POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Systems and Methods for Big and Unstructured Data Project

Author(s): **Jiaxiang Yi 10765316**

Yang Hao Mao 10705881

Group Number: **19**

Academic Year: 2024-2025

Contents

1 | Introduction

1.1. Objective

The project involves using a publicly accessible dataset and analyzing it with a NoSQL technology studied in the course, specifically **Neo4j**, **MongoDB**, or **Elasticsearch**. The primary goal is to demonstrate how to ingest, store, and query real-world data with twenty specific queries, extracting meaningful insights while showcasing the database's querying capabilities. By implementing and applying these queries, the project highlights how modern NoSQL technologies can support data-driven analysis in the health domain.

1.2. Problem Specification

The dataset, titled **"Global Health Statistics"**, covers diverse health metrics such as disease rates, healthcare expenditures, and life expectancies across various countries and timelines. The problem lies in efficiently accessing and searching these metrics in a way that accommodates complex queries, for example, filtering by specific diseases, regions, or time ranges to reveal patterns and trends. Storing and querying the data in a NoSQL database allows for flexible schema design, fast data retrieval, and robust scalability, all of which are vital for timely health analytics and decision-making.

Reference link of dataset: www.kaggle.com/datasets/malaiarasugraj/global-health-statistics

1.3. DBMS Technology Selection

Elasticsearch is selected for this project because of its powerful full-text search capabilities, scalability, and near real-time data processing. It excels at handling both structured and unstructured data, making it well-suited for an interdisciplinary dataset that includes numerical metrics (e.g., mortality rates, healthcare costs) and textual fields (e.g., disease names, country names). Elasticsearch also provides efficient indexing for complex queries and aggregations, enabling interactive, real-time exploration of large health datasets.

Kibana, a companion tool for Elasticsearch, will be used for data visualization and dashboard creation. Kibana's visual interface allows users to quickly build charts, maps, and tables, offering interactive insights into global health trends and facilitating the discovery of patterns and correlations across the dataset.

By leveraging **Elasticsearch** and **Kibana**, we aim to create a workable proof of concept that demonstrates how global health metrics can be quickly ingested, queried, and visualized. This project will focus on practical queries and basic dashboards, serving as a learning exercise in applying NoSQL tools to real-world health data.

2 | Data Wrangling

We loaded the CSV (Global Health Statistics.csv) into a Python notebook (Data Wrangling.ipynb) environment. After inspecting the dataset (displaying the first few rows, checking column data types, and verifying missing values), we found that:

- No missing values existed in any of the columns.
- All fields had appropriate data types (e.g., numeric or object/string).
- The columns were already in a suitable format for analysis, requiring no further cleaning or wrangling steps.

Since the dataset was already clean, no data wrangling (such as imputations, dropping duplicates, or transforming columns) was performed. No additional synthetic data was generated. We simply proceeded with exploratory data analysis on the dataset as-is.

3 | Dataset

3.1. Global Health Statistics dataset

The Dataset used for the project is named "Global Health Statistics.csv" has 1'000'000 rows, 22 columns and it has been downloaded from Kaggle in following link: www.kaggle.com/datasets/malaiarasugraj/global-health-statistics

3.2. Fields Overview

The dataset is structured with following fields, each contributing to a comprehensive overview of the channels:

0. **Field (data type)**: Description

1. **Country (string)**: The name of the country where the health data was recorded.
2. **Year (int)**: The year in which the data was collected.
3. **Disease Name (string)**: The name of the disease or health condition tracked.
4. **Disease Category (string)**: The category of the disease.
5. **Prevalence Rate(%) (float)**:The percentage of the population affected by the disease.
6. **Incidence Rate(%) (float)**: The percentage of new or newly diagnosed cases.
7. **Moratality Rate(%) (float)**: The percentage of the affected population that dies from the disease.
8. **Age Group (string)**: The age range most affected by the disease.
9. **Gender (string)**: The gender(s) affected by the disease (Male, Female, Both).
10. **Population Affected (int)**: The total number of individuals affected by the disease.

11. **Healthcare Access(%) (float)**: The percentage of the population with access to healthcare.
12. **Doctors per 1000 (float)**: The number of doctors per 1000 people.
13. **Hospital Beds per 1000 (float)**: The number of hospital beds available per 1000 people.
14. **Treatment Type (string)**: The primary treatment method for the disease (e.g., Medication, Surgery).
15. **Average Treatment Cost (USD) (int)**: The average cost of treating the disease in USD.
16. **Availability of Vaccines/Treatment (string)**: Whether vaccines or treatments are available.
17. **Recovery Rate(%) (float)**: The percentage of people who recover from the disease.
18. **DALYs (int)**: Disability-Adjusted Life Years, a measure of disease burden.
19. **Improvement in 5 Years(%) (float)**: The improvement in disease outcomes over the last five years.
20. **Per Capita Income (USD) (int)**: The average income per person in the country.
21. **Education Index (float)**: The average level of education in the country.
22. **Urbanization Rate(%) (float)**: The percentage of the population living in urban areas.

3.3. Toward to NoSQL

Unlike relational databases, NoSQL databases as Elasticsearch do not require a fixed schema. This flexibility allows for the addition of new data types or fields as the dataset evolves. The data was indexed in Elasticsearch with customized mappings for each field to optimize both search and analysis capabilities.

Fields representing categorical data, like **Country**, **Disease Name**, **Disease Category**, **Age Group**,**Gender**,**Treatment Type** and **Availability of Vaccines/Treatmen** were indexed as keywords to enable precise, exact match searches. Numerical values, critical for statistical computations, were stored as integers or doubles (float64) depending on their

need for precision. This strategic mapping ensures not only efficient query performance but also delivers highly relevant search results.

For visual reference, a screenshot of the Elasticsearch index mapping is provided in the delivery folder, named "mapping.txt." This file offers a clear depiction of the data schema utilized in this project, highlighting the detailed organization and indexing strategies employed. Below, a small excerpt from the mapping is displayed to illustrate these points:

```
{
  "globalhealth19": {
    "aliases": {},
    "mappings": {
      "_meta": {
        "created_by": "file-data-visualizer"
      },
      "properties": {
        "Age Group": {
          "type": "keyword"
        },
        "Availability of Vaccines/Treatment": {
          "type": "keyword"
        },
        "Average Treatment Cost (USD)": {
          "type": "long"
        },
        "Country": {
          "type": "keyword"
        },
        "DALYs": {
          "type": "long"
        },
        "Disease Category": {
          "type": "keyword"
        },
        "Disease Name": {
          "type": "keyword"
        },
```

Figure 3.1: small portion of mapping.txt

4 | Queries

4.1. Filter documents with a specific Country

In real-world scenarios, analysts often need to focus on data from a particular country to study localized health outcomes or disease prevalence. For instance, retrieving documents where Country is "Italy" helps you analyze patterns specific to Italy, such as the Mortality Rate (%) in that region.

4.1.1. query

```
GET globalhealth19/_search
{
  "query": {
    "match": {
      "Country": "Italy"
    }
  }
}
```

Figure 4.1: query 4.1

4.1.2. output(partial)

```
{
  "took": 3,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 10000,
      "relation": "gte"
    },
    "max_score": 3.008238,
    "hits": [
      {
        "_index": "globalhealth19",
        "_id": "nJoxsZMBhWjrNWL3UHVR",
        "_score": 3.008238,
        "_source": {
          "Disease Category": "Autoimmune",
          "Incidence Rate (%)": 4.86,
          "Mortality Rate (%)": 0.72,
          "Doctors per 1000": 4.1,
          "Education Index": 0.46,
          "Healthcare Access (%)": 51.72,
          "Recovery Rate (%)": 84.52,
          "Treatment Type": "Medication",
          "Improvement in 5 Years (%)": 3.97,
          "Gender": "Other",
          "Population Affected": 616894,
          "Per Capita Income (USD)": 80067,
          "Hospital Beds per 1000": 3.2,
          "Urbanization Rate (%)": 52.81,
          "Year": 2023,
          "Prevalence Rate (%)": 0.43,
          "Disease Name": "Cancer",
          "DALYs": 4909,
          "Availability of Vaccines/Treatment": "Yes",
          "Country": "Italy",
          "Age Group": "0-18",
          "Average Treatment Cost (USD)": 42869
        }
      },
```


Figure 4.2: output(partial) of query 4.1

4.2. Retrieve documents within a certain time range (Year)

Health analysts frequently investigate a specific time window (e.g., 2015–2020) to understand disease trends within that interval. This query uses a range filter on the numeric Year field.

4.2.1. query

```
GET globalhealth19/_search
{
  "query": {
    "range": {
      "Year": {
        "gte": 2020,
        "lte": 2022
      }
    }
  }
}
```

Figure 4.3: query 4.2

4.2.2. output (partial)

```
{
  "took": 652,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 10000,
      "relation": "gte"
    },
    "max_score": 1,
    "hits": [
      {
        "_index": "globalhealth19",
        "_id": "0ZEvsZMBhWjrNWL3RpuN",
        "_score": 1,
        "_source": {
          "Disease Category": "Parasitic",
          "Incidence Rate (%)": 10.49,
          "Mortality Rate (%)": 9.51,
          "Doctors per 1000": 3.5,
          "Education Index": 0.73,
          "Healthcare Access (%)": 96.94,
          "Recovery Rate (%)": 59.59,
          "Treatment Type": "Vaccination",
          "Improvement in 5 Years (%)": 1.87,
          "Gender": "Male",
          "Population Affected": 640330,
          "Per Capita Income (USD)": 22233,
          "Hospital Beds per 1000": 1.96,
          "Urbanization Rate (%)": 69.02,
          "Year": 2020,
          "Prevalence Rate (%)": 16.49,
          "Disease Name": "Measles",
          "DALYs": 3237,
          "Availability of Vaccines/Treatment": "No",
          "Country": "Nigeria",
          "Age Group": "61+",
          "Average Treatment Cost (USD)": 28406
        }
      },
```


Figure 4.4: output(partial) of query 4.2

4.3. Retrieve data for a specific demographic and disease category

This query shows how you might isolate data for a specific demographic (e.g., Gender = "Female") combined with a disease category (e.g., "Infectious"). Real-world usage includes analyzing how certain diseases disproportionately affect certain genders within a category like “Infectious.”

4.3.1. query

```
GET globalhealth19/_search
{
  "query": {
    "bool": {
      "must": [
        { "term": { "Gender": "Female" } },
        { "term": { "Disease Category": "Infectious" } }
      ]
    }
  }
}
```

Figure 4.5: query 4.3

4.3.2. output(partial)

```
{
  "took": 164,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 10000,
      "relation": "gte"
    },
    "max_score": 3.5129046,
    "hits": [
      {
        "_index": "globalhealth19",
        "_id": "yJYwsZMBhWjrNWL3aH_l",
        "_score": 3.5129046,
        "_source": {
          "Disease Category": "Infectious",
          "Incidence Rate (%)": 4.56,
          "Mortality Rate (%)": 9.7,
          "Doctors per 1000": 4.94,
          "Education Index": 0.84,
          "Healthcare Access (%)": 55.64,
          "Recovery Rate (%)": 80.37,
          "Treatment Type": "Surgery",
          "Improvement in 5 Years (%)": 0.02,
          "Gender": "Female",
          "Population Affected": 691815,
          "Per Capita Income (USD)": 72989,
          "Hospital Beds per 1000": 3.82,
          "Urbanization Rate (%)": 64.8,
          "Year": 2010,
          "Prevalence Rate (%)": 6.79,
          "Disease Name": "Dengue",
          "DALYs": 1586,
          "Availability of Vaccines/Treatment": "Yes",
          "Country": "Germany",
          "Age Group": "36-60",
          "Average Treatment Cost (USD)": 24856
        }
      },
```


Figure 4.6: output(partial) of query 4.3

4.4. Identify the top N countries by document count

This aggregation helps identify which countries dominate the dataset. Often used to quickly see the top 5 or 10 countries based on the number of records, possibly indicating higher incidence or broader coverage in your data.

4.4.1. query

```
GET globalhealth19/_search
{
  "size": 0,
  "aggs": {
    "top_countries": {
      "terms": {
        "field": "Country",
        "size": 5
      }
    }
  }
}
```

Figure 4.7: query 4.4

4.4.2. output

```
{
  "took": 212,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 10000,
      "relation": "gte"
    },
    "max_score": null,
    "hits": []
  },
  "aggregations": {
    "top_countries": {
      "doc_count_error_upper_bound": 49531,
      "sum_other_doc_count": 748578,
      "buckets": [
        {
          "key": "Russia",
          "doc_count": 50532
        },
        {
          "key": "South Africa",
          "doc_count": 50408
        },
        {
          "key": "South Korea",
          "doc_count": 50181
        },
        {
          "key": "Germany",
          "doc_count": 50176
        },
        {
          "key": "UK",
          "doc_count": 50125
        }
      ]
    }
  }
}
```


Figure 4.8: output of query 4.4

4.5. Retrieve documents matching high mortality rate OR high recovery rate

In practice, analysts may want to flag potential outliers: diseases with a mortality rate $\geq$ 5% or a recovery rate $\geq$ 90%. This query captures documents meeting either condition to investigate them further (e.g., exploring contributing factors or success stories).

4.5.1. query

```
55  GET globalhealth19/_search
56  {
57    "query": {
58      "bool": {
59        "should": [
60          { "range": { "Mortality Rate (%)": { "gte": 5 } } },
61          { "range": { "Recovery Rate (%)": { "gte": 90 } } }
62        ],
63        "minimum_should_match": 1
64      }
65    }
66  }
```

Figure 4.9: query 4.5

The `minimum_should_match` ensure must meet at least 1 of the should conditions to be included.

4.5.2. output(partial)

```
{
  "took": 35,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 10000,
      "relation": "gte"
    },
    "max_score": 2,
    "hits": [
      {
        "_index": "globalhealth19",
        "_id": "tpEvsZMBhWjrNWL3RpuN",
        "_score": 2,
        "_source": {
          "Disease Category": "Autoimmune",
          "Incidence Rate (%)": 12.16,
          "Mortality Rate (%)": 8.84,
          "Doctors per 1000": 1.02,
          "Education Index": 0.84,
          "Healthcare Access (%)": 54.99,
          "Recovery Rate (%)": 98.48,
          "Treatment Type": "Surgery",
          "Improvement in 5 Years (%)": 3.02,
          "Gender": "Male",
          "Population Affected": 417787,
          "Per Capita Income (USD)": 50634,
          "Hospital Beds per 1000": 9.85,
          "Urbanization Rate (%)": 31.33,
          "Year": 2000,
          "Prevalence Rate (%)": 3.32,
          "Disease Name": "Malaria",
          "DALYs": 3052,
          "Availability of Vaccines/Treatment": "No",
          "Country": "Nigeria",
          "Age Group": "36-60",
          "Average Treatment Cost (USD)": 49373
        }
      },
```


Figure 4.10: output(partial) of query 4.5

4.6. Search for disease names allowing minor typos

Analysts searching for “Tuberculosis” might misspell it as “tuberculosi.” Fuzzy matching accommodates typographical errors, increasing recall for textual disease names.

4.6.1. query

```
GET globalhealth19/_search
{
  "query": {
    "match": {
      "Disease Name": {
        "query": "tuberculosi",
        "fuzziness": "AUTO"
      }
    }
  }
}
```

Figure 4.11: query 4.6

The `fuzziness:AUTO` helps compensate for potential misspellings

4.6.2. output(partial)

```
{
  "took": 121,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 10000,
      "relation": "gte"
    },
    "max_score": 2.4573126,
    "hits": [
      {
        "_index": "globalhealth19",
        "_id": "8ZIvsZMBhWjrNWL3ga3j",
        "_score": 2.4573126,
        "_source": {
          "Disease Category": "Metabolic",
          "Incidence Rate (%)": 14.07,
          "Mortality Rate (%)": 8.15,
          "Doctors per 1000": 2.02,
          "Education Index": 0.77,
          "Healthcare Access (%)": 95.73,
          "Recovery Rate (%)": 77.03,
          "Treatment Type": "Vaccination",
          "Improvement in 5 Years (%)": 7.9,
          "Gender": "Other",
          "Population Affected": 367660,
          "Per Capita Income (USD)": 68200,
          "Hospital Beds per 1000": 5.44,
          "Urbanization Rate (%)": 62.15,
          "Year": 2001,
          "Prevalence Rate (%)": 13.03,
          "Disease Name": "Tuberculosis",
          "DALYs": 3869,
          "Availability of Vaccines/Treatment": "Yes",
          "Country": "Saudi Arabia",
          "Age Group": "61+",
          "Average Treatment Cost (USD)": 36017
        }
      },
```


Figure 4.12: output(partial) of query 4.6

4.7. Calculate global average of doctors per 1000

Healthcare capacity metrics, like doctors per 1000 people, are crucial. This aggregation calculates an overall average across all documents to give a sense of how well-staffed healthcare systems are in the dataset.

4.7.1. query

```
GET globalhealth19/_search
{
  "size": 0,
  "aggs": {
    "avg_doctors": {
      "avg": {
        "field": "Doctors per 1000"
      }
    }
  }
}
```

Figure 4.13: query 4.7

4.7.2. output

```
{
  "took": 68,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 10000,
      "relation": "gte"
    },
    "max_score": null,
    "hits": []
  },
  "aggregations": {
    "avg_doctors": {
      "value": 2.74792922
    }
  }
}
```


Figure 4.14: output of query 4.7

4.8. Compare vaccine availability categories side by side

It's common to compare how many documents indicate Availability of Vaccines/Treatment = "Yes" vs. "No". This parallel filters aggregation approach helps visualize coverage gaps or success rates in vaccine availability.

4.8.1. query

```
GET globalhealth19/_search
{
  "size": 0,
  "aggs": {
    "vaccine_comparison": {
      "filters": {
        "filters": {
          "has_vaccine": {
            "term": {
              "Availability of Vaccines/Treatment": "Yes"
            }
          },
          "no_vaccine": {
            "term": {
              "Availability of Vaccines/Treatment": "No"
            }
          }
        }
      }
    }
  }
}
```

Figure 4.15: query 4.8

4.8.2. output

```
{
  "took": 1,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 10000,
      "relation": "gte"
    },
    "max_score": null,
    "hits": []
  },
  "aggregations": {
    "vaccine_comparison": {
      "buckets": {
        "has_vaccine": {
          "doc_count": 500354
        },
        "no_vaccine": {
          "doc_count": 499646
        }
      }
    }
  }
}
```


Figure 4.16: output of query 4.8

4.9. Retrieve the latest record for each disease category

Each disease category may span multiple years. Grouping by Disease Category and returning only the doc from the latest Year is a common approach for summarizing the most recent data across categories.

4.9.1. query

```
{
  "size": 0,
  "aggs": {
    "by_category": {
      "terms": {
        "field": "Disease Category",
        "size": 10
      },
      "aggs": {
        "latest_year": {
          "top_hits": {
            "sort": [
              { "Year": { "order": "desc" } }
            ],
            "size": 1
          }
        }
      }
    }
  }
}

```

Figure 4.17: query 4.9

4.9.2. output(partial)

```
"aggregations": {
  "by_category": {
    "doc_count_error_upper_bound": 0,
    "sum_other_doc_count": 90445,
    "buckets": [
      {
        "key": "Metabolic",
        "doc_count": 91332,
        "latest_year": {
          "hits": {
            "total": {
              "value": 91332,
              "relation": "eq"
            },
            "max_score": null,
            "hits": [
              {
                "_index": "globalhealth19",
                "_id": "P5oxsZMBhWjrNWL3UIZT",
                "_score": null,
                "_source": {
                  "Disease Category": "Metabolic",
                  "Incidence Rate (%)": 6.28,
                  "Mortality Rate (%)": 4.95,
                  "Doctors per 1000": 3.2,
                  "Education Index": 0.61,
                  "Healthcare Access (%)": 81.48,
                  "Recovery Rate (%)": 71.49,
                  "Treatment Type": "Vaccination",
                  "Improvement in 5 Years (%)": 3.91,
                  "Gender": "Male",
                  "Population Affected": 642914,
                  "Per Capita Income (USD)": 75213,
                  "Hospital Beds per 1000": 0.88,
                  "Urbanization Rate (%)": 48.67,
                  "Year": 2024,
                  "Prevalence Rate (%)": 6.02,
                  "Disease Name": "Hepatitis",
                  "DALYs": 4104,
                  "Availability of Vaccines/Treatment": "No",
                  "Country": "France",
                  "Age Group": "61+",
                  "Average Treatment Cost (USD)": 44697
                },
                "sort": [
                  2024
                ]
              }
            ]
          }
        }
      },
```


Figure 4.18: output(partial) of query 4.9

4.10. Find the costliest diseases by average treatment cost

Some diseases can be very expensive to treat. This query groups documents by disease name and calculates the average Average Treatment Cost (USD), sorting them in descending order to highlight the top costliest diseases in the dataset.

4.10.1. query

```
GET globalhealth19/_search
{
  "size": 0,
  "aggs": {
    "costliest_diseases": {
      "terms": {
        "field": "Disease Name",
        "size": 5,
        "order": {
          "avg_cost": "desc"
        }
      },
      "aggs": {
        "avg_cost": {
          "avg": {
            "field": "Average Treatment Cost (USD)"
          }
        }
      }
    }
  }
}
```

Figure 4.19: query 4.10

4.10.2. output

```
  "aggregations": {
    "costliest_diseases": {
      "doc_count_error_upper_bound": -1,
      "sum_other_doc_count": 780796,
      "buckets": [
        {
          "key": "Measles",
          "doc_count": 39734,
          "avg_cost": {
            "value": 25195.426863643228
          }
        },
        {
          "key": "Influenza",
          "doc_count": 49919,
          "avg_cost": {
            "value": 25138.26358701096
          }
        },
        {
          "key": "Cancer",
          "doc_count": 40163,
          "avg_cost": {
            "value": 25129.76767173767
          }
        },
        {
          "key": "Hypertension",
          "doc_count": 39418,
          "avg_cost": {
            "value": 25091.27550865087
          }
        },
        {
          "key": "Hepatitis",
          "doc_count": 49970,
          "avg_cost": {
            "value": 25083.49457674605
          }
        }
      ]
    }
  }
}
```


Figure 4.20: output of query 4.10

4.11. Group year data in intervals and keep high-affected populations only

This approach divides the Year field into multi-year buckets (e.g., every 5 years) and computes the average Population Affected. It then filters to retain only the buckets that exceed a certain threshold (e.g., >500 000). Researchers can spot time intervals with particularly large affected populations.

4.11.1. query

```
GET globalhealth19/_search
{
  "size": 0,
  "query": {
    "bool": {
      "must": [
        { "term": { "Disease Category": "Chronic" } }
      ]
    }
  },
  "aggs": {
    "year_hist": {
      "histogram": {
        "field": "Year",
        "interval": 5
      },
      "aggs": {
        "avg_population": {
          "avg": {
            "field": "Population Affected"
          }
        },
        "population_filter": {
          "bucket_selector": {
            "buckets_path": {
              "avgPop": "avg_population"
            },
            "script": "params.avgPop > 1000000"
          }
        }
      }
    }
  }
}
```

Figure 4.21: query 4.11

4.11.2. output(partial)


```
{
  "took": 2,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 10000,
      "relation": "gte"
    },
    "max_score": null,
    "hits": []
  },
  "aggregations": {
    "year_hist": {
      "buckets": [
        {
          "key": 2005,
          "doc_count": 18026,
          "avg_population": {
            "value": 501765.6664817486
          }
        },
        {
          "key": 2015,
          "doc_count": 18195,
          "avg_population": {
            "value": 500894.8512228634
          }
        }
      ]
    }
  }
}
```


Figure 4.22: output(partial) of query 4.11

4.12. Get extended statistics for the population affected by a specific disease

In many analyses, you want more than just an average or sum of the affected population. Extended statistics (e.g., min, max, variance, std dev) provide a deeper look at the distribution of the Population Affected field for a specific disease like "Diabetes". This helps epidemiologists or health data scientists see not only central tendencies but also variability within the affected population.

4.12.1. query

```
GET globalhealth19/_search
{
  "size": 0,
  "query": {
    "term": {
      "Disease Name": "Diabetes"
    }
  },
  "aggs": {
    "population_stats": {
      "extended_stats": {
        "field": "Population Affected"
      }
    }
  }
}
```

Figure 4.23: query 4.12

4.12.2. output

```
{
  "took": 1,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 10000,
      "relation": "gte"
    },
    "max_score": null,
    "hits": []
  },
  "aggregations": {
    "population_stats": {
      "count": 50020,
      "min": 1004,
      "max": 999981,
      "avg": 501187.25407836866,
      "sum": 25069386449,
      "sum_of_squares": 16734459843636860,
      "variance": 83366711072.23279,
      "variance_population": 83366711072.23279,
      "variance_sampling": 83368377773.1079,
      "std_deviation": 288732.94074669207,
      "std_deviation_population": 288732.94074669207,
      "std_deviation_sampling": 288735.8269649056,
      "std_deviation_bounds": {
        "upper": 1078653.1355717527,
        "lower": -76278.62741501548,
        "upper_population": 1078653.1355717527,
        "lower_population": -76278.62741501548,
        "upper_sampling": 1078658.9080081799,
        "lower_sampling": -76284.39985144255
      }
    }
  }
}
```


Figure 4.24: output of query 4.12

4.13. Prioritize documents from wealthier contexts

Some health analyses may start by examining diseases in regions with higher resources first. Here, each document's importance is influenced by its Per Capita Income (USD) for a specific disease category (like Respiratory), so that higher-income entries appear more prominently in the results.

4.13.1. query

```
GET globalhealth19/_search
{
  "query": {
    "function_score": {
      "query": {
        "term": {
          "Disease Category": "Non-communicable"
        }
      },
      "script_score": {
        "script": {
          "source": "doc['Per Capita Income (USD)'].value"
        }
      },
      "boost_mode": "replace"
    }
  }
}
```

Figure 4.25: query 4.13

4.13.2. output(partial)

```
{
  "took": 1343,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 10000,
      "relation": "gte"
    },
    "max_score": 99999,
    "hits": [
      {
        "_index": "globalhealth19",
        "_id": "NJMvsZMBhWjrNWL3rF9V",
        "_score": 99999,
        "_source": {
          "Disease Category": "Respiratory",
          "Incidence Rate (%)": 10.33,
          "Mortality Rate (%)": 6.62,
          "Doctors per 1000": 2.4,
          "Education Index": 0.71,
          "Healthcare Access (%)": 56.34,
          "Recovery Rate (%)": 67.93,
          "Treatment Type": "Surgery",
          "Improvement in 5 Years (%)": 3.99,
          "Gender": "Other",
          "Population Affected": 78699,
          "Per Capita Income (USD)": 99999,
          "Hospital Beds per 1000": 4.8,
          "Urbanization Rate (%)": 61.2,
          "Year": 2022,
          "Prevalence Rate (%)": 16.91,
          "Disease Name": "Parkinson's Disease",
          "DALYs": 2479,
          "Availability of Vaccines/Treatment": "Yes",
          "Country": "Australia",
          "Age Group": "36-60",
          "Average Treatment Cost (USD)": 23134
        }
      },
```


Figure 4.26: output(partial) of query 4.13

4.14. Compare age-group mortality side by side

To investigate how mortality differs across age brackets (like "0-18" vs. "19-35" vs. "36-60" vs. "61+"), we set up separate conditions for each bracket. This yields parallel results and can highlight which diseases are more severe in each age range.

4.14.1. query

```
GET globalhealth19/_search
{
  "size": 0,
  "aggs": {
    "age_comparison": {
      "filters": {
        "filters": {
          "0_to_18": {
            "term": { "Age Group": "0-18" }
          },
          "19_to_35": {
            "term": { "Age Group": "19-35" }
          },
          "36_to_60": {
            "term": { "Age Group": "36-60" }
          },
          "61_plus": {
            "term": { "Age Group": "61+" }
          }
        }
      },
      "aggs": {
        "avg_mortality": {
          "avg": {
            "field": "Mortality Rate (%)"
          }
        }
      }
    }
  }
}
```

Figure 4.27: query 4.14

4.14.2. output


```
"aggregations": {
  "age_comparison": {
    "buckets": {
      "0_to_18": {
        "doc_count": 249605,
        "avg_mortality": {
          "value": 5.049840588129244
        }
      },
      "19_to_35": {
        "doc_count": 251201,
        "avg_mortality": {
          "value": 5.050999956210365
        }
      },
      "36_to_60": {
        "doc_count": 249205,
        "avg_mortality": {
          "value": 5.0527357797797
        }
      },
      "61_plus": {
        "doc_count": 249989,
        "avg_mortality": {
          "value": 5.046102588513895
        }
      }
    }
  }
}
```


Figure 4.28: output of query 4.14

4.15. Find diseases that stand out in a subpopulation

Within a certain subpopulation (e.g., Gender="Male"), some diseases might appear disproportionately often compared to the general dataset. Identifying these outliers provides insight into at-risk groups.

4.15.1. query

```
GET globalhealth19/_search
{
  "query": {
    "term": {
      "Gender": "Male"
    }
  },
  "size": 0,
  "aggs": {
    "significant_diseases": {
      "significant_terms": {
        "field": "Disease Name",
        "size": 5
      }
    }
  }
}
```

Figure 4.29: query 4.15

4.15.2. output

```
"aggregations": {
  "significant_diseases": {
    "doc_count": 333676,
    "bg_count": 1000000,
    "buckets": [
      {
        "key": "Leprosy",
        "doc_count": 16974,
        "score": 0.0008186725862958704,
        "bg_count": 50064
      },
      {
        "key": "Dengue",
        "doc_count": 17004,
        "score": 0.000679556277939564,
        "bg_count": 50289
      },
      {
        "key": "Cancer",
        "doc_count": 16882,
        "score": 0.00031088804524608964,
        "bg_count": 50285
      },
      {
        "key": "Cholera",
        "doc_count": 16867,
        "score": 0.00030182709683728906,
        "bg_count": 50249
      },
      {
        "key": "Diabetes",
        "doc_count": 16783,
        "score": 0.0002788316099633605,
        "bg_count": 50020
      }
    ]
  }
}
```


Figure 4.30: output of query 4.15

4.16. Highlight partial matches in disease names

When searching disease names by substring (e.g. “flu”), it can be helpful to highlight where that substring appears. This is particularly relevant for UI or user-facing search

results.

4.16.1. query

```
#16.Highlight partial matches in disease names
GET globalhealth19/_search
{
  "query": {
    "wildcard": {
      "Disease Name": "*flu*"
    }
  },
  "highlight": {
    "fields": {
      "Disease Name": {}
    }
  }
}
```

Figure 4.31: query 4.16

4.16.2. output(partial)

```
"hits": {
  "total": {
    "value": 10000,
    "relation": "gte"
  },
  "max_score": 1,
  "hits": [
    {
      "_index": "globalhealth19",
      "_id": "xJEvsZMBhWjrNWL3RpyN",
      "_score": 1,
      "_source": {
        "Disease Category": "Autoimmune",
        "Incidence Rate (%)": 9.79,
        "Mortality Rate (%)": 8.9,
        "Doctors per 1000": 2.85,
        "Education Index": 0.81,
        "Healthcare Access (%)": 99.98,
        "Recovery Rate (%)": 86.15,
        "Treatment Type": "Medication",
        "Improvement in 5 Years (%)": 1.58,
        "Gender": "Other",
        "Population Affected": 968551,
        "Per Capita Income (USD)": 18834,
        "Hospital Beds per 1000": 8.7,
        "Urbanization Rate (%)": 28.83,
        "Year": 2004,
        "Prevalence Rate (%)": 5.47,
        "Disease Name": "Influenza",
        "DALYs": 3808,
        "Availability of Vaccines/Treatment": "No",
        "Country": "Argentina",
        "Age Group": "61+",
        "Average Treatment Cost (USD)": 33391
      },
      "highlight": {
        "Disease Name": [
          "<em>Influenza</em>"
        ]
      }
    },
```


Figure 4.32: output(partial) of query 4.16

4.17. Break down data by year to compute average mortality

Here, the Year field is numeric, so we treat each distinct year as a bucket and then calculate the average Mortality Rate (%). This is useful for year-by-year comparisons without a formal date field.

4.17.1. query

```
GET globalhealth19/_search
{
  "size": 0,
  "aggs": {
    "yearly_distribution": {
      "histogram": {
        "field": "Year",
        "interval": 1
      },
      "aggs": {
        "avg_mortality": {
          "avg": {
            "field": "Mortality Rate (%)"
          }
        }
      }
    }
  }
}
```

Figure 4.33: query 4.17

4.17.2. output(partial)

```
{
  "took": 59,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 10000,
      "relation": "gte"
    },
    "max_score": null,
    "hits": []
  },
  "aggregations": {
    "yearly_distribution": {
      "buckets": [
        {
          "key": 2000,
          "doc_count": 40268,
          "avg_mortality": {
            "value": 5.054627992450581
          }
        },
        {
          "key": 2001,
          "doc_count": 39896,
          "avg_mortality": {
            "value": 5.038353719671145
          }
        },
```


Figure 4.34: output(partial) of query 4.17

4.18. Dynamically compute the gap between healthcare access and education

Policy analysts sometimes compare Healthcare Access (%) and the Education Index. By generating a difference on the fly for each document, it becomes easier to see where healthcare is outpacing or lagging behind education levels.

4.18.1. query

```
GET globalhealth19/_search
{
  "_source": ["Country", "Healthcare Access (%)", "Education
    Index"],
  "script_fields": {
    "access_education_gap": {
      "script": {
        "lang": "painless",
        "source": "doc['Healthcare Access (%)'].value - doc
          ['Education Index'].value"
      }
    }
  },
  "query": {
    "match_all": {}
  }
}
```

Figure 4.35: query 4.18

4.18.2. output(partial)

```
{
  "took": 17,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 10000,
      "relation": "gte"
    },
    "max_score": 1,
    "hits": [
      {
        "_index": "globalhealth19",
        "_id": "sJEvsZMBhWjrNWL3RpuN",
        "_score": 1,
        "_source": {
          "Education Index": 0.42,
          "Healthcare Access (%)": 50.99,
          "Country": "South Korea"
        },
        "fields": {
          "access_education_gap": [
            50.57
          ]
        }
      },
```


Figure 4.36: output(partial) of query 4.18

4.19. Smooth data trends over time with a rolling average

Yearly metrics can fluctuate significantly. Applying a rolling average (e.g. 3-year window) to Hospital Beds per 1000 helps reveal long-term trends.

4.19.1. query

```
GET globalhealth19/_search
{
  "size": 0,
  "aggs": {
    "year_buckets": {
      "histogram": {
        "field": "Year",
        "interval": 1
      },
      "aggs": {
        "beds_avg": {
          "avg": {
            "field": "Hospital Beds per 1000"
          }
        },
        "moving_avg_beds": {
          "moving_fn": {
            "buckets_path": "beds_avg",
            "window": 3,
            "script": "MovingFunctions.unweightedAvg(values)"
          }
        }
      }
    }
  }
}
```

Figure 4.37: query 4.19

4.19.2. output(partial)

```
{
  "took": 107,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 10000,
      "relation": "gte"
    },
    "max_score": null,
    "hits": []
  },
  "aggregations": {
    "year_buckets": {
      "buckets": [
        {
          "key": 2000,
          "doc_count": 40268,
          "beds_avg": {
            "value": 5.272970597000099
          },
          "moving_avg_beds": {
            "value": null
          }
        },
```


Figure 4.38: output(partial) of query 4.19

4.20. Filter results by cost or urbanization, then rank by year

Some researchers may want to see diseases where the treatment cost is above a threshold or the urbanization rate is high, while excluding unknown vaccine availability. Once filtered, results are ranked higher if they're from more recent years, making it easier to focus on the latest data.

4.20.1. query

```
GET globalhealth19/_search
{
  "query": {
    "function_score": {
      "query": {
        "bool": {
          "should": [
            {
              "range": { "Average Treatment Cost (USD)": {
                "gt": 2000 } }
            },
            {
              "range": { "Urbanization Rate (%)": { "gt":
                60 } }
            }
          ],
          "must_not": [
            {
              "term": { "Availability of Vaccines/
                Treatment": "Unknown" }
            }
          ],
          "minimum_should_match": 1
        }
      },
      "script_score": {
        "script": {
          "source": "doc['Year'].value"
        }
      },
      "boost_mode": "multiply"
    }
  }
}
```

Figure 4.39: query 4.20

4.20.2. output(partial)


```
{
  "took": 1295,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 10000,
      "relation": "gte"
    },
    "max_score": 4048,
    "hits": [
      {
        "_index": "globalhealth19",
        "_id": "PZEvsZMBhWjrNWL3RqCN",
        "_score": 4048,
        "_source": {
          "Disease Category": "Respiratory",
          "Incidence Rate (%)": 7.77,
          "Mortality Rate (%)": 7,
          "Doctors per 1000": 1.04,
          "Education Index": 0.44,
          "Healthcare Access (%)": 75.12,
          "Recovery Rate (%)": 66.06,
          "Treatment Type": "Vaccination",
          "Improvement in 5 Years (%)": 6.77,
          "Gender": "Female",
          "Population Affected": 526173,
          "Per Capita Income (USD)": 17993,
          "Hospital Beds per 1000": 9.02,
          "Urbanization Rate (%)": 80.42,
          "Year": 2024,
          "Prevalence Rate (%)": 9.29,
          "Disease Name": "COVID-19",
          "DALYs": 904,
          "Availability of Vaccines/Treatment": "No",
          "Country": "China",
          "Age Group": "61+",
          "Average Treatment Cost (USD)": 26968
        }
      },
```


Figure 4.40: output(partial) of query 4.20

5 | Dashboard Description

https://calcio-c70f89.kb.us-east-1.aws.elastic.cloud/app/r/s/ESwdn

5.1. Top 3 Diseases Distribution by Age Group

This chart displays the distribution of the top 3 disease categories for each age group, segmented by 0–18, 19–35, 36–60, and 61+. The inner segments show the proportion of individuals affected, while the outer segments break down the disease categories.

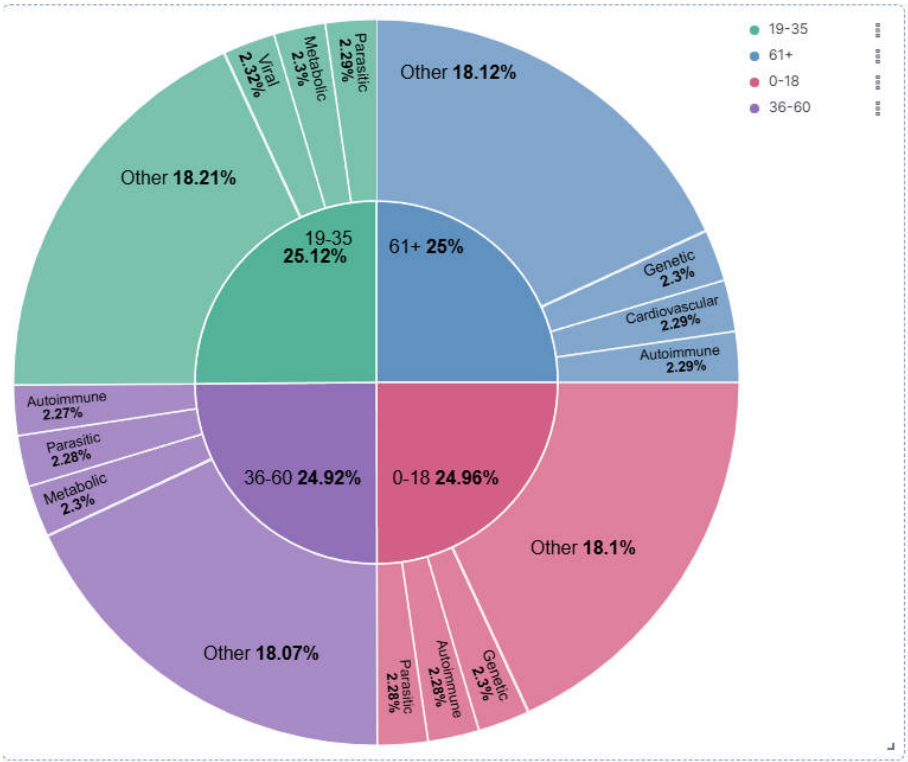

Figure 5.1: Top 3 Diseases Distribution by Age Group

5.2. Top 10 Countries by Alzheimer's Disease Distribution

A treemap chart highlighting the top 10 countries based on the proportion of Alzheimer's disease cases. Each segment represents a country, with the percentage denoting its share.

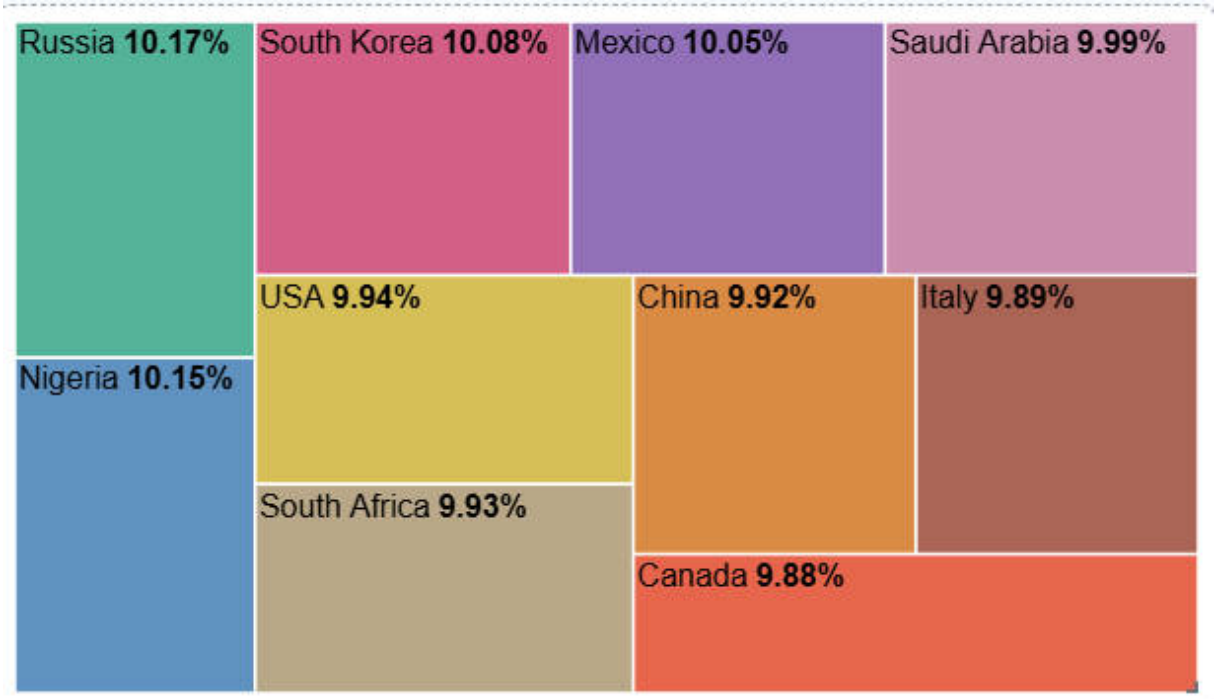

Figure 5.2: Top 10 Countries by Alzheimer's Disease Distribution

5.3. 2000–2024 Count of Records Distribution by Age Group

A line chart showing the trends in record counts across different age groups (0–18, 19–35, 36–60, and 61+) over the years 2000–2024, indicating variations in population affected.

Figure 5.3: 2000–2024 Count of Records Distribution by Age Group

5.4. Top 5 Disease Categories in 5 Countries

A stacked bar chart showcasing the distribution of the top 5 disease categories in five countries. Each bar represents a country, with colored segments denoting the disease categories.

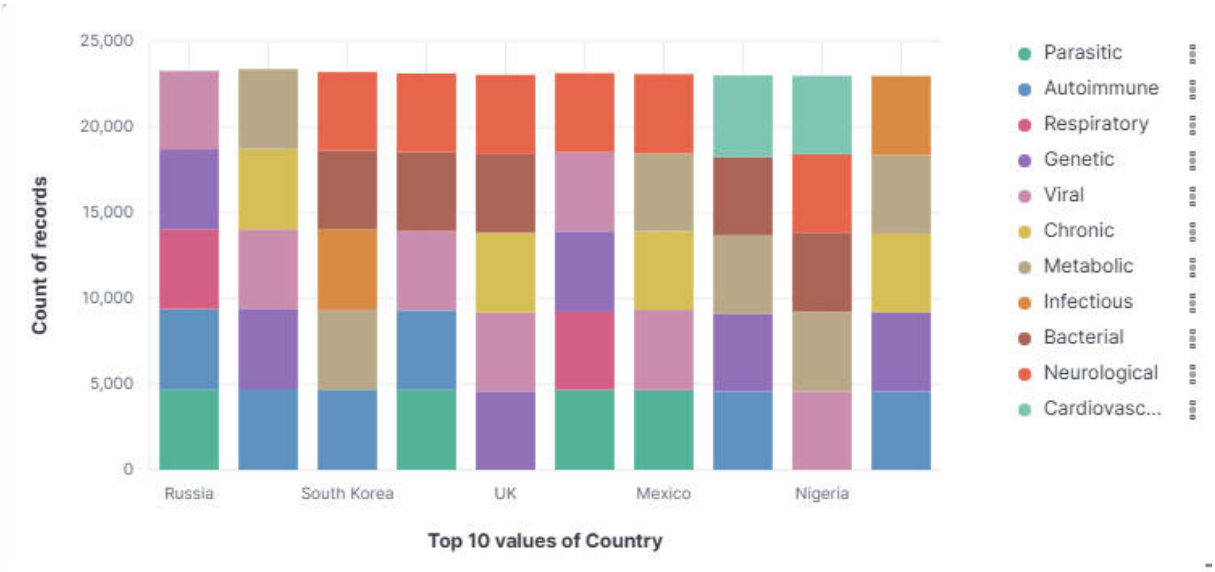

Figure 5.4: Top 5 Disease Categories in 5 Countries

5.5. 2014–2024 Changes in Alzheimer's Disease Counts

A line graph showing the trends in Alzheimer's disease case counts from 2014 to 2024, highlighting changes over time.

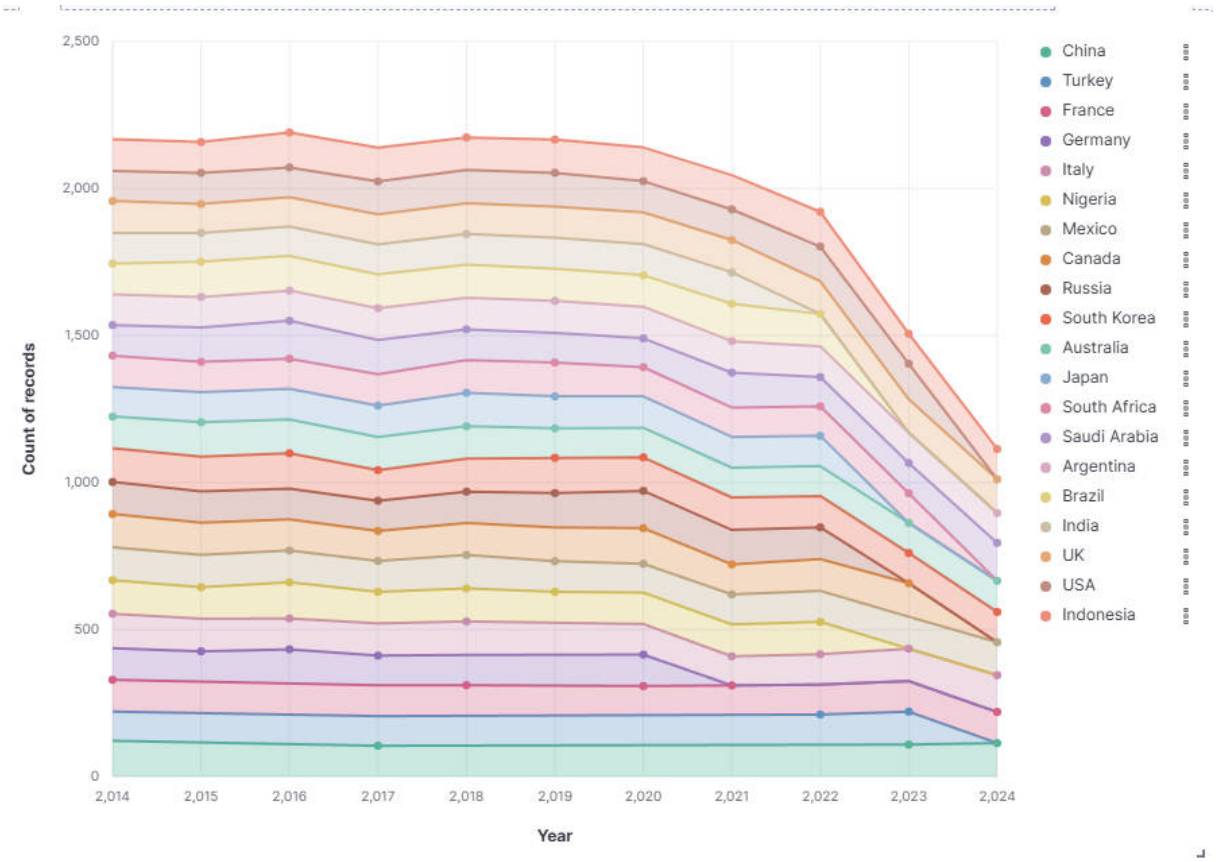

Figure 5.5: 2014–2024 Changes in Alzheimer's Disease Counts

5.6. Top 10 Countries by Minimum Average Treatment Cost (USD)

A word cloud visualizing the top 10 countries with the lowest average treatment costs in USD. The size of the country name correlates with its rank, with larger text indicating lower costs.

Figure 5.6: Top 10 Countries by Minimum Average Treatment Cost (USD)