

# What is this all about?

How to understand GDELT with analysis with MongoDB, Atlas, Atlas Charts and ggplot2 (R)

MongoDB World Hackathon '22

The MongoDB World Hackathon '22 is a six week challenge with amazing prizes and swag up for grabs. Participate in a team of up to 4 hackers, or as an individual.

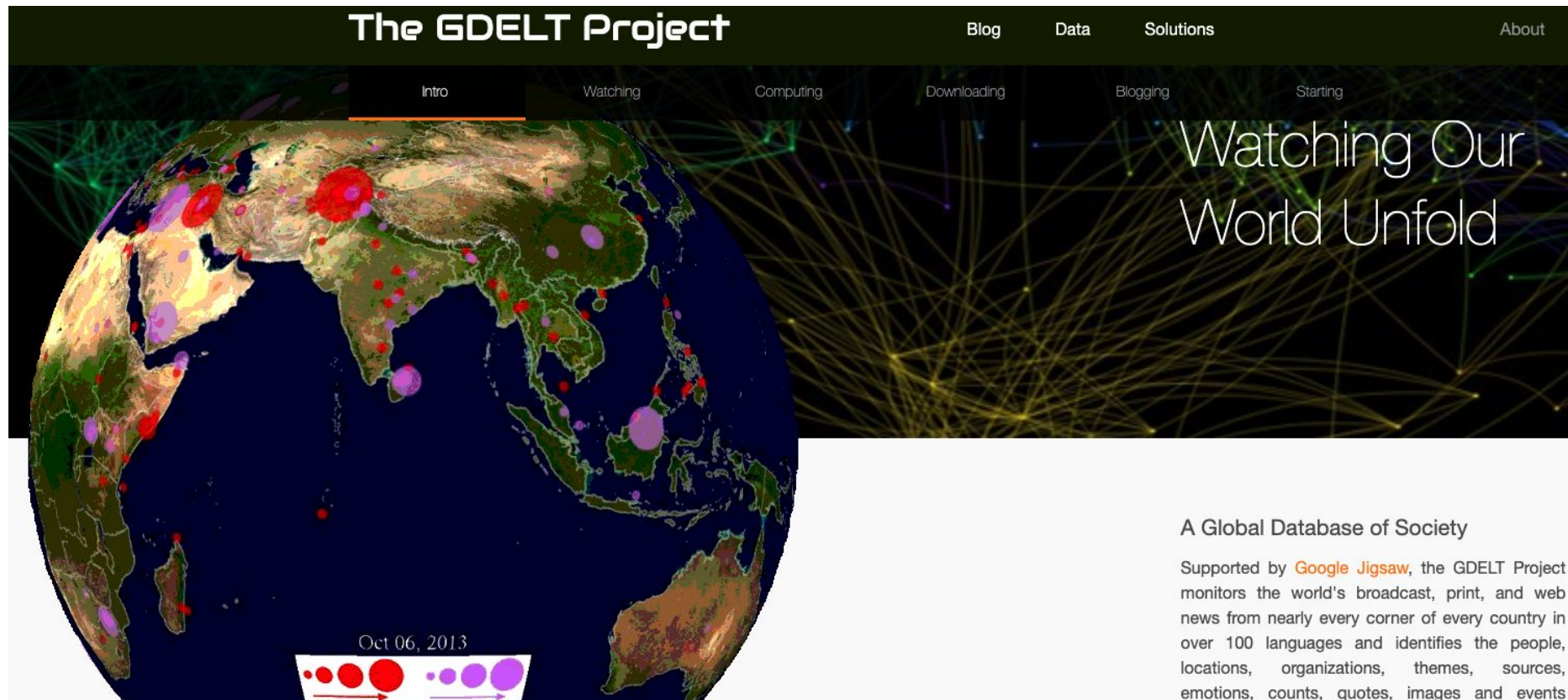
■ MongoDB World Hackathon '22 ▸ all ▸ all tags ▸ all ▸ Latest New (3) Unread (3) Top Events Event

+ New Topic 🔔

Category Topics Latest

The screenshot shows a dark-themed web interface for the MongoDB World Hackathon '22. At the top, the title 'MongoDB World Hackathon '22' is underlined. Below it, a paragraph describes the event as a six-week challenge with prizes and swag, open to teams of up to 4 hackers or individuals. A navigation bar contains several buttons: a category button 'MongoDB World Hackathon '22' with a right arrow, followed by 'all', 'all tags', and 'all', each also with a right arrow. The 'Latest' button is highlighted in red. To the right of 'Latest' are links for 'New (3)', 'Unread (3)', 'Top', 'Events', and 'Event'. Below the navigation bar, there is a '+ New Topic' button in a red rounded rectangle and a bell icon for notifications. At the bottom, the word 'Category' is on the left, and 'Topics' and 'Latest' are on the right.

# GDELT - An on-line source of current events around the world



# GDELT has Events

Events have Actors (Govt, People, Organizations)

Actor1 - Initiator

Actor2 - Responder

EventCode - What happened

## Cameo Event Codes

---

### Chapter 6

### CAMEO EVENT CODES

#### **01: MAKE PUBLIC STATEMENT**

- 010: Make statement, not specified below
- 011: Decline comment
- 012: Make pessimistic comment
- 013: Make optimistic comment
- 014: Consider policy option
- 015: Acknowledge or claim responsibility
- 016: Deny responsibility
- 017: Engage in symbolic act
- 018: Make empathetic comment
- 019: Express accord

#### **02: APPEAL**

- 020: Make an appeal or request, not specified below
- 021: Appeal for material cooperation, not specified below
  - 0211: Appeal for economic cooperation
  - 0212: Appeal for military cooperation
  - 0213: Appeal for judicial cooperation
  - 0214: Appeal for intelligence
- 022: Appeal for diplomatic cooperation (such as policy support)
- 023: Appeal for aid, not specified below
  - 0231: Appeal for economic aid
  - 0232: Appeal for military aid
  - 0233: Appeal for humanitarian aid
  - 0234: Appeal for military protection or peacekeeping
- 024: Appeal for political reform, not specified below
  - 0241: Appeal for change in leadership
  - 0242: Appeal for policy change
  - 0243: Appeal for rights
  - 0244: Appeal for change in institutions, regime

# How to work with the Data

Files distributed as zip files CSV format

Many events means a large dataset

New events are being added continuously.

Are all countries of equal interest, or some countries get more coverage?

What can be learnt from the which Actors are paired together (Actor1 / Actor2)?

What events dominate?, GoldsteinScale, AvgTone ...

**WHAT IS ALL THIS ABOUT?**

# Mongo Rules

Atlas is fast, and there are plans for learning and development *M0 cluster*

Mongo Query Language is rich, and beyond simple queries there is a lambda logic

DSL - Aggregation.

Tooling: Compass

Atlas Charts

Interoperability

See my [blog](#)

mgd22.eventscsv

529.0k  
DOCUMENTS INDEX

The screenshot shows the MongoDB Compass interface with the 'Aggregations' tab selected. The pipeline consists of two stages: '\$addFields' and '\$convert'. The '\$addFields' stage is currently selected, and its configuration is visible in the left pane. The right pane shows the output of the pipeline, displaying two sample documents with their transformed fields.

```
1 {  
2   "geoCodes": {  
3     "$map": {  
4       "input": "$geoCodes",  
5       "as": "a",  
6       "in": {  
7         "$map": {  
8           "input": "$a",  
9           "as": "b",  
10          "in": {  
11            "$convert": {  
12              "input": "$b",  
13              "to": "double",  
14              "onError": ""  
15            }  
16          }  
17        }  
18      }  
19    }  
20  }  
21 }
```

Output after \$addFields stage (Sample of 20 documents)

<pre>{   "_id": ObjectId('627b346f6a78567c9063674b'),   "GlobalEventId": 1036450750,   "Day": 20120331,   "MonthYear": 201203,   "Year": 2012,   "FractionDate": 2012.2493,   "Actor1Code": "USA",   "Actor1Name": "CALIFORNIA" }</pre>	<pre>{   "_id": ObjectId('627b346f6a78567c9063674c'),   "GlobalEventId": 1036450751,   "Day": 20210329,   "MonthYear": 202103,   "Year": 2021,   "FractionDate": 2021.2438,   "Actor1Code": "",   "Actor1Name": "" }</pre>
---	--

# R - ggplot2

How to get the data into R:

Use mongolite as follows:

```
library( dplyr)
```

```
library( ggplot2)
```

```
library( mongolite)
```

```
hackDB = mongo( collection="eventscsv", db="mgd22", url=mongoworld)
```

Where mongoworld =  
mongodb+srv://yyyy:xxxxxxxxx@mongoworld.rgeiy.mongodb.net/test?retryWrites=true&w=majority

# Using the mongolite connection

Just like the console - that sort of simple

% Create a query or an aggregation:

```
Filter2 = { "Actor1Code": { "$in": [ "AUS", "FIN", "IRN", "ISR", "ITA", "JEW", "JPN", "FRA", "RUS", "SWZ", "TUR",  
"UKR" ] }, "SOURCEURL": { "$exists": 1 } }
```

% Load the data into a data.frame (R is painfully similar to Pandas)

```
mongodata <- hackDB$find( filter2)
```

```
> summary(mongodata$Day)
```

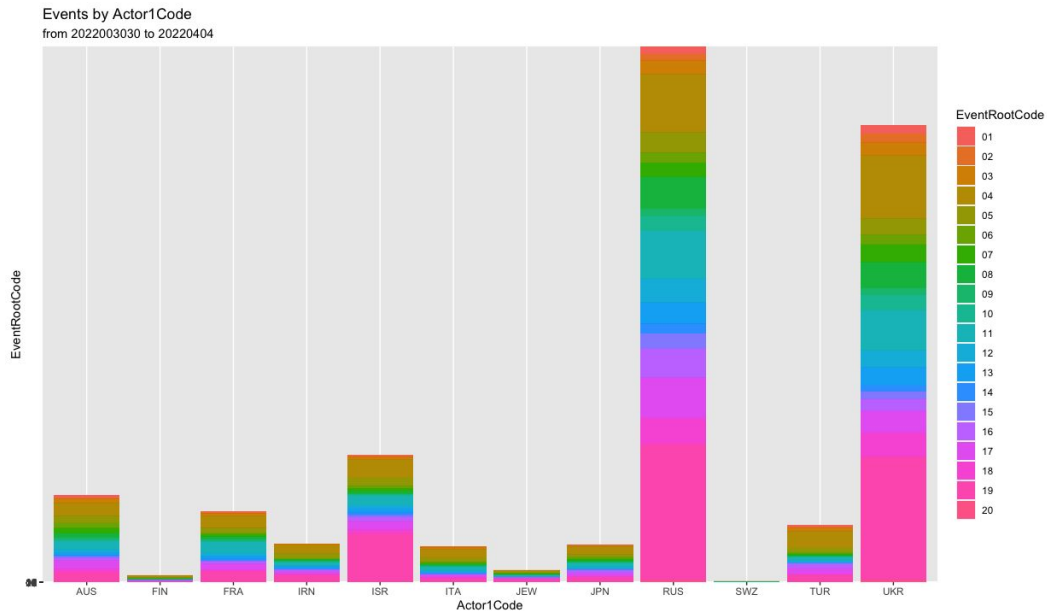
Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
------	---------	--------	------	---------	------

20120331	20220329	20220401	20220292	20220403	20220404
----------	----------	----------	----------	----------	----------

# How much data ?

```
> nrow(mongodata)
```

[1] 69165





# What was that command again?

```
mongodata %>%  
  group_by(Actor1Code,EventRootCode) %>%  
  ggplot( aes( x=Actor1Code,fill=EventRootCode, y=EventRootCode)) +  
  geom_col() +  
  labs( title="Events by Actor1Code",  
        subtitle="from 2022003030 to 20220404")
```

R is powerful but ggplot2 is a learning investment

# Atlas Charts also powerful but also really easy

