Rumour Spreading Analysis on Twitter

SydneySiege

Agenda

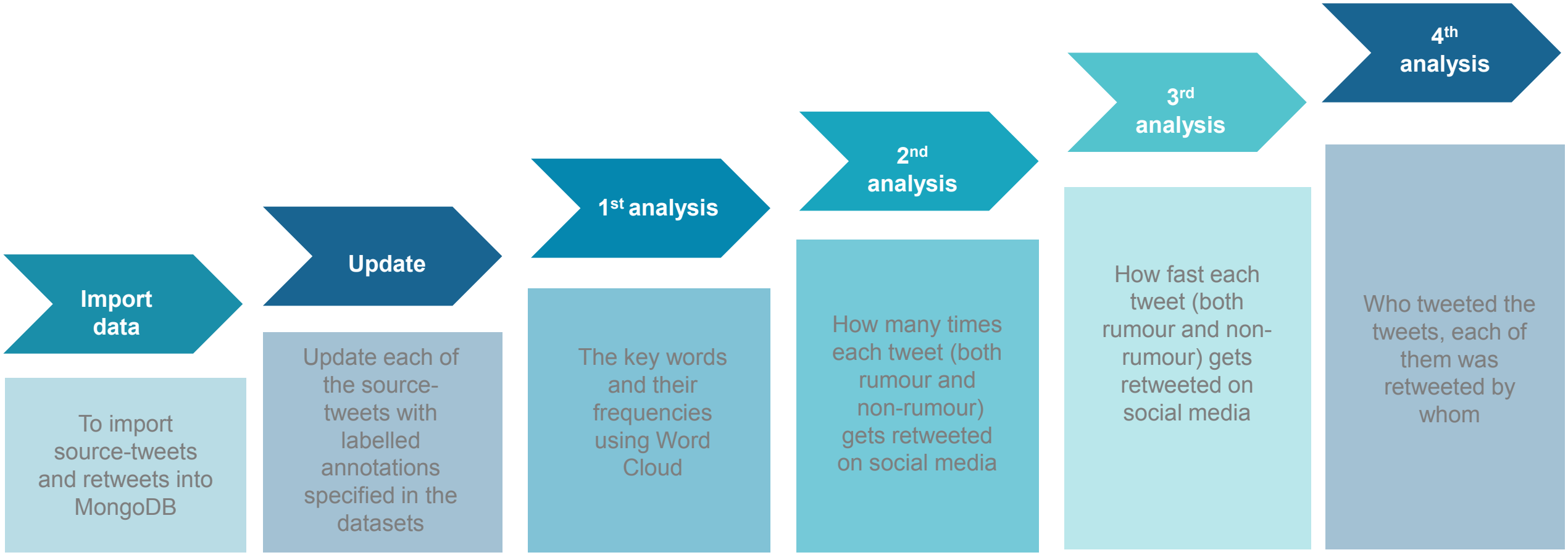
01 Introduction

02 The project steps

03 results

Rumour Spreading Analysis on Twitter

The project steps



“

Where each tweet is tweeted by visualizing users’ locations on the map using aforementioned tools or other visualization tools such as Google Maps API.

”

5th ANALYSIS

EXTRA POINTS

IMPORT DATA

python™

```
import pymongo
import json
from pymongo import MongoClient

#Connect with Mongodb Atlas

client = pymongo.MongoClient("mongodb://wisitsak:wp11223344@cluster0-shard-00-00.99fit.mongodb.net:27017,cluster0-shard-00-01.99fit.mongodb.net:27017,cluster0-shard-00-02.99fit.mongodb.net:27017/Bigdata?ssl=true&replicaSet=atlas-uuej29-shard-0&authSource=admin&retryWrites=true&w=majority")
db = client.sydneysiege

# create Collection on
db.create_collection('source-tweets')
db.create_collection('retweets')

collection = db. source-tweets
collection2 = db. retweets
collection3 = db. annotation
```

IMPORT DATA

python™

```
path = "*/source-tweets/*.json"
for filename in glob.glob(path, recursive=True):
    with open(filename) as f:
        file_data = json.load(f)
        collection.insert_one(file_data)

path2 = "*/retweets.json"
for filename in glob.glob(path2, recursive=True):
    with open(filename) as f:
        for line in f:
            file_data = json.loads(line)
            collection2.insert_one(file_data)
```

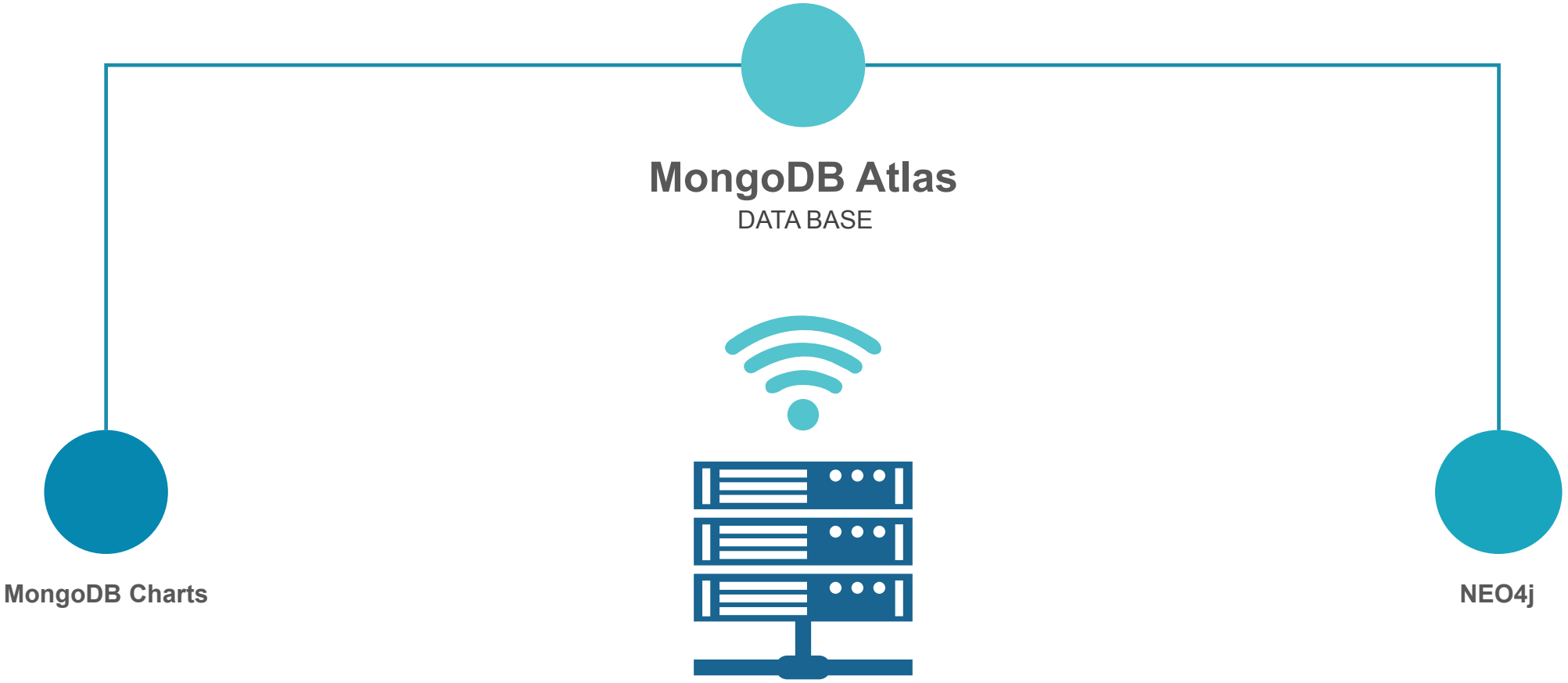
Update DATA

python™

```
path3 = "*/annotation.json"
for filename in glob.glob(path3, recursive=True):
    with open(filename) as f:
        file_data = json.load(f)
        collection.update_many(
{"id":int(filename.strip("\\annotation.json"))},{"$set":{"misinformation":int(file_
data["misinformation"])}})

fake_word = []
for x in collection.find({"misinformation":1}):
    word = re.findall(r"[\w']+", x['text'])
    for y in word:
        fake_word.append(y.lower())
print(fake_word)
```

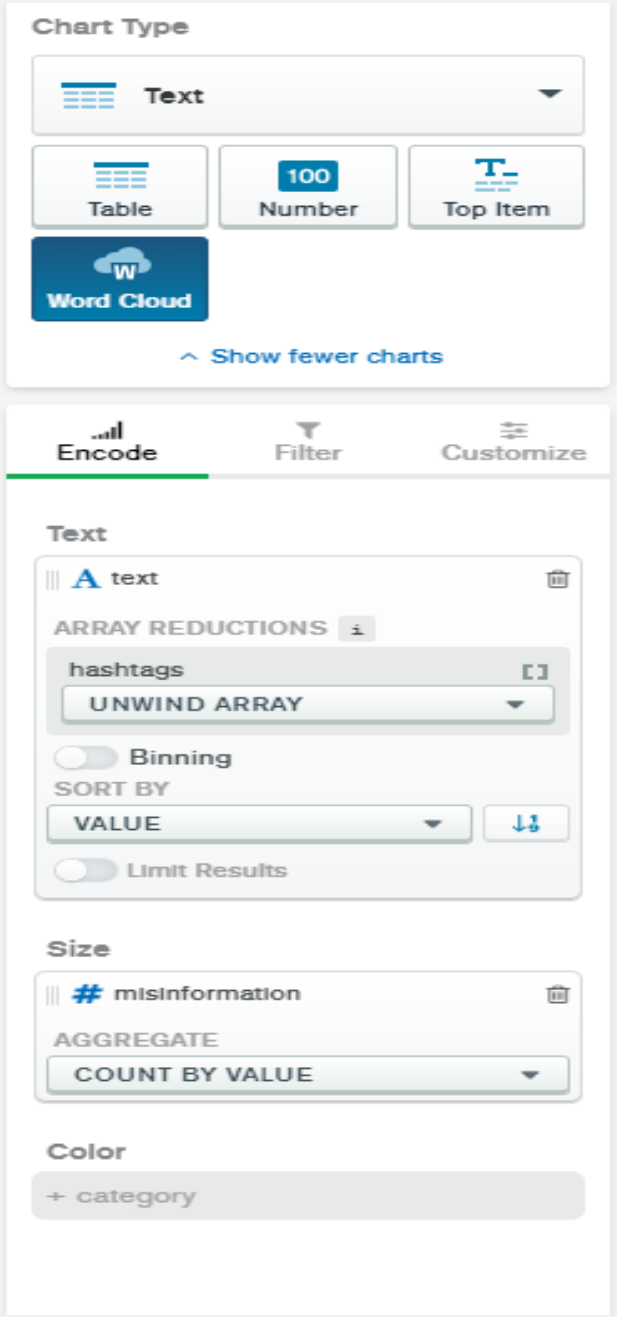
ANALYSIS



1st ANALYSIS

MongoDB Charts

The key words and their frequencies using Word Cloud

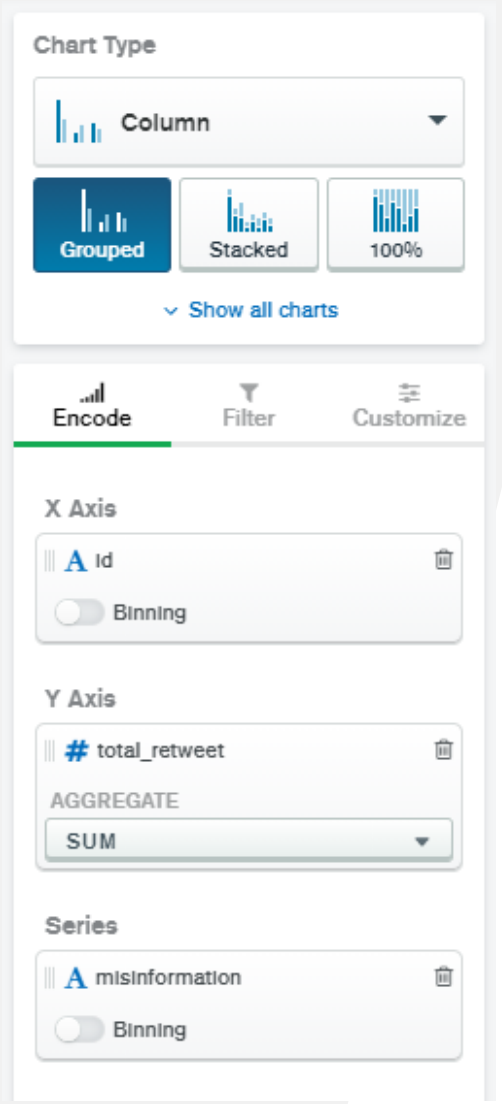


2nd ANALYSIS

MongoDB Charts

How many times each tweet (both rumour and non-rumour) gets retweeted on social media

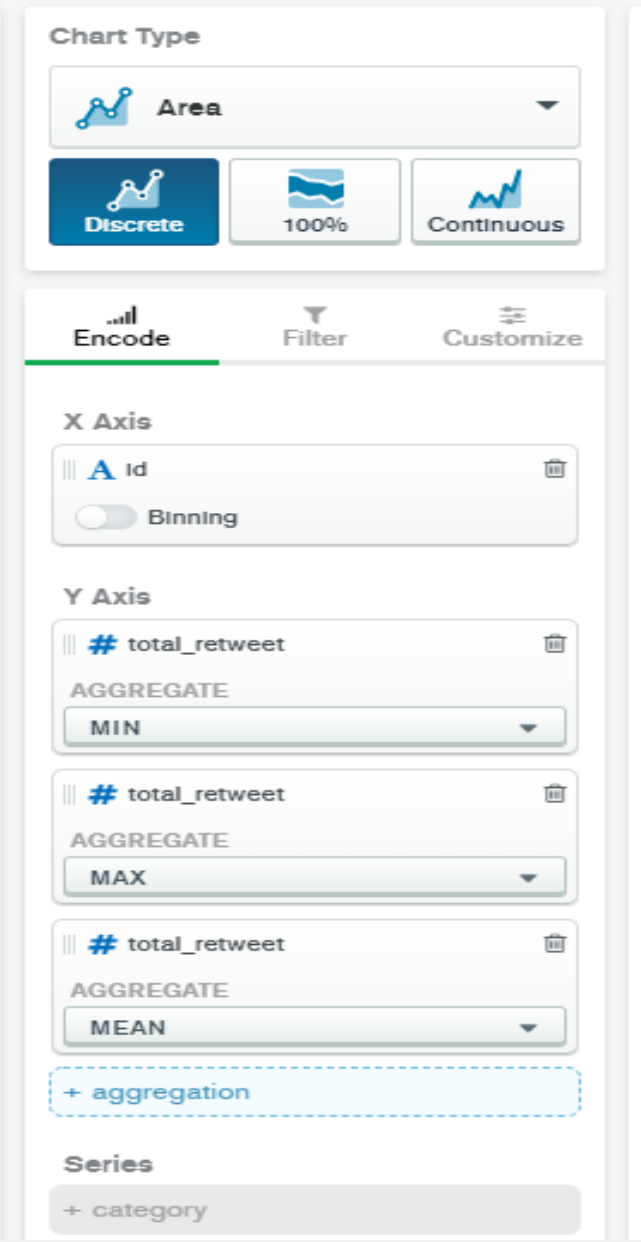
```
{
    "$lookup": {
        "from": "retweets",
        "localField": "id",
        "foreignField": "retweeted_status.id",
        "as": "retweets"
    }
},
{
    "$project": {
        "id": 1,
        "text": 1,
        "city": 1,
        "misinformation": 1,
        "created_at": 1,
        "_id": 0,
        "total_retweet": {
            "$size": "$retweets"
        }
```



3rd ANALYSIS

MongoDB Charts

How fast each tweet (both rumour and non-rumour) gets retweeted on social media



4th ANALYSIS

NEO4j

Who tweeted the tweets, each of them was retweeted by whom by converting json objects

[19]:
```
from py2neo import Graph
```

[33]:
```
neo4j = Graph('bolt://localhost:7687', auth=('neo4j','11223344'))
```

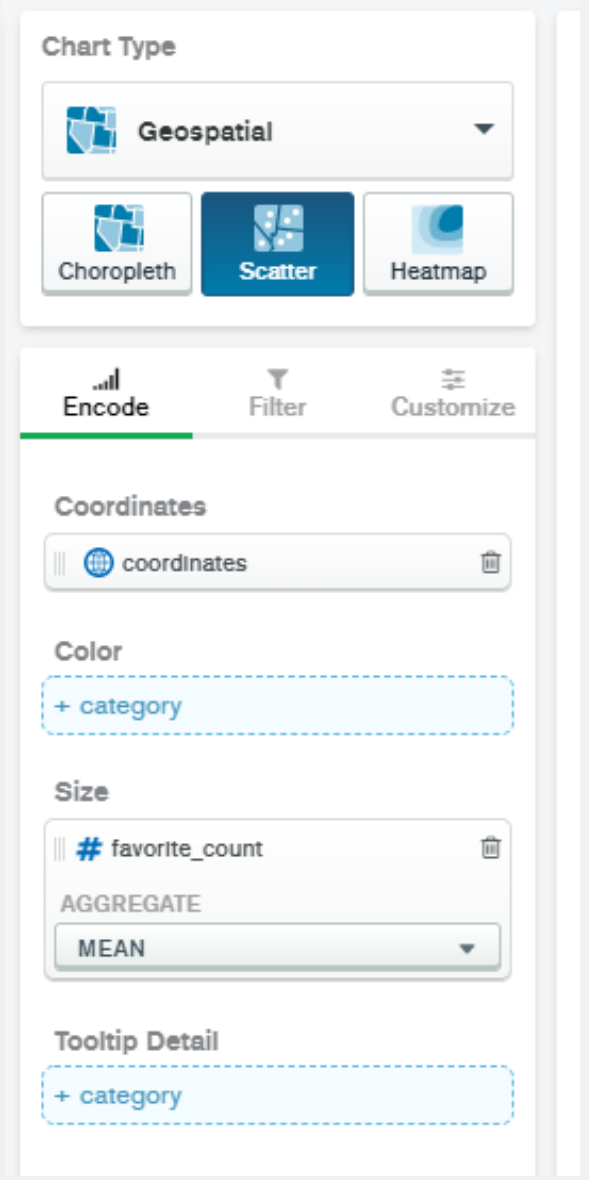
[24]:
```
tweets = db.sourcetweets
retweets = db.retweets
```

[34]:
```
# Create Tweet Node
for tweet in collection.find({}):
    neo4j.run('CREATE (:Tweet {id: $id, user_name: $user_name, text: $text, misinformation: $misinformation})',
           id = tweet['id'], user_name = tweet['user']['name'], text = tweet['text'], misinformation = tweet['misinformation']
    )
```

[35]:
```
# Create Retweet Node
for retweet in collection2.find({}):
    neo4j.run('MATCH (t:Tweet {id: $tweet_id}) \
              CREATE (r:Retweet{id: $id, user_name: $user_name, text: $text})-[:RETWEETKUB]->(t) RETURN r',
             tweet_id = retweet['retweeted_status']['id'], id = retweet['id'], user_name = retweet['user']['name'], text = retweet['text'])
```

5th ANALYSIS

MongoDB Charts




RESULTS

RESULTS

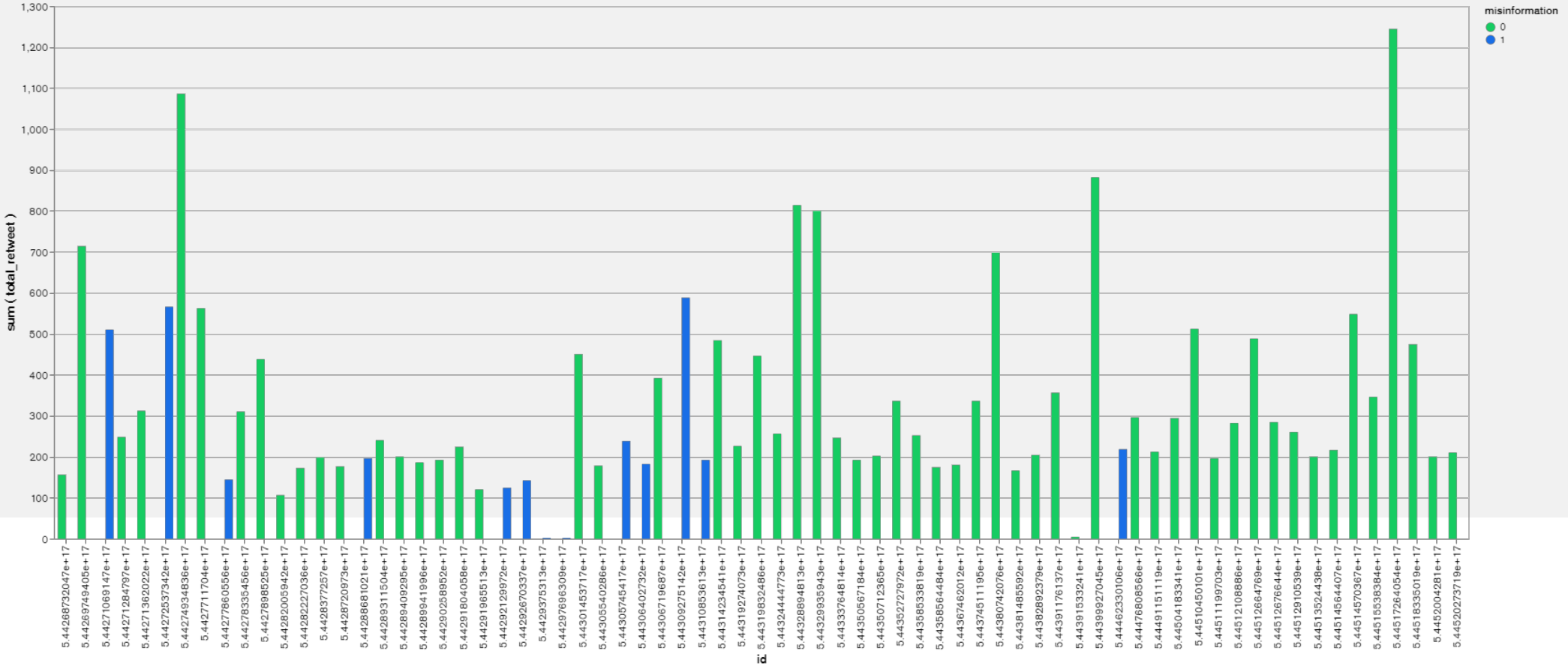
1st ANALYSIS

Word Cloud

RESULTS

2nd ANALYSIS

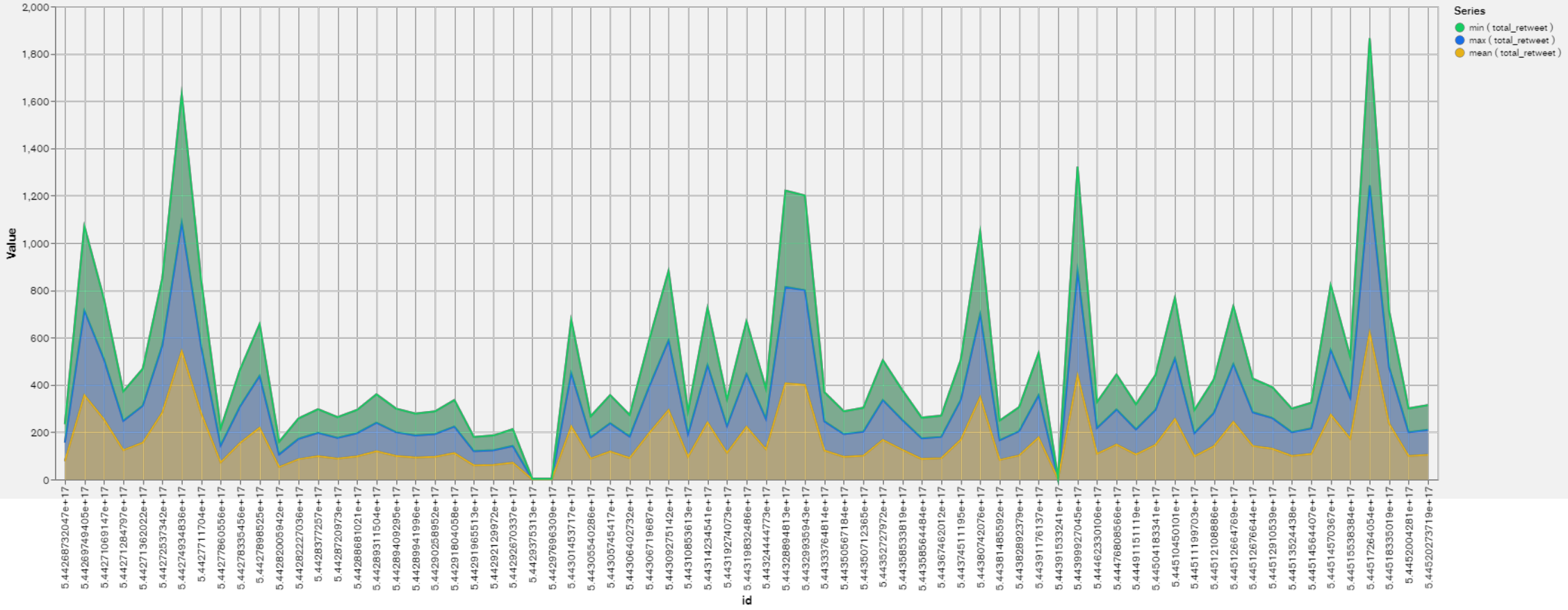
Times each tweet



RESULTS

3rd ANALYSIS

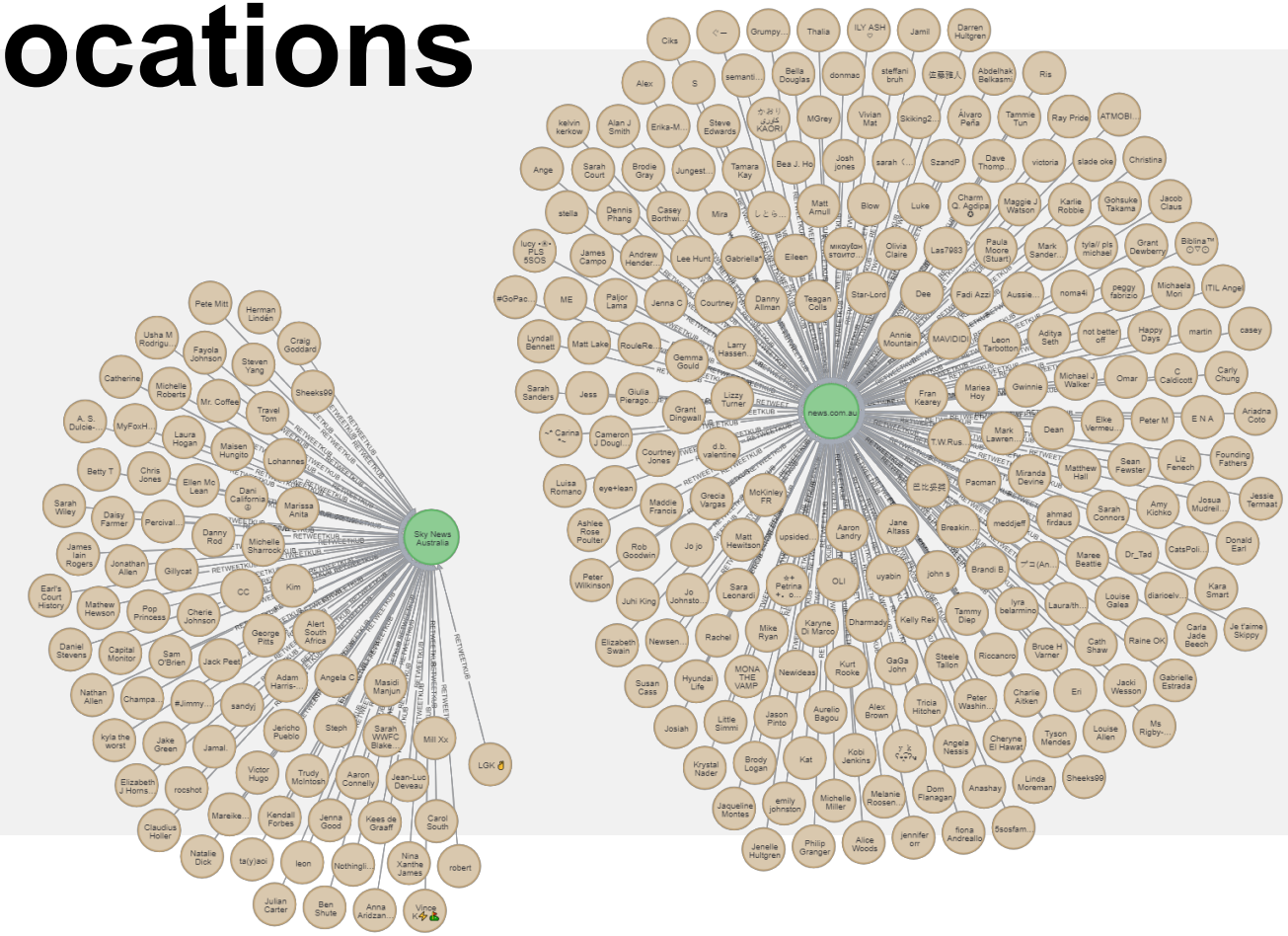
Times each tweet



RESULTS

4th ANALYSIS

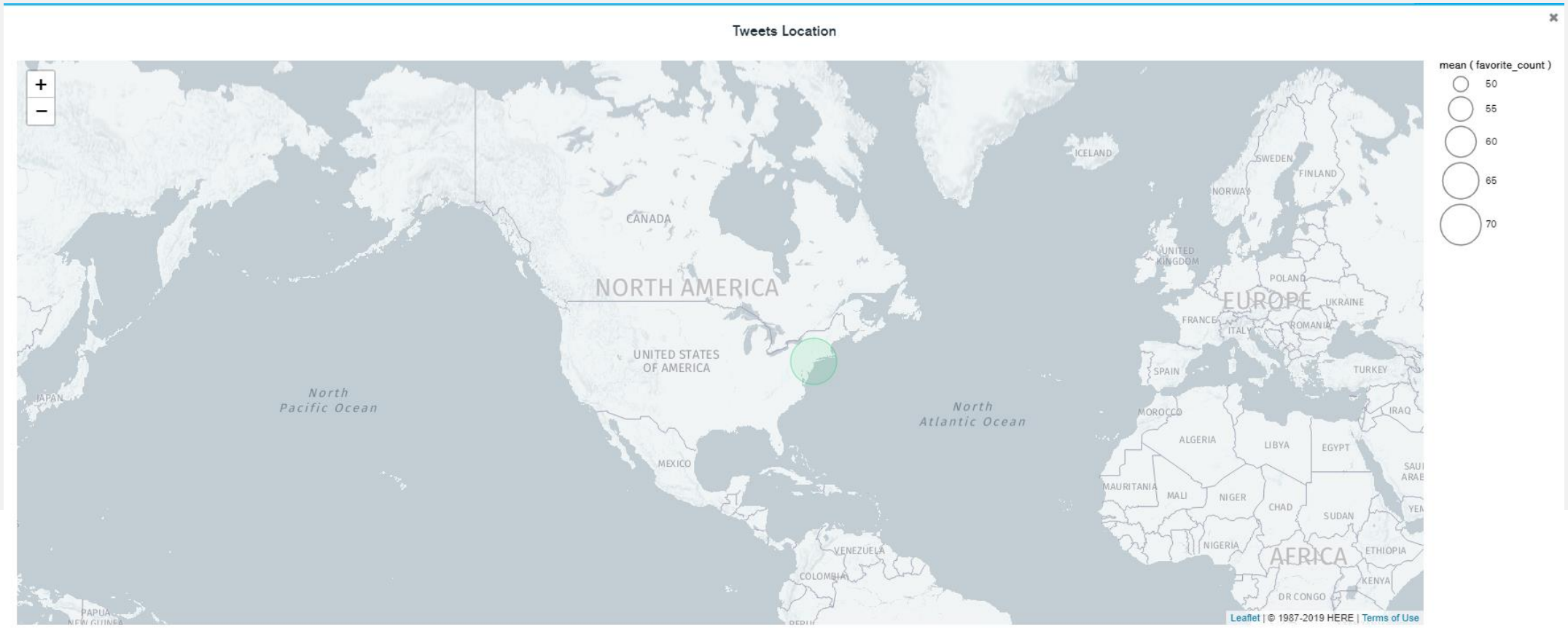
Tweeted locations



RESULTS

5th ANALYSIS

Tweeted locations




THANK YOU