Filters

Entity type

Ship × Entity × Individual ×

Regime

ISIL (Da'esh) and… × Afghanistan × Counter-Terroris… × Iraq × Somalia × Democratic Rep… × Sudan × Belarus × Iran (Nuclear) × Iran × Zimbabwe × Democratic Peo… × Guinea × Libya × Syria × Guinea-Bissau × Counter-Terroris… × Russia × Global Human Ri… × Central African R… × South Sudan × Yemen × Venezuela × Myanmar × Chemical Weapons × Nicaragua × Cyber × Global Anti-Corr… × Bosnia and Herz… × Mali × Haiti ×

UK Sanctions List Explorer 1

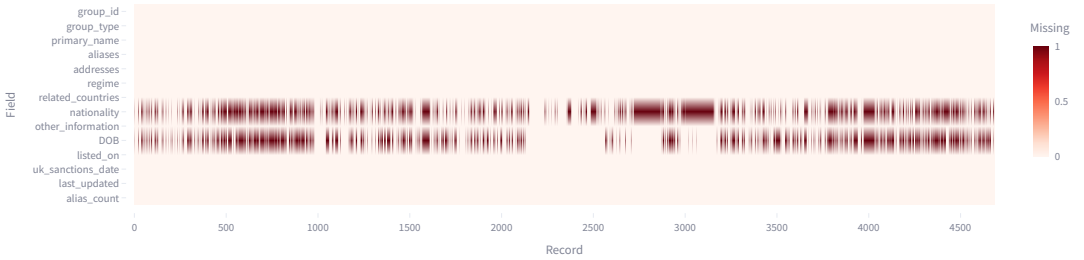
Data Quality Dashboard

Dataset Overview

Total Records	Unique Entities	Complete Records	Overall Completeness
4686	4683	2162	94.1%
	↑ 99.9% unique		

Missing Data Heatmap

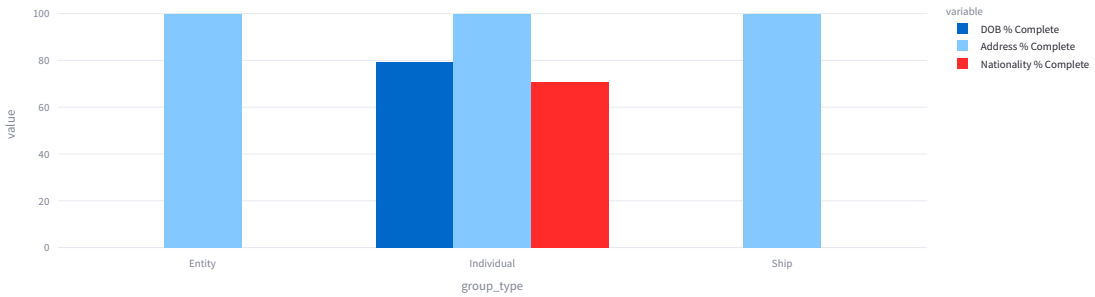
Field Completeness Across Records



Field Completeness

By Field | By Entity Type

Completeness by Entity Type



Potential Duplicates Analysis

Found 6 potential duplicates by name

Inspect potential duplicates:

Bortnikov Aleksandr Vasilievich

Records
2

Unique DOBs
1

Nationalities
1

	group_type	DOB	nationality	regime	listed_on
1087	Individual	1951-11-15 00:00:00	Russia	Russia	2014-07-25 00:00:00
1895	Individual	1951-11-15 00:00:00	Russia	Chemical Weapons	2020-10-15 00:00:00

Field Matching:

DOB: Yes

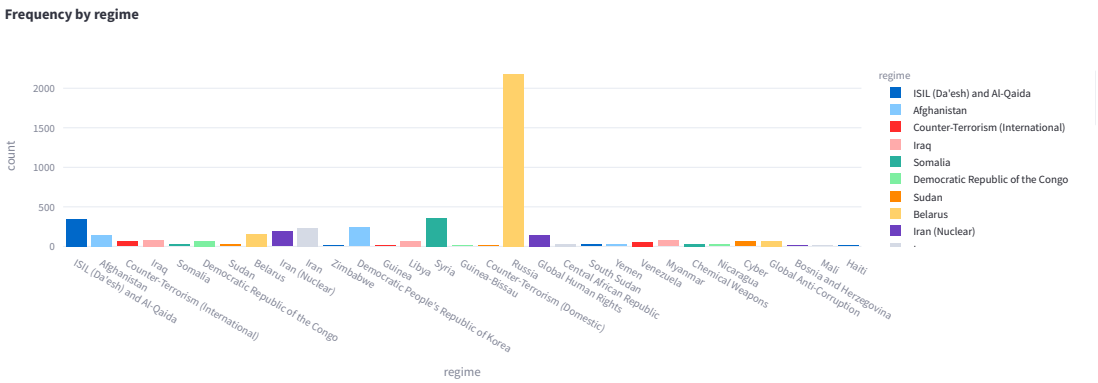
Nationality: Yes

Regime: No

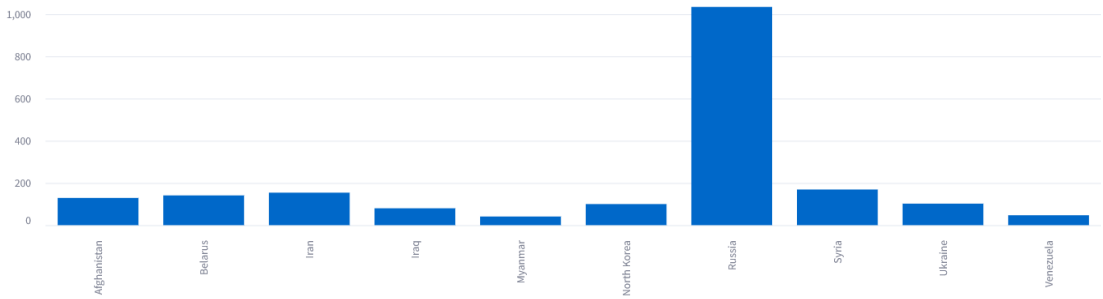
Completeness Over Time

Field Completeness Over Time

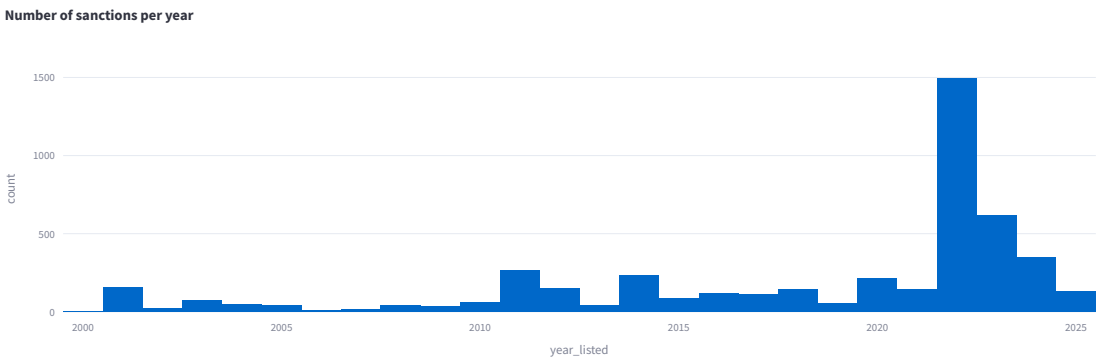
Top regimes



Top nationalities



Sanction timeline



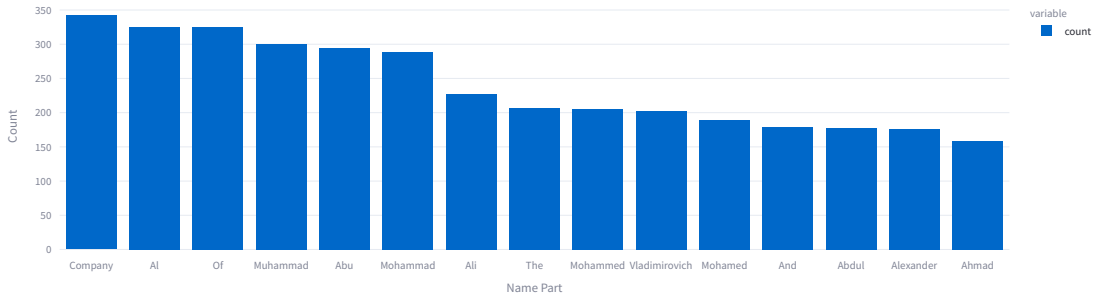
Alias Analysis Center

Distribution Frequency Patterns Lookup

Name Pattern Detection

Common Name Components

Most Frequent Name Parts



Find Similar Aliases

Select an alias to analyze:

Adam

Aliases containing parts of 'Adam':

	count
Adam	3
Eaman Adam Ramsey	1
Adam Danial	1
Adam Hassaan Hussein	1
Adam Abu	1
Achekzai Adam Khan	1
Khan Adam	1
Abbas Adam Del Toro	1
Abdallah Mohamed Adam Brem	1
Adam Nureldine	1

NLP Section

nlp = spacy.load("en_core_web_sm")

def extract_entities(text): doc = nlp(text) return [(ent.text, ent.label_) for ent in doc.ents if ent.label_ in ["PERSON", "ORG", "GPE"]]

st.header("NLP Insights")

#Relations etwee entities st.subheader("Knowledge Graph: Relationships between Entities")

100 text

sampled_infos = filtered_df['other_information'].dropna().sample(n=100, random_state=42) docs = list(nlp.pipe(sampled_infos, batch_size=20))

Graph

G = nx.Graph() for doc in docs: entities = [(ent.text, ent.label_) for ent in doc.ents if ent.label_ in ["PERSON", "ORG", "GPE"]] for i in range(len(entities)): for j in range(i+1, len(entities)): G.add_edge(entities[i][0], entities[j][0])

Limit

if len(G.nodes) > 100: G = G.subgraph(list(G.nodes)[:100])

plt.figure(figsize=(12, 8)) nx.draw(G, with_labels=True, node_color='skyblue', edge_color='gray', node_size=1500, font_size=10) st.pyplot(plt)

Word Cloud

st.subheader("Word Cloud from Other Information") text_data = " ".join(filtered_df['other_information'].dropna().tolist()) wordcloud = WordCloud(width=800, height=400, background_color='white').generate(text_data) st.image(wordcloud.to_array(), use_column_width=True)

Frequency Analysis

st.subheader("Frequent Terms in Descriptions") vectorizer = CountVectorizer(stop_words='english', max_features=30) X = vectorizer.fit_transform(filtered_df['other_information'].dropna()) frequencies = pd.Series(X.toarray().sum(axis=0), index=vectorizer.get_feature_names_out()) st.bar_chart(frequencies.sort_values(ascending=False))

Sample Entity Extraction

st.subheader("Sample Named Entity Recognition Examples") sample_texts = filtered_df["other_information"].dropna().sample(5, random_state=42).tolist() for idx, text in enumerate(sample_texts): doc = nlp(text) st.markdown(f"### Example {idx+1}") st.write(f"**Original text:** {text}") if doc.ents: ents_data = [(ent.text, ent.label_) for ent in doc.ents] ents_df = pd.DataFrame(ents_data, columns=["Entity", "Label"]) st.dataframe(ents_df) else: st.info("No named entities found in this sample.")