I have no affiliation with either,
I’m just an excited user!
What is this elasticsearch?

- Document database built for search
- Distributed
- Centered around a RESTful API and Json
Why use Elasticsearch?

- It’s fast!!!
- Lot’s of nifty functions
- Open Source
Why NOT use elasticsearch?

- no-SQL
- not relationel-database
- non-tabular

It’s built for speed
not total accuracy
(they say so, but never been a problem for me)

Can be a bit complex
nested lists etc.
What can Elasticsearch be used for?

Search the world!
...in millions of text documents
...near a geolocation
...within a given timespan
...

Fast...!

Analyze stuff
...text
...timeseries
...significant terms
...
Basic concept

Index

Type

Document

Field

Field

Field

Field

...
Pack it up!

library(elastic)

https://github.com/ropensci/elastic
https://cran.r-project.org/web/packages/elastic/index.html
ShowTime

library(elastic)

elastic::connect(es_host = "127.0.0.1",
es_port = "9200")

Your in...
To do a search

Index: ‘twitter_tweet’

Type: ‘tweet’

Document
GET THIS

Field
"data" in message
To do a search

```python
elastic::Search(index = "twitter_tweet",
                type = "tweet",
                q = "message: 'data'",
                asdf = TRUE,
                size = 30)
```

≈ 0.55 sec.
Searched 3.573.908 documents

Return the top 30 out of 9829
What about aggregation?

Count no. of tweets pr. month

```r
cpy <- list(
  aggregations = list(
    tweets_over_time = list(
      date_histogram = list(
        field = "created_time",
        interval = "month"
      )
    )
  )
)

ES_data <- elastic::Search(index = "twitter_tweet",
type = "tweet",
body = query,
asdf = TRUE,
size = 0)
```
What about aggregation?

```r
library(ggplot2)

ggplot(data = ES_data$aggregations$tweets_over_time$buckets) +
  geom_line(aes(y = key, x = doc_count))
```
Combine search and aggregation

```python
list(
    query = list(
        match = list(
            message = "data"
        )
    ),
    aggregations = list(
        tweets_over_time = list(
            date_histogram = list(
                field = "created_time",
                interval = "month"
            )
        )
    )
)`
Significant Terms

```python
list(
    query = list(
        match = list(
            message = "data"
        ),
        aggregations = list(
            significant_Hashtags = list(
                significant_terms = list(
                    field = "tags"
                )
            )
        )
    )
)
```

```
+----------+----------+----------------+----------+
| key      | doc_count| score           | bg_count |
|----------+----------+----------------+----------+
| data     | 456      | 18.0137084      | 467      |
| Data     | 48       | 1.9420388       | 48       |
| NOAA     | 72       | 1.9395971       | 108      |
| emotionalagility | 33      | 1.2957838       | 34       |
| trád     | 44       | 1.2620697       | 62       |
| datajob  | 28       | 1.1328560       | 28       |
| bigdata  | 102      | 1.0040346       | 416      |
| CO2      | 75       | 0.7737221       | 292      |
| ...      |          |                 |          |
```
To sum up

It’s fast!!!

powerful

...and fairly simple to use