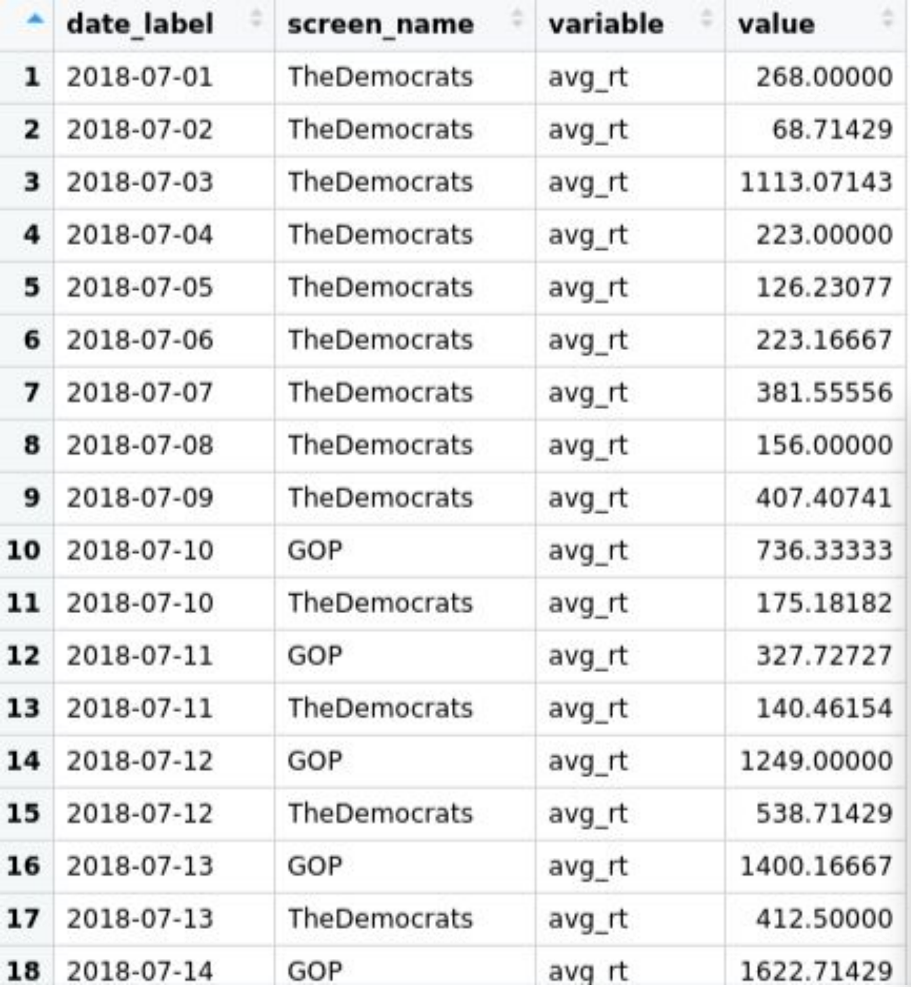
	created_at	created_date
1	2019-01-30 11:50:00	2019-01-30
2	2019-01-30 10:30:00	2019-01-30
3	2019-01-30 09:15:00	2019-01-30
4	2019-01-30 08:00:26	2019-01-30
5	2019-01-30 07:40:00	2019-01-30
6	2019-01-30 06:30:00	2019-01-30
7	2019-01-30 06:00:01	2019-01-30
8	2019-01-30 05:03:12	2019-01-30
9	2019-01-30 03:33:53	2019-01-30
10	2019-01-30 01:52:27	2019-01-30
11	2019-01-30 00:30:54	2019-01-30
12	2019-01-29 11:34:43	2019-01-29
13	2019-01-29 11:25:00	2019-01-29
14	2019-01-29 10:10:00	2019-01-29
15	2019-01-29 09:15:13	2019-01-29
16	2019-01-29 08:22:28	2019-01-29
17	2019-01-29 08:13:34	2019-01-29
18	2019-01-29 07:35:00	2019-01-29



```
partytweets$date_label <- as.factor(partytweets$created_date)

daily_count <- partytweets %>%
  group_by(date_label,screen_name) %>%
  summarise(avg_rt = mean(retweet_count),
            avg_fav = mean(favorite_count),
            num_retweeted =  length(is_retweet[is_retweet==TRUE]),
            tweet_count = length(unique(status_id))) %>% melt
```

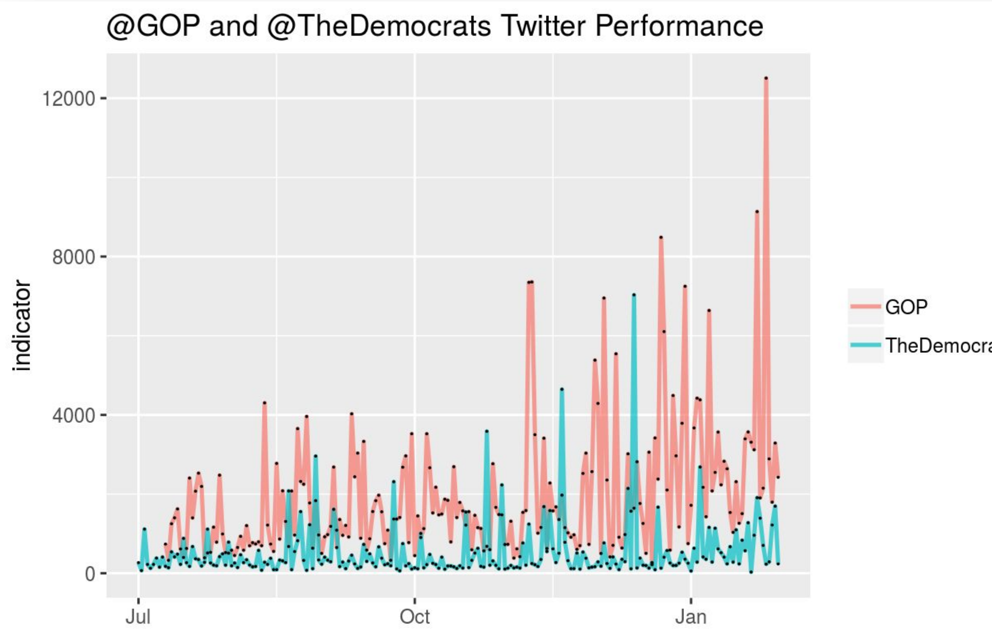
Daily_count is created from *partytweets*
based on a summary of data by *screen_name*
and *date_label.*

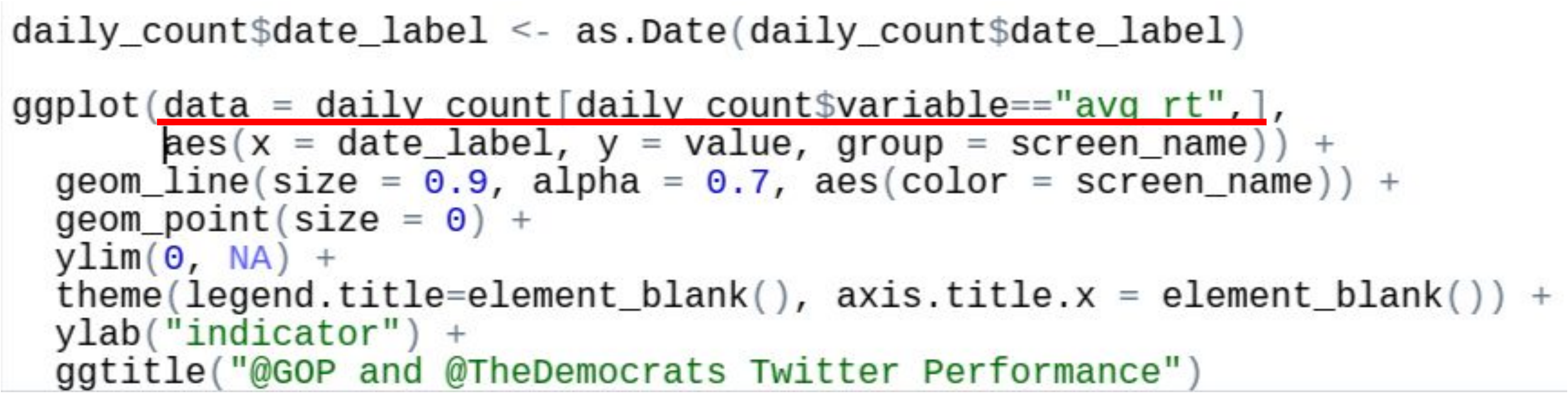


	date_label	screen_name	variable	value
1	2018-07-01	TheDemocrats	avg_rt	268.00000
2	2018-07-02	TheDemocrats	avg_rt	68.71429
3	2018-07-03	TheDemocrats	avg_rt	1113.07143
4	2018-07-04	TheDemocrats	avg_rt	223.00000
5	2018-07-05	TheDemocrats	avg_rt	126.23077
6	2018-07-06	TheDemocrats	avg_rt	223.16667
7	2018-07-07	TheDemocrats	avg_rt	381.55556
8	2018-07-08	TheDemocrats	avg_rt	156.00000
9	2018-07-09	TheDemocrats	avg_rt	407.40741
10	2018-07-10	GOP	avg_rt	736.33333
11	2018-07-10	TheDemocrats	avg_rt	175.18182
12	2018-07-11	GOP	avg_rt	327.72727
13	2018-07-11	TheDemocrats	avg_rt	140.46154
14	2018-07-12	GOP	avg_rt	1249.00000
15	2018-07-12	TheDemocrats	avg_rt	538.71429
16	2018-07-13	GOP	avg_rt	1400.16667
17	2018-07-13	TheDemocrats	avg_rt	412.50000
18	2018-07-14	GOP	avg_rt	1622.71429

X axis == ??? column

Y axis == ??? column

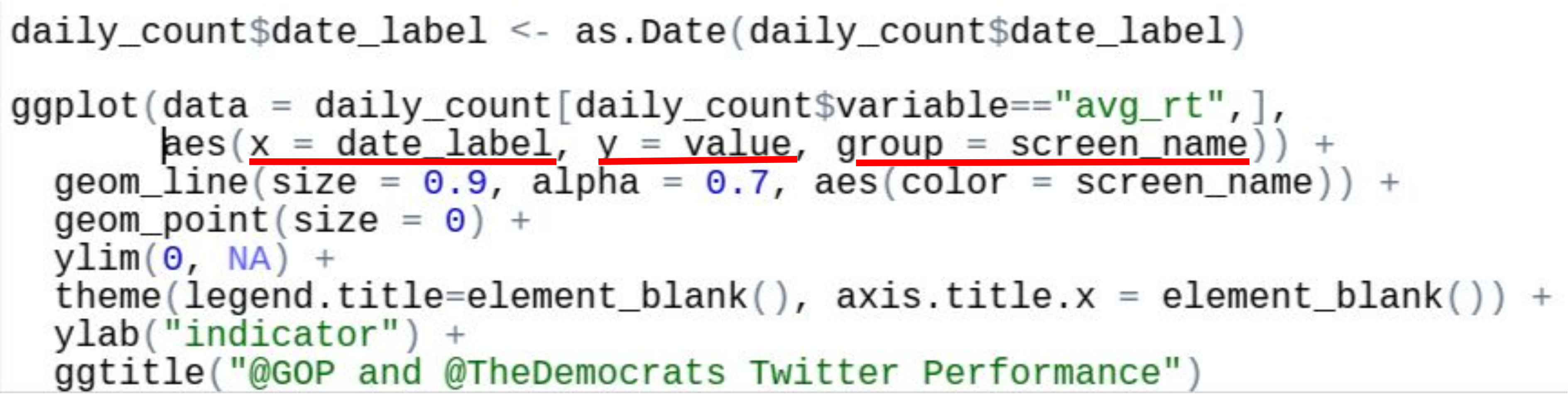




```r
daily_count$date_label <- as.Date(daily_count$date_label)

ggplot(data = daily_count[daily_count$variable=="avg_rt",],
       aes(x = date_label, y = value, group = screen_name)) +
  geom_line(size = 0.9, alpha = 0.7, aes(color = screen_name)) +
  geom_point(size = 0) +
  ylim(0, NA) +
  theme(legend.title=element_blank(), axis.title.x = element_blank()) +
  ylab("indicator") +
  ggtitle("@GOP and @TheDemocrats Twitter Performance")
```

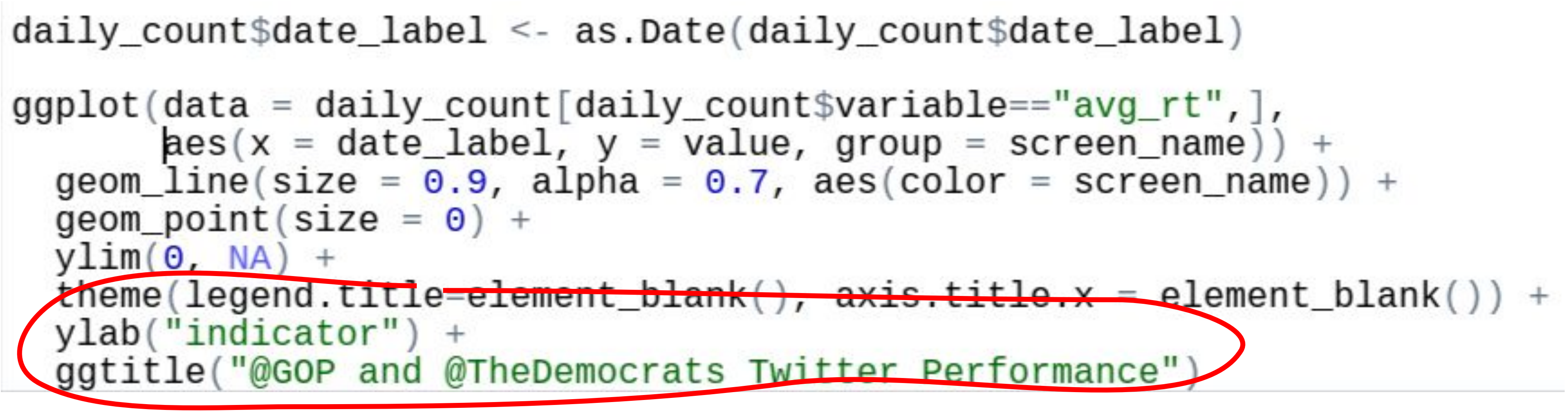
To visualize average daily retweet count, we need to select cases in *daily_count* with a value that matches “avg_rt” on the *variable* column



```r
daily_count$date_label <- as.Date(daily_count$date_label)

ggplot(data = daily_count[daily_count$variable=="avg_rt",],
       aes(x = date_label, y = value, group = screen_name)) +
  geom_line(size = 0.9, alpha = 0.7, aes(color = screen_name)) +
  geom_point(size = 0) +
  ylim(0, NA) +
  theme(legend.title=element_blank(), axis.title.x = element_blank()) +
  ylab("indicator") +
  ggtitle("@GOP and @TheDemocrats Twitter Performance")
```

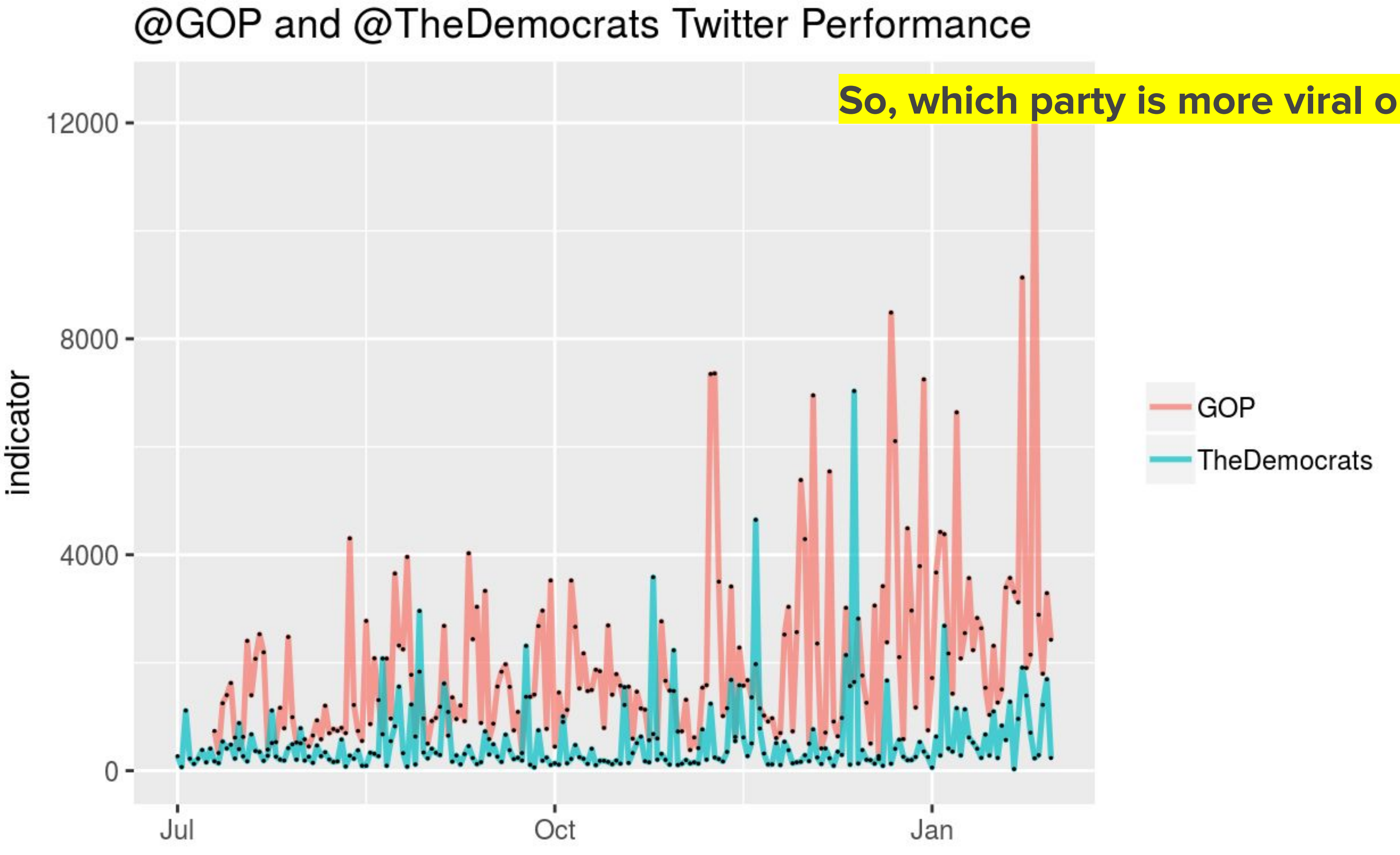
Assign values for the x and y axis and set the grouping variable



```r
daily_count$date_label <- as.Date(daily_count$date_label)

ggplot(data = daily_count[daily_count$variable=="avg_rt",],
       aes(x = date_label, y = value, group = screen_name)) +
  geom_line(size = 0.9, alpha = 0.7, aes(color = screen_name)) +
  geom_point(size = 0) +
  ylim(0, NA) +
  theme(legend.title=element_blank(), axis.title.x = element_blank()) +
  ylab("indicator") +
  ggtitle("@GOP and @TheDemocrats Twitter Performance")
```

Set labels for x and y axis



So, which party is more viral on Twitter?

Practice

- Make sure the source code can produce on your machine the same output as you see on the previous page;
- Instead of plotting daily average retweets, let's create a plot for daily average favorite count.
- Make the code work for your data

Practice script at https://curiositybits.cc/post/r_analytics8/