Refugee Movement Around the World

DataSci 205 - Project 3

Steve Lanciotti, Nura Hossainzadeh, Bianka Paul

Table of contents

Scenario Description

Refugee Movement Around the World Analysis

Data: Sourced from the United Nations High Commissioner for Refugees (UNHCR), via GitHub data package.

Description: The data covers forcibly displaced populations, including refugees, asylum-seekers, internally displaced people, stateless people, and others between 2010 and 2022.

Analysis: Determine the most influential countries for creating refugees and for offering asylum, as well as observe the flow of refugees across countries.

International Displacement

+200% Increase Over 12 Years

11,384,794
2010

11,438,475
2012

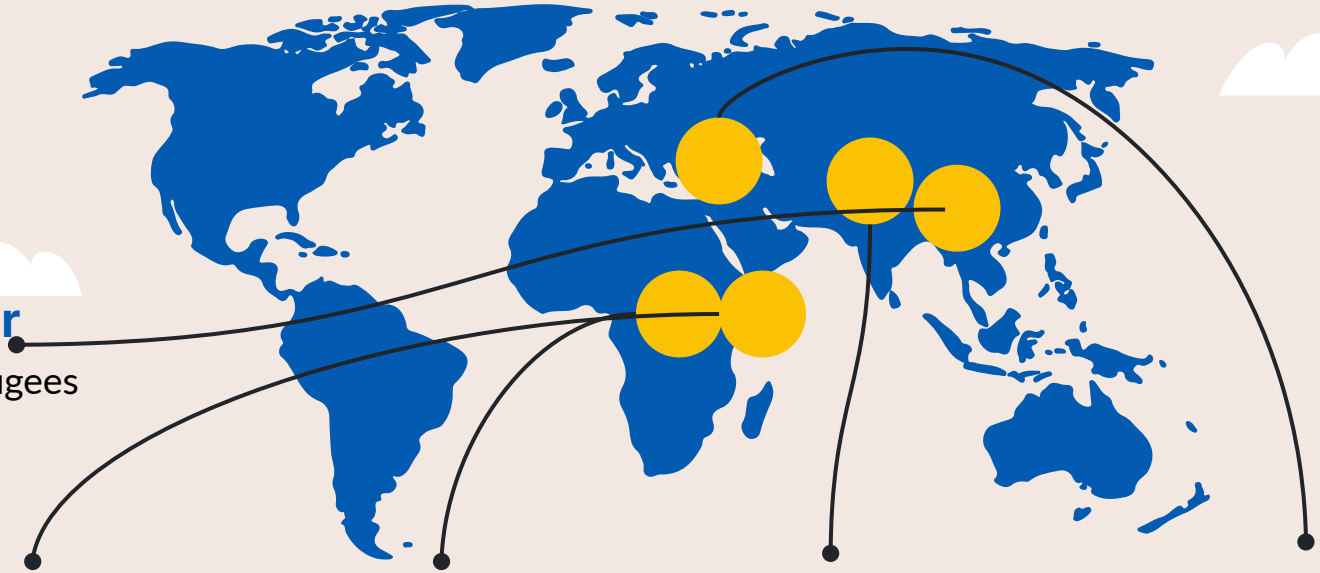
19,333,736
2015

23,030,073
2017

24,846,767
2020

34,851,942
2022

Top 5 Refugee-Producing Countries (2010-2022)



Myanmar
10,585,463 refugees

Somalia
13,191,897 refugees

South Sudan
16,942,541 refugees

Afghanistan
40,246,709 refugees

Syrian Arab Republic
58,610,385 refugees

Graph 1 - Emigration

Data Preprocessing

- Dataset included refugees, asylum seekers, returned refugees, internally displaced, among others.
  - To account for the different types of refugees tracked in the data set, we created a total refugee column which summed refugees and asylum seekers.
- Instances of “internally displaced” were removed from the graph since there was no outbound movement to track.

Country of Origin to Country of Asylum

Design:

- Nodes: Country of Origin, Country of Asylum
- Relationships: (Country of Origin) → (Country of Asylum)
  - Weights: total number of refugees (refugees plus asylum seekers)
  - Additional Attribute: Year of Migration

Use Case:

This graph would be used to analyze the flow of refugees to countries of asylum.

	node_name	labels
0	Afghanistan	[Country]
1	Albania	[Country]
2	Algeria	[Country]
3	Andorra	[Country]
4	Angola	[Country]
...	...	...
207	Viet Nam	[Country]
208	Western Sahara	[Country]
209	Yemen	[Country]
210	Zambia	[Country]
211	Zimbabwe	[Country]

212 rows × 2 columns

```
--------------------------
  Relationships:
--------------------------
```

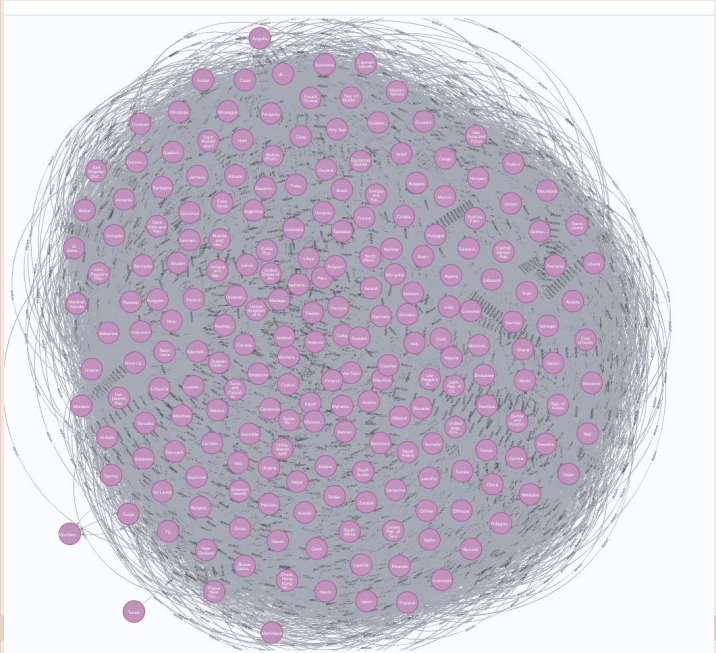
	node_name_1	node_1_labels	relationship_type	year	node_name_2	node_2_labels
0	Afghanistan	[Country]	asylum	2020	Albania	[Country]
1	Afghanistan	[Country]	asylum	2021	Albania	[Country]
2	Afghanistan	[Country]	asylum	2022	Albania	[Country]
3	Afghanistan	[Country]	asylum	2011	Algeria	[Country]
4	Afghanistan	[Country]	asylum	2019	Argentina	[Country]
...	...	...	...	...	...	...
64246	Zimbabwe	[Country]	asylum	2013	Zambia	[Country]
64247	Zimbabwe	[Country]	asylum	2014	Zambia	[Country]
64248	Zimbabwe	[Country]	asylum	2015	Zambia	[Country]
64249	Zimbabwe	[Country]	asylum	2016	Zambia	[Country]
64250	Zimbabwe	[Country]	asylum	2017	Zambia	[Country]

64251 rows × 6 columns

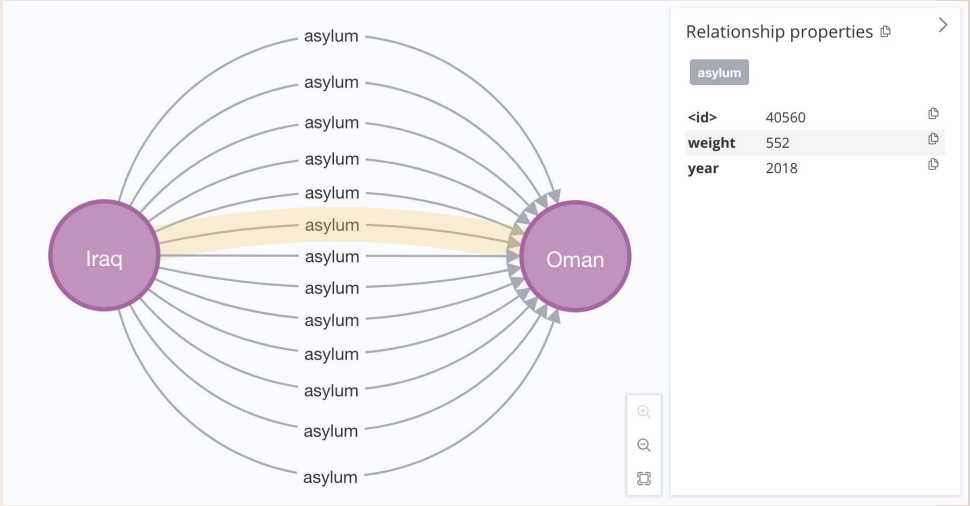
```
--------------------------
  Density: 2.9
--------------------------
```

Graph 1 - Network Graph

Neo4J Graph "Emigration" Bird's Eye View (So Complicated!)



Neo4J Graph"Emigration" Zoomed In (Phew, this makes more sense!)



Graph 1: Algorithm 1

PageRank Algorithm

- Node (country) influence: is a country connected to other influential countries?
- Is an asylum-granter connected to other big asylum-granters?
- What is the “pull” of a country?
- When using a relational database, it’s more difficult to measure the influence (“pull”) of a node (country) by tracing its connection, through the network, to multiple nodes

	name	page_rank
0	United States of America	2.258903
1	Canada	1.879711
2	Germany	1.554083
3	Australia	1.552349
4	United Kingdom of Great Britain and Northern I...	1.474410
...	...	...
207	Stateless	0.950000
208	Tibetan	0.950000
209	Tonga	0.950000
210	Tuvalu	0.950000
211	Western Sahara	0.950000

212 rows × 2 columns

Graph 1: Algorithm 2

Degree Centrality Algorithm

- Number of relationships a node (country) has, both incoming and outgoing
- We can infer that these countries had a high number of outgoing connections (people fleeing the country)
- Again, in a relational database, harder to track this mapping of countries between other countries

Degree Centrality (Most connected countries):

	country	score
0	Syrian Arab Rep.	1341.0
1	Somalia	1332.0
2	Iraq	1268.0
3	Dem. Rep. of the Congo	1256.0
4	Sudan	1215.0
5	Afghanistan	1141.0
6	Iran (Islamic Rep. of)	1079.0
7	Eritrea	1077.0
8	Stateless	1075.0
9	Ethiopia	1056.0

Graph 1: Algorithm 3

Betweenness Centrality Algorithm

- How much of an intermediary–like a bridge–is a country in the graph?
- How frequently is a node (country) between other nodes (countries), based on an all pairs shortest path calculation?
- We had to inverse the weight (refugee count) so the “shortest path” would actually be countries connected by larger weights (higher number of refugees), since we are interested in countries that stand between high refugee flow.
- In a relational database, hard to track betweenness

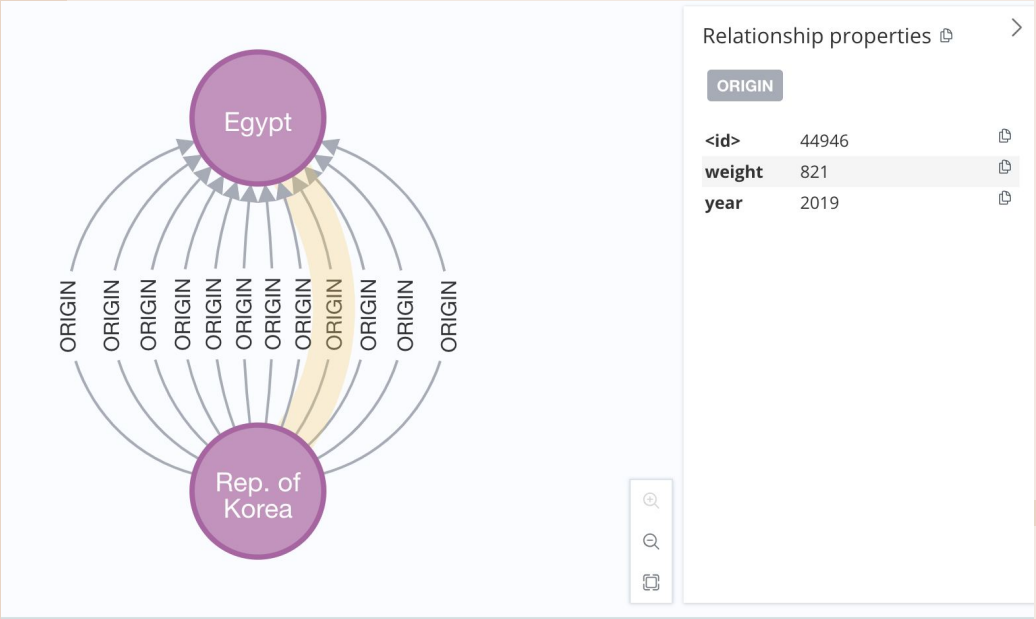
Betweenness Centrality (countries that are strong intermediaries):

	country	score
0	Belgium	2.886156e+10
1	Iraq	2.874825e+10
2	Azerbaijan	2.851470e+10
3	Yemen	2.829025e+10
4	Somalia	2.822219e+10
5	Libya	2.763071e+10
6	Burundi	2.731731e+10
7	Kyrgyzstan	2.597380e+10
8	Tajikistan	2.577197e+10
9	Ethiopia	2.574963e+10

Graph 2: Immigration

Immigration Network Graph

- **Design**: (Country of Asylum) → (Country of Origin)
- **Use Case**: This graph would be used to analyze the flow of refugees from countries of origin.



PageRank Algorithm

- How powerful are nodes (countries) in the graph network in terms of creating refugees?

	name	page_rank
0	Stateless	1.409711
1	Iraq	1.269865
2	Syrian Arab Rep.	1.266019
3	Venezuela (Bolivarian Republic of)	1.245139
4	Sri Lanka	1.235741
...	...	...
207	Bermuda	0.950043
208	Micronesia (Federated States of)	0.950042
209	Aruba	0.950021
210	Montserrat	0.950000
211	Sint Maarten (Dutch part)	0.950000

MongoDB: Use Case

Scenario Use Case

MongoDB excels at dealing with unstructured data and semi-structured data, like JSon data.

- **Storing ID data**
  - Tracking individual refugee profiles (names, ID numbers, medical records, etc..). Geographic and biometric data (fingerprints, facial data, etc..)

Limitations

1. It would not be able to support relationship-based querying, which essentially makes it a bad pick to conduct analysis of relationships between nodes.

2. It also does not do well with supporting graph algorithms such as PageRank and Centrality. This is a problem because these algorithms help us to understand flows in network data.

Redis: Use Case

Scenario Use Case

Redis performs well when it comes to storing key values due to high performance in memory.

- **Travel tracking, monitoring, & alerts**
  - We could use Redis to power a system that can alert when refugees exceed a set number of border crossings.

Limitations

Redis is not usually suitable for structured and queryable data (e.g., graph data)

- Which is why it usually lacks built-in support when it comes to exploring the structure of graphs, like the capacity to apply the Centrality algorithm and other algorithms.

Key Takeaway

- Our main goal was to analyze refugee movement patterns between countries without tracking live updates.
- By applying PageRank and Centrality algorithms, we were able to discern:
  - which countries **attracted** refugees,
  - which **produced** refugees
  - which were **passageways and facilitators** of refugee flow.
- Neo4j was the best at supporting our goal because it is good at graph-based analysis, like gauging centrality and flow paths.
- MongoDB and Redis would have lacked the ability to model and explore the relationships effectively.

Thanks.

CREDITS: This presentation template was created by **Slidesgo**, and includes icons by **Flaticon**, and infographics & images by **Freepik**