NETWORK SCIENCE OF ONLINE INTERACTIONS

Chapter 7 exercises + Handling large datasets/databases

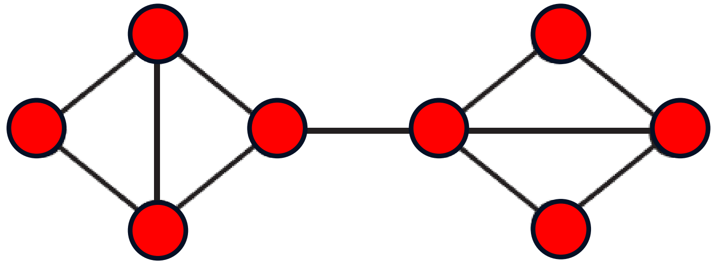
Joao Neto

14/Jun/2023

CHAPTER 7 EXERCISES

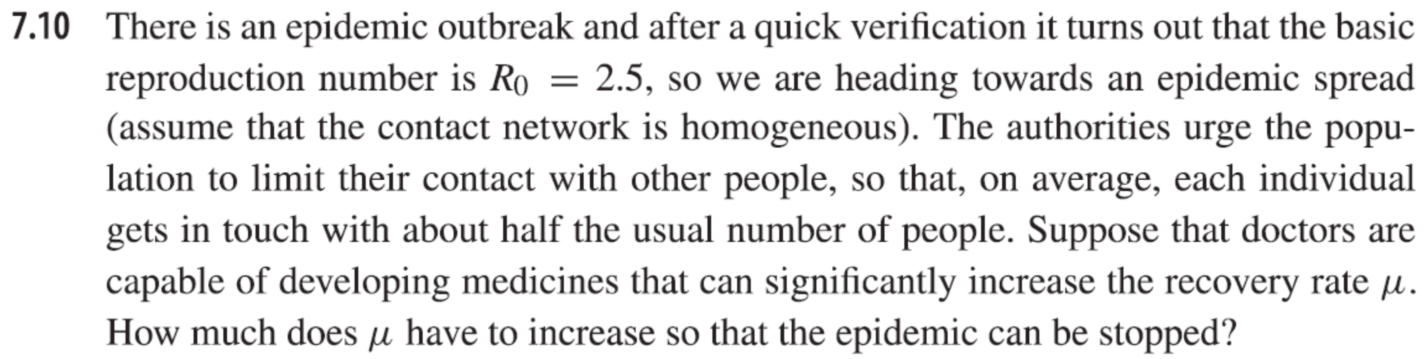
- Exercises 7.4, 7.10, 7.19

7.4 Apply the fractional threshold model to the network in Figure 7.20. The threshold is 1/2 for all nodes. Which node should we activate to obtain the largest cascade? Is the solution unique? What is the minimum number of initial influencers that are needed to activate the whole network?



- Initially, neighbours with $K = 2$ activate
- Nodes 2,3,5 and 8 have those.
- One influencer is enough to activate all

CHAPTER 7 EXERCISES



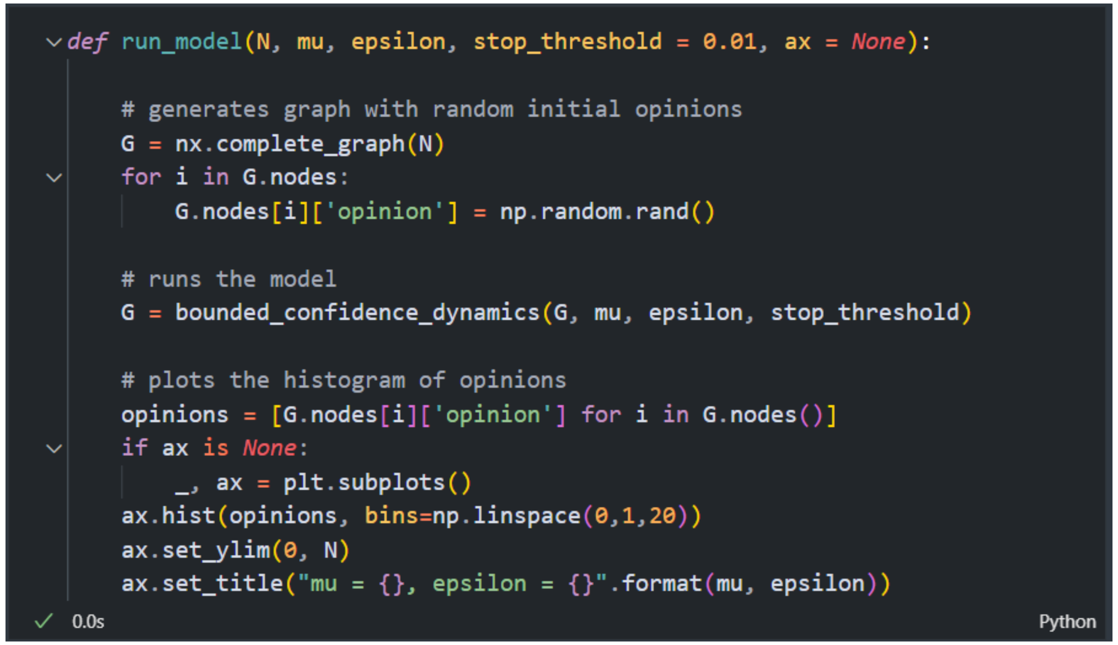
7.10 There is an epidemic outbreak and after a quick verification it turns out that the basic reproduction number is $R_0 = 2.5$, so we are heading towards an epidemic spread (assume that the contact network is homogeneous). The authorities urge the population to limit their contact with other people, so that, on average, each individual gets in touch with about half the usual number of people. Suppose that doctors are capable of developing medicines that can significantly increase the recovery rate μ. How much does μ have to increase so that the epidemic can be stopped?

$$\frac{\mathrm{d}S}{\mathrm{dt}} = -\frac{\beta SI}{N} \qquad \frac{\mathrm{d}I}{\mathrm{dt}} = \frac{\beta SI}{N} - \mu I \qquad \frac{\mathrm{d}R}{\mathrm{dt}} = \mu I$$

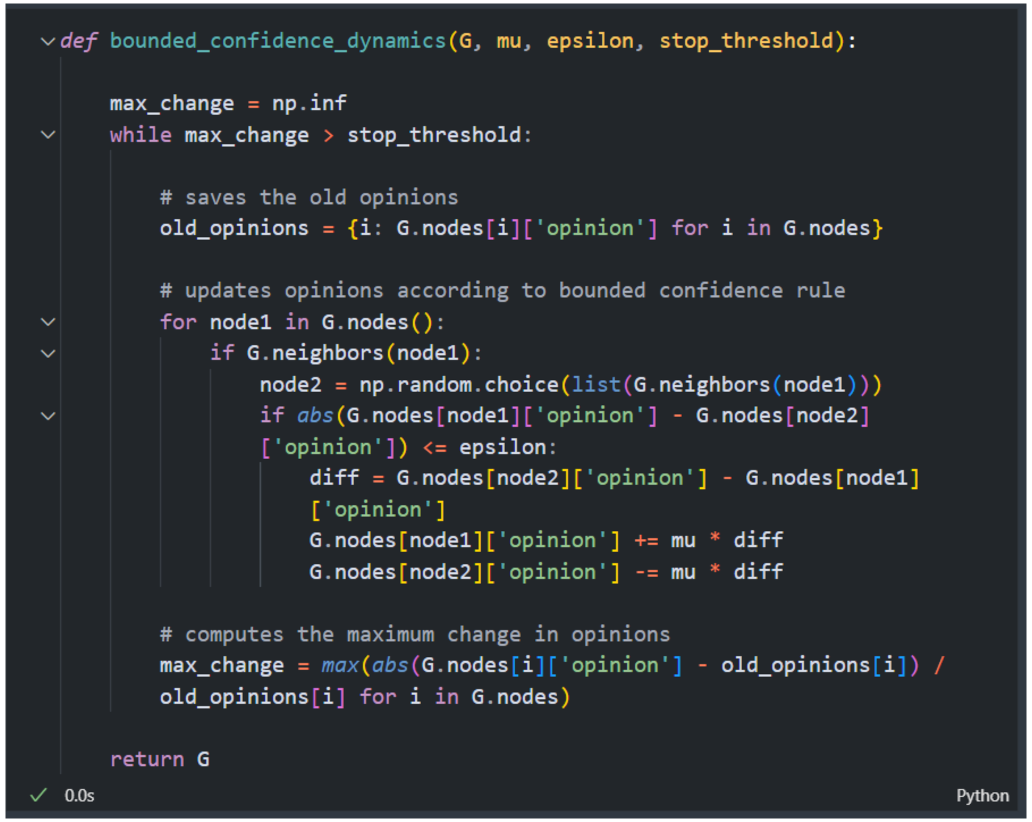
- Outbreak with recovery: SIR
- R_0: spreading rate in a naïve population
- Discrete network model: $I_{t+1} = \beta\langle k\rangle I_t - \mu I_t$
  - $R_0 \equiv \frac{\beta\langle k\rangle}{\mu} = 2.5$
  - $R = \frac{\beta\langle k\rangle}{2\mu'} \leq 1 \quad \therefore \mu' \geq 1.25\mu$

CHAPTER 7 EXERCISES

7.19 Simulate the bounded-confidence model of opinion dynamics on a complete network with $N = 1000$ nodes. The initial opinions are random numbers between zero and one. Consider three different values of the confidence bound: $\epsilon = 0.125, 0.25, 0.5$. For each ϵ, use different values for the convergence parameter, say $\mu = 0.1, 0.3, 0.5$. Run every simulation until each opinion varies by less than 1% between consecutive iterations, and plot a histogram of the final opinions. Does the number of final opinion clusters depend on ϵ? Why or why not? Does it depend on μ? Why or why not? (*Hint*: Feel free to modify the code in this chapter's tutorial to run the simulations.)

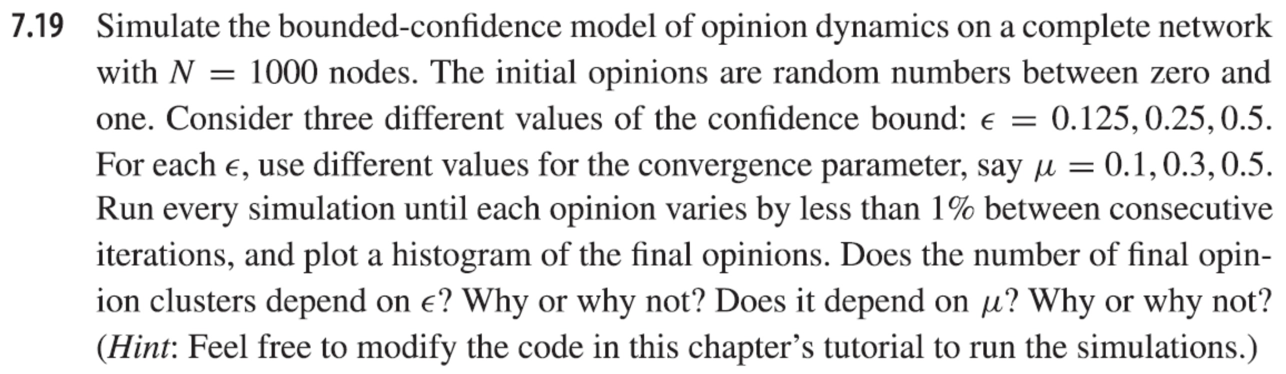


```python
def run_model(N, mu, epsilon, stop_threshold = 0.01, ax = None):

    # generates graph with random initial opinions
    G = nx.complete_graph(N)
    for i in G.nodes:
        G.nodes[i]['opinion'] = np.random.rand()

    # runs the model
    G = bounded_confidence_dynamics(G, mu, epsilon, stop_threshold)

    # plots the histogram of opinions
    opinions = [G.nodes[i]['opinion'] for i in G.nodes()]
    if ax is None:
        _, ax = plt.subplots()
    ax.hist(opinions, bins=np.linspace(0,1,20))
    ax.set_ylim(0, N)
    ax.set_title("mu = {}, epsilon = {}".format(mu, epsilon))
```



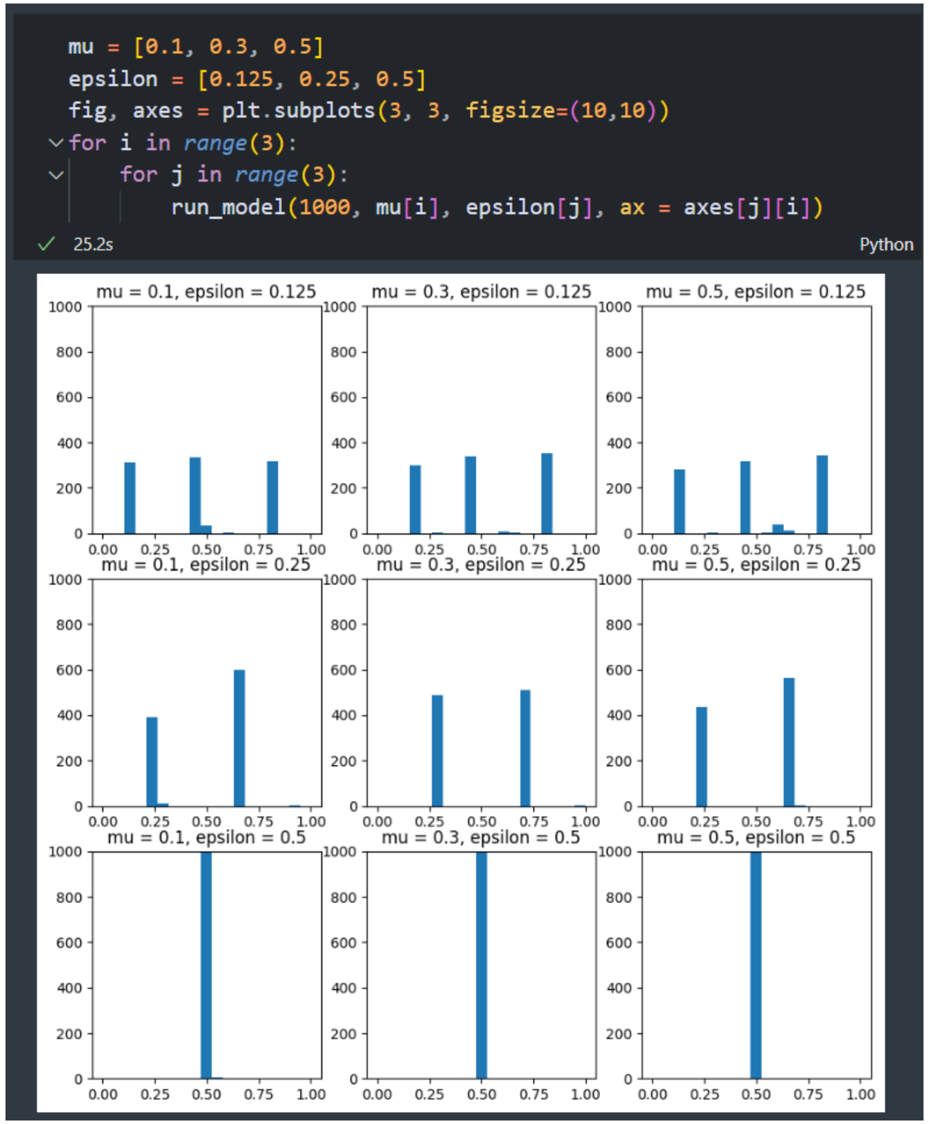
```python
def bounded_confidence_dynamics(G, mu, epsilon, stop_threshold):

    max_change = np.inf
    while max_change > stop_threshold:

        # saves the old opinions
        old_opinions = {i: G.nodes[i]['opinion'] for i in G.nodes}

        # updates opinions according to bounded confidence rule
        for node1 in G.nodes():
            if G.neighbors(node1):
                node2 = np.random.choice(list(G.neighbors(node1)))
                if abs(G.nodes[node1]['opinion'] - G.nodes[node2]
                ['opinion']) <= epsilon:
                    diff = G.nodes[node2]['opinion'] - G.nodes[node1]
                    ['opinion']
                    G.nodes[node1]['opinion'] += mu * diff
                    G.nodes[node2]['opinion'] -= mu * diff

        # computes the maximum change in opinions
        max_change = max(abs(G.nodes[i]['opinion'] - old_opinions[i]) /
        old_opinions[i] for i in G.nodes)

    return G
```

CHAPTER 7 EXERCISES



7.19 Simulate the bounded-confidence model of opinion dynamics on a complete network with $N = 1000$ nodes. The initial opinions are random numbers between zero and one. Consider three different values of the confidence bound: $\epsilon = 0.125, 0.25, 0.5$. For each ϵ, use different values for the convergence parameter, say $\mu = 0.1, 0.3, 0.5$. Run every simulation until each opinion varies by less than 1% between consecutive iterations, and plot a histogram of the final opinions. Does the number of final opinion clusters depend on ϵ? Why or why not? Does it depend on μ? Why or why not? (*Hint*: Feel free to modify the code in this chapter's tutorial to run the simulations.)

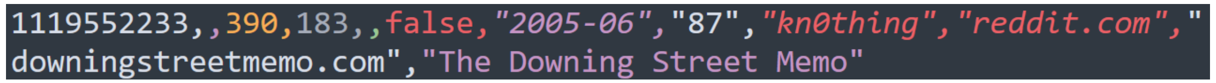
- Number of clusters depends on ϵ
  - Clusters only won't eventually merge if their distance is larger than ϵ
- μ doesn't affect number of clusters, just convergence speed
  - Simple dynamics with smooth convergence to fixed point



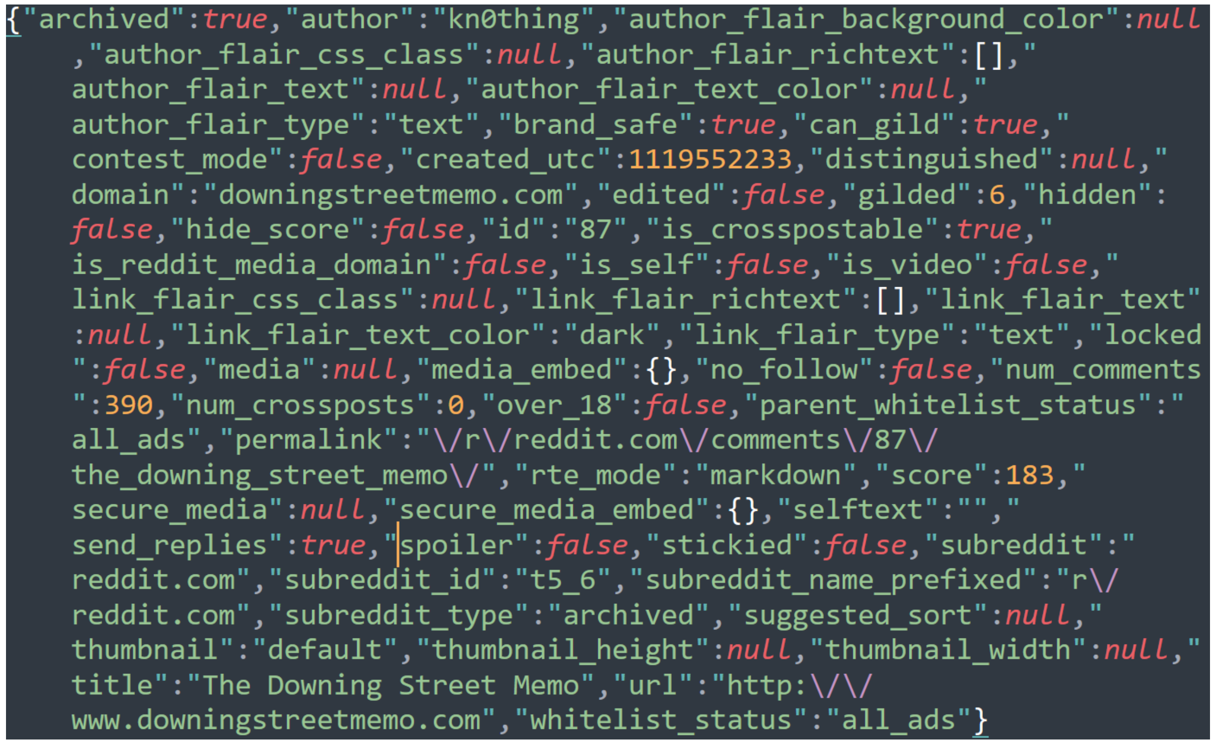
```
mu = [0.1, 0.3, 0.5]
epsilon = [0.125, 0.25, 0.5]
fig, axes = plt.subplots(3, 3, figsize=(10,10))
for i in range(3):
    for j in range(3):
        run_model(1000, mu[i], epsilon[j], ax = axes[j][i])
```

DATA HANDLING

- Data comes in various formats
  - Plain text encodings: CSV, TSV, JSON
  - Special formats: HDF5, parquet, etc
- **CSV**: Comma Separated Values
  - Flat format
  - Not a standardized format
    - Does it have a header? What is the escape character?
    - Headache handling special characters (like \n)
  - Enforces a data schema, storage efficient
- **JSON**: JavaScript Object Notation
  - Nested format
  - Each data entry contains both header and content
  - Schema-less
  - Storage inefficient
  - **Web content (from e.g. APIs) comes in JSON**



```
1119552233,,390,183,,false,"2005-06","87","kn0thing","reddit.com","
downingstreetmemo.com","The Downing Street Memo"
```



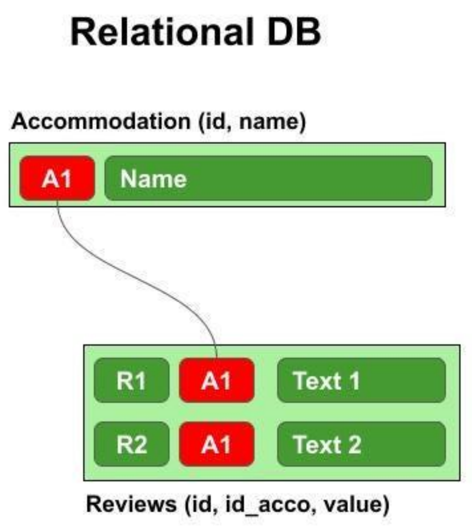
```
{"archived":true,"author":"kn0thing","author_flair_background_color":null
    ,"author_flair_css_class":null,"author_flair_richtext":[],"
    author_flair_text":null,"author_flair_text_color":null,"
    author_flair_type":"text","brand_safe":true,"can_gild":true,"
    contest_mode":false,"created_utc":1119552233,"distinguished":null,"
    domain":"downingstreetmemo.com","edited":false,"gilded":6,"hidden":
    false,"hide_score":false,"id":"87","is_crosspostable":true,"
    is_reddit_media_domain":false,"is_self":false,"is_video":false,"
    link_flair_css_class":null,"link_flair_richtext":[],"link_flair_text"
    :null,"link_flair_text_color":"dark","link_flair_type":"text","locked
    ":false,"media":null,"media_embed":{},"no_follow":false,"num_comments
    ":390,"num_crossposts":0,"over_18":false,"parent_whitelist_status":"
    all_ads","permalink":"\/r\/reddit.com\/comments\/87\/
    the_downing_street_memo\/","rte_mode":"markdown","score":183,"
    secure_media":null,"secure_media_embed":{},"selftext":"","
    send_replies":true,"spoiler":false,"stickied":false,"subreddit":"
    reddit.com","subreddit_id":"t5_6","subreddit_name_prefixed":"r\/
    reddit.com","subreddit_type":"archived","suggested_sort":null,"
    thumbnail":"default","thumbnail_height":null,"thumbnail_width":null,"
    title":"The Downing Street Memo","url":"http:\/\/
    www.downingstreetmemo.com","whitelist_status":"all_ads"}
```

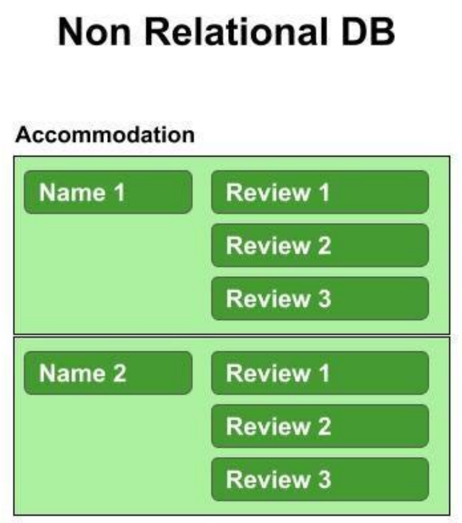
DATA HANDLING AND SIZE

- Data handling depends on size
- Data fits into RAM comfortably: anything works
- Data barely fits into RAM
  - Works for simpler analysis
  - Operations with copies can require >2X the data size
- Data doesn't fit into RAM (e.g. 21TB of reddit data)
  - Chunked data
    - Only possible for some analyses
    - May require multiple data reads
  - **Databases**

DATABASES

- Why databases?
  - Handles all the underlying complexity
  - Handles concurrent operations
    - ACID: Atomicity, Consistency, Isolation, Durability
  - **Fast**
- Types of databases
  - Relational/SQL
    - Data in tables, made for *joining* data between tables
    - Data has schema
    - Examples: PostgreSQL, MySQL, MariaDB, Microsoft SQL Server
  - Non-relational/NoSQL
    - No enforced schema
    - MongoDB, Redis, vector databases (Milvus, Pinecone)

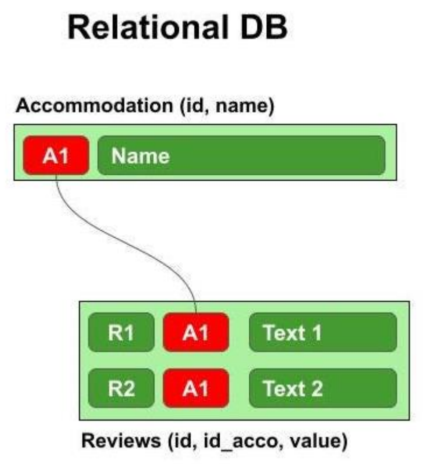




DATABASES

- Which one to use?
- Depends on type of workload
  - Online Transaction Processing (OLTP)
    - Many fast, simple queries concurrently
  - Data Warehouse/Business Intelligence (BI)
    - Few very large, complex queries with a lot of data
- MongoDB
  - Designed for OLTP
  - Bad at BI
- PostgreSQL
  - Highly configurable, good for both for both OLTP and BI

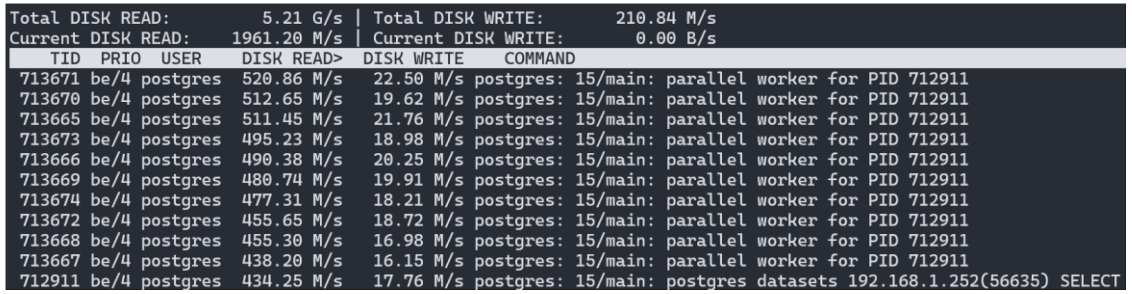
- What to use should depend on
  - Data type (schema vs schema-less)
  - Data size
  - Workload type
  - If your data is relational
    - In other words, if you do table joins
    - Many industry horror stories of MongoDB implementations wrongly assuming data wasn't relational



DATABASES

- MongoDB
  - Handles JSON natively
  - Handles millions of concurrent operations
  - Native data compression
  - Single-threaded
  - Very slow table joins

- PostgreSQL
  - JSON data must be cleaned and converted first
  - Language syntax can take a while to get used to
  - No native compression, relies on filesystem/plugins
  - Multi-threaded, scales to dozens of cores
  - Fast table joins



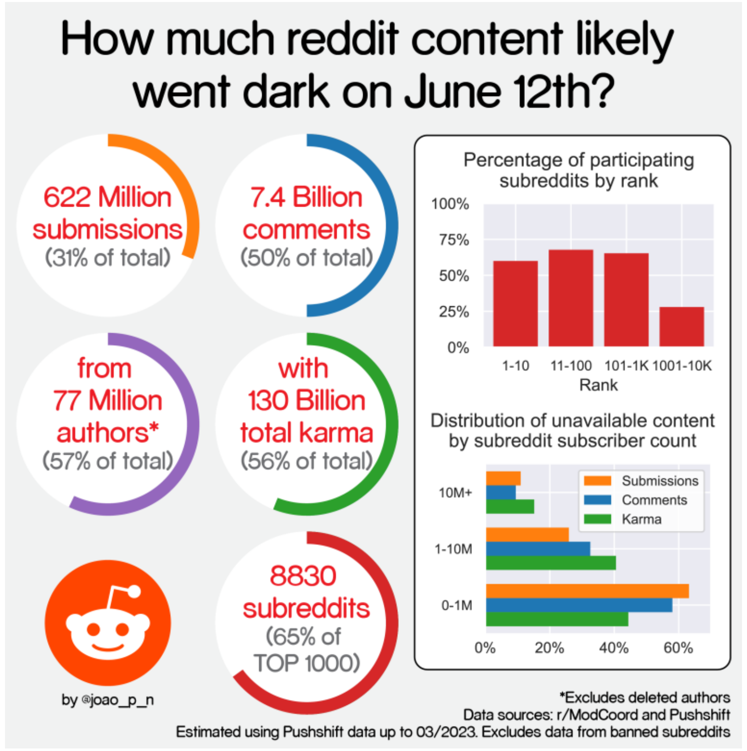
```
Total DISK READ:         5.21 G/s | Total DISK WRITE:       210.84 M/s
Current DISK READ:    1961.20 M/s | Current DISK WRITE:       0.00 B/s
    TID  PRIO  USER     DISK READ>  DISK WRITE    COMMAND
 713671 be/4 postgres  520.86 M/s   22.50 M/s postgres: 15/main: parallel worker for PID 712911
 713670 be/4 postgres  512.65 M/s   19.62 M/s postgres: 15/main: parallel worker for PID 712911
 713665 be/4 postgres  511.45 M/s   21.76 M/s postgres: 15/main: parallel worker for PID 712911
 713673 be/4 postgres  495.23 M/s   18.98 M/s postgres: 15/main: parallel worker for PID 712911
 713666 be/4 postgres  490.38 M/s   20.25 M/s postgres: 15/main: parallel worker for PID 712911
 713669 be/4 postgres  480.74 M/s   19.91 M/s postgres: 15/main: parallel worker for PID 712911
 713674 be/4 postgres  477.31 M/s   18.21 M/s postgres: 15/main: parallel worker for PID 712911
 713672 be/4 postgres  455.65 M/s   18.72 M/s postgres: 15/main: parallel worker for PID 712911
 713668 be/4 postgres  455.30 M/s   16.98 M/s postgres: 15/main: parallel worker for PID 712911
 713667 be/4 postgres  438.20 M/s   16.15 M/s postgres: 15/main: parallel worker for PID 712911
 712911 be/4 postgres  434.25 M/s   17.76 M/s postgres: 15/main: postgres datasets 192.168.1.252(56635) SELECT
```

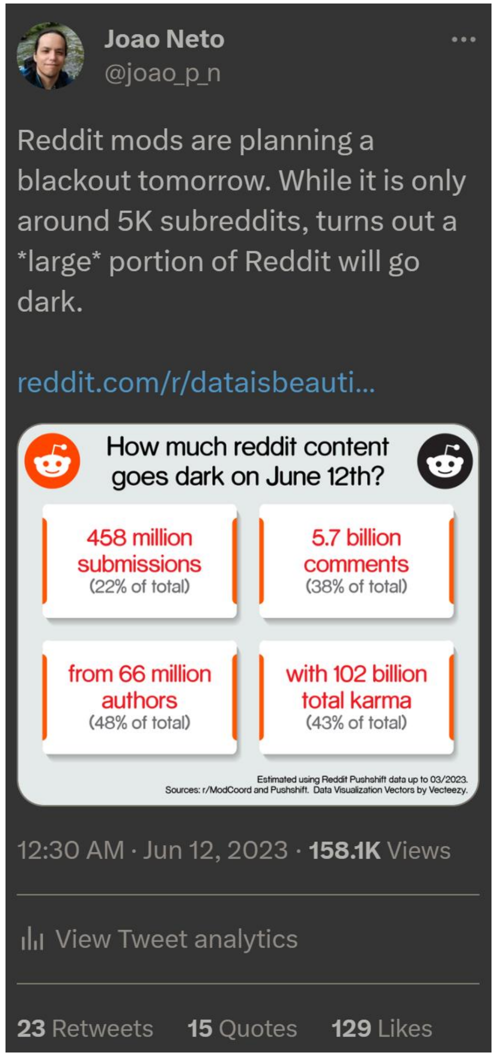
DATABASES

- Overall, PostgreSQL is better for large-scale social media analysis
  - Flexible
  - More widespread usage, more help online
  - ChatGPT SQL code is pretty good
    - Tip: tell it the table sizes for optimization
  - Highly sought industry skill
- But
  - Hard to set up and configure (DBA is a well-paying job)
  - Data cleaning takes a long time
  - MongoDB has much lower barrier of entry

DATABASES

- Databases allows analyses that are impossible otherwise
- Reddit blackout: how much of Reddit was dark?
  - Have to analyze the entire dataset (18B items) in multiple ways




Joao Neto
@joao_p_n

Reddit mods are planning a blackout tomorrow. While it is only around 5K subreddits, turns out a *large* portion of Reddit will go dark.

reddit.com/r/dataisbeauti...

How much reddit content goes dark on June 12th?

458 million submissions (22% of total)

5.7 billion comments (38% of total)

from 66 million authors (48% of total)

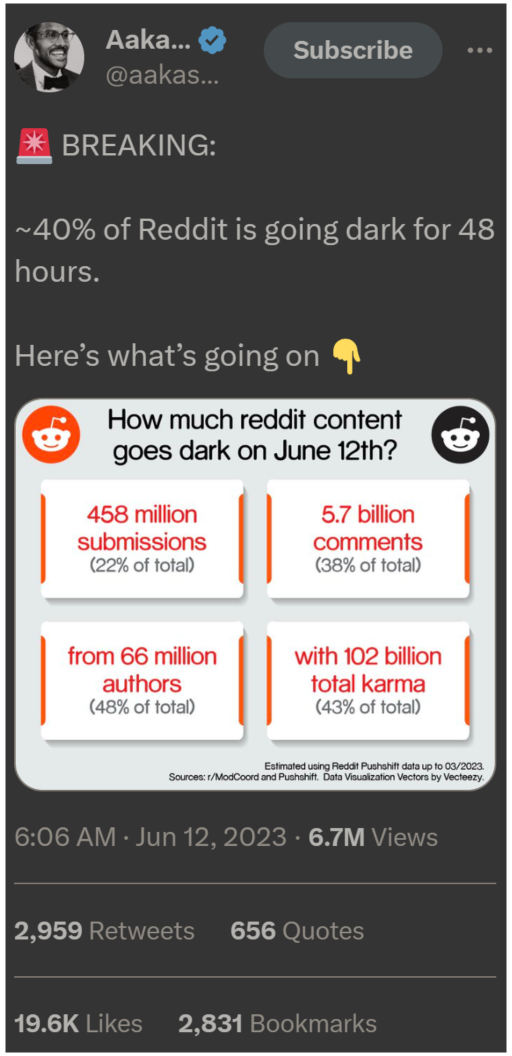
with 102 billion total karma (43% of total)

Estimated using Reddit Pushshift data up to 03/2023.
Sources: r/ModCoord and Pushshift. Data Visualization Vectors by Vecteezy.

12:30 AM · Jun 12, 2023 · **158.1K** Views

View Tweet analytics

23 Retweets **15** Quotes **129** Likes


Aaka...
@aakas...
Subscribe

🚨 BREAKING:

~40% of Reddit is going dark for 48 hours.

Here’s what’s going on 👇

How much reddit content goes dark on June 12th?

458 million submissions (22% of total)

5.7 billion comments (38% of total)

from 66 million authors (48% of total)

with 102 billion total karma (43% of total)

Estimated using Reddit Pushshift data up to 03/2023.
Sources: r/ModCoord and Pushshift. Data Visualization Vectors by Vecteezy.

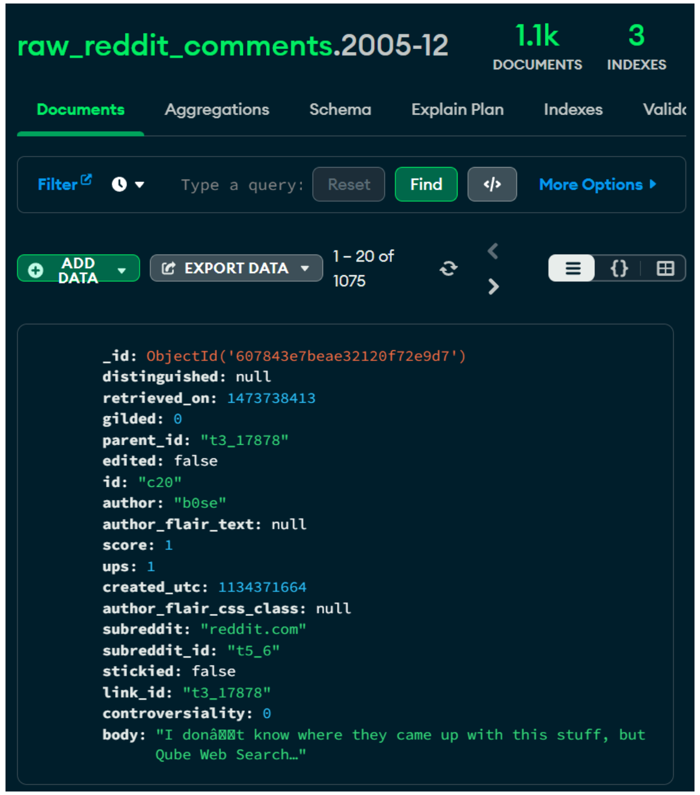
6:06 AM · Jun 12, 2023 · **6.7M** Views

2,959 Retweets **656** Quotes

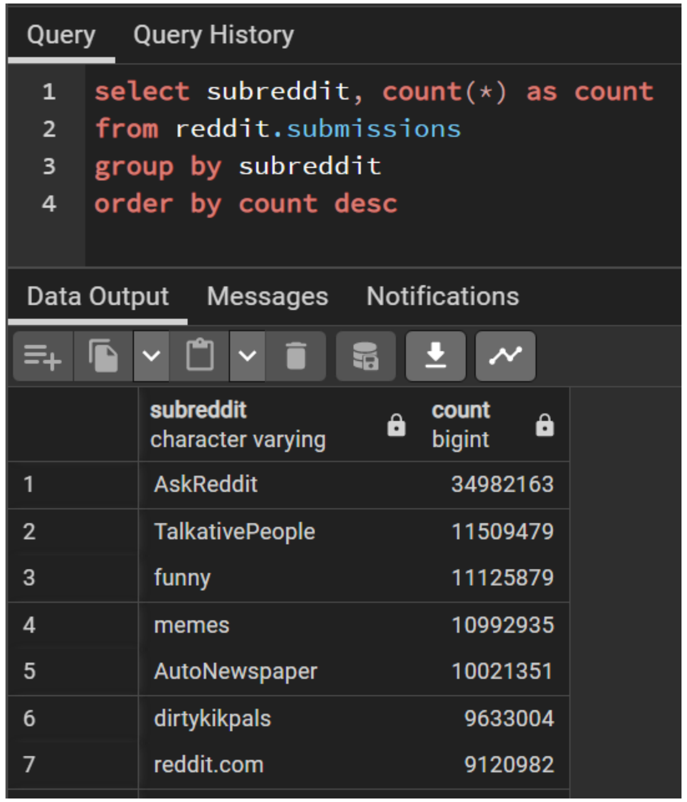
19.6K Likes **2,831** Bookmarks

DATABASES

- One idea
  - MongoDB for raw data storage and lookup
    - Compressed data
    - Still faster than file parsing
    - Indexes can be added to greatly speed it up
    - Allows checking of pre-cleaned data
  - PostgreSQL for data analysis
    - Just plain faster



```
_id: ObjectId('607843e7beae32120f72e9d7')
distinguished: null
retrieved_on: 1473738413
gilded: 0
parent_id: "t3_17878"
edited: false
id: "c20"
author: "b0se"
author_flair_text: null
score: 1
ups: 1
created_utc: 1134371664
author_flair_css_class: null
subreddit: "reddit.com"
subreddit_id: "t5_6"
stickied: false
link_id: "t3_17878"
controversiality: 0
body: "I donâ€™t know where they came up with this stuff, but
      Qube Web Search…"
```



```sql
select subreddit, count(*) as count
from reddit.submissions
group by subreddit
order by count desc
```

	subreddit character varying	count bigint
1	AskReddit	34982163
2	TalkativePeople	11509479
3	funny	11125879
4	memes	10992935
5	AutoNewspaper	10021351
6	dirtykikpals	9633004
7	reddit.com	9120982