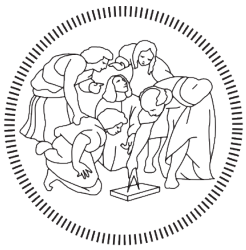
Academic Year 2020/2021



POLITECNICO
MILANO 1863

Computer Science and Engineering

System and Methods for Big and Unstructured Data

Project Report - ElastickStack

Matteo Falzi - 10638723
Flavio La Manna - 10620549
Filippo Manzardo - 10864201
Paolo Marzolo - 10668259
Matteo Regge - 10619213

Prof. Marco Brambilla

January 10, 2022

1 Introduction

In this report, we outlined the design and implementation choices behind the third part of the SMBUD Fall 2021 project at Polimi. The project highlights the potential of ElasticStack, using ElastichSearch as engine and Kibana as data visualizator.

1.1 Document structure

This document will be organized as follows: initially, we will briefly review the delivery specifications and the general objective of this project section. Then, we will provide an overview of the data schema as provided by the instructors, and what we reached with our processing and import in ElasticSearch and Kibana. Following that, we will briefly introduce what it took to include a small additional database, and outline an alternative to large joins more in line with ElasticSearch's philosophy, which will later be used in queries 7 and 8. After the specification of eight queries and two commands (in addition to the "bonus" ones we provide in the aforementioned section), we will conclude with our Kibana dashboard implementation together with some screenshots of the completed product.

1.2 Delivery specification

We need to design, store and query data on a NoSQL DB supporting a data analysis scenario over data about COVID-19 vaccination statistics. The purpose is that of building a comprehensive database of vaccinations, using this database.

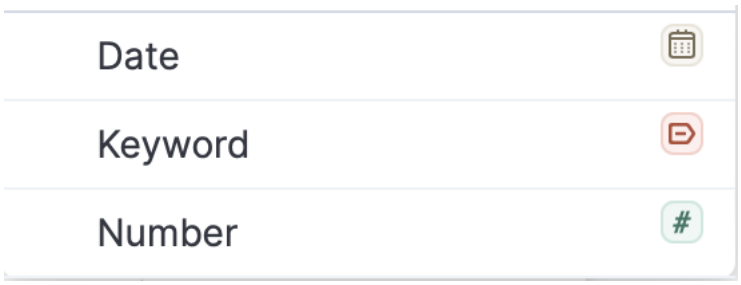
By utilizing an ElasticSearch installation: import the data of the dataset, apply the appropriate schema design choices, and implement some queries aiming at exploring the data statistics and design a visualization dashboard of the results.

1.3 Vaccine Data Schema

The database schema is its structure described in a formal language supported by the database management system. The term "schema" refers to the organization of data as a blueprint of how the database is constructed. When importing data from files, Kibana tries to infer the data type. This is the result of its inference:

Type	Name ↑	Documents (%)	Distinct values	Distributions
	area	999 (100%)	21	top 10 of 21 categories
	codice_NUTS1	999 (100%)	5	5 categories
	codice_NUTS2	999 (100%)	21	top 10 of 21 categories
#	codice_regione_ISTAT	999 (100%)	20	min 1 / median 9 / max 20
	data_somministrazione	999 (100%)	10	
#	dose_addizionale_booster	999 (100%)	1	min 0 / median 0 / max 0
	fascia_anagrafica	999 (100%)	9	9 categories
	fornitore	999 (100%)	3	3 categories
	nome_area	999 (100%)	21	top 10 of 21 categories
#	pregressa_infezione	999 (100%)	7	min 0 / median 0 / max 6
#	prima_dose	999 (100%)	399	min 0 / median 62 / max 2690
#	seconda_dose	999 (100%)	1	min 0 / median 0 / max 0
#	sesso_femminile	999 (100%)	327	min 0 / median 34 / max 1717
#	sesso_maschile	999 (100%)	270	min 0 / median 24 / max 1257
#	vaccini_totali	999 (100%)	399	min 0 / median 62 / max 2690

The Data Type is represented by the left icon, with this correspondence:



Which is correct. An example of the data found in the dataset is the following:

data_somministrazione	fornitore	area	fascia_anagrafica	sesso_maschile	sesso_femminile	prima_dose	seconda_dose	pregressa_infezione	dose_addizionale_booster	codice_NUTS1	codice_NUTS2	codice_regione_ISTAT	nome_area	vaccini_totali
2020-12-27	Pfizer/BioNTech	ABR	20-29	1	1	2	0	0	0	ITF	ITF1	13	Abruzzo	2
2020-12-27	Pfizer/BioNTech	ABR	30-39	1	4	5	0	0	0	ITF	ITF1	13	Abruzzo	5
2020-12-27	Pfizer/BioNTech	ABR	40-49	1	8	9	0	0	0	ITF	ITF1	13	Abruzzo	9
2020-12-27	Pfizer/BioNTech	ABR	50-59	7	6	13	0	0	0	ITF	ITF1	13	Abruzzo	13
2020-12-27	Pfizer/BioNTech	ABR	60-69	12	4	16	0	0	0	ITF	ITF1	13	Abruzzo	16
2020-12-27	Pfizer/BioNTech	ABR	70-79	1	0	1	0	0	0	ITF	ITF1	13	Abruzzo	1
2020-12-27	Pfizer/BioNTech	ABR	80-89	1	0	1	0	0	0	ITF	ITF1	13	Abruzzo	1
2020-12-27	Pfizer/BioNTech	BAS	20-29	4	5	9	0	0	0	ITF	ITF5	17	Basilicata	9
2020-12-27	Pfizer/BioNTech	BAS	30-39	10	18	28	0	0	0	ITF	ITF5	17	Basilicata	28
2020-12-27	Pfizer/BioNTech	BAS	40-49	7	24	31	0	0	0	ITF	ITF5	17	Basilicata	31
2020-12-27	Pfizer/BioNTech	BAS	50-59	11	15	26	0	0	0	ITF	ITF5	17	Basilicata	26
2020-12-27	Pfizer/BioNTech	BAS	60-69	9	4	13	0	0	0	ITF	ITF5	17	Basilicata	13
2020-12-27	Pfizer/BioNTech	CAL	20-29	18	42	60	0	0	0	ITF	ITF6	18	Calabria	60
2020-12-27	Pfizer/BioNTech	CAL	30-39	25	25	50	0	0	0	ITF	ITF6	18	Calabria	50

Finally, here is an excerpt from the analysis completed by Kibana automapping engine:

- Using character encoding [UTF-8], which matched the input with [15%] confidence - first [8kB] of input was pure ASCII.
- Not NDJSON because there was a parsing exception: [...; line: 1, column: 22]].

- Not XML because there was a parsing exception: [ParseError at [row,col]:[1,1] Message: Content is not allowed in prolog.].
- Deciding sample is CSV.
- First row is unusual...
- First sample timestamp match [ISO8601] for field [data_somministrazione].
- Guessing timestamp field is [data_somministrazione] with format [ISO8601].

1.4 Regional Database

ElasticSearch and other, by extension, document-based data storage solutions often aren't designed for long, complex queries. In fact, ElasticSearch does not allow for joins "SQL style" at all. In order to stimulate our curiosity and increase our takeaways, we decided to tackle a situation which, classically, can be most easily modelled in a relational database with joins: the usage of two different databases to retrieve data.
In this case, we started by elaborating a relevant query, which will be reviewed in the next chapter: finding out what percentage of the population of each region has had at least one does of the vaccine. This, obviously, entailed obtaining two pieces of information: the first was the number of first doses that had been administered in every region, and was possible to extract from the original database; the second one was the total population for every region.

1.4.1 Database source and handling

The dataset we used for the second one was downloaded from ISTAT's StatBase, available at this link. Interestingly, it featured a fairly outdated "ITTER107" code, close to the NUTS code included in the original database, but featuring enough discrepancies to make for an interesting debugging session, as we realized with a wikipedia search.

1.4.2 Regional database cleanup

After having obtained the data, it was time to include it as an index on our online ElasticSearch instance (we called it "`population_by_region`") after a simple cleanup script, which we include for completeness:

```
import csv
# import csv file only keep the ones referring to an entire region
# substitute deprecated codes
with open('italian_population.csv') as csvfile:
    reader = csv.reader(csvfile)
    with open('italian_population_by_region.csv', 'w+') as newfile:
        writer = csv.writer(newfile)
        first_row = next(reader)
        first_row[0] = "NUTS2"
        # remove flag trash
        writer.writerow(first_row[:13])
        for row in reader:
            if len(row[0]) == 4 and row[5] == row[7] == row[9] == "totale":
```

```
            writer.writerow(
                [row[0].replace("ITD", "ITH").replace("ITE", "ITI"),
                *row[1:13]])
```

1.4.3 Ingestion pipeline

After this, we tackled how to connect the two together: the most idiomatic solution we found was creating an ingestion pipeline with an enrich processor. This meant that every document run through the pipeline would gain a new `region` field, *enriched* with a population field inside it. We will discuss the advantage of this approach after the implementation.
Since our datasets were ready, there were 5 steps we needed to enrich our existing database.

Creating the ingestion policy

First, we created an ingestion policy, which specifies an interface to the data that we will use to enrich pipelined documents. This is necessary, but also speeds up following operations.

```
PUT /_enrich/policy/region-population-enrich
{
  "match":{
    "indices": [
      "population_by_region"
      ],
      "match_field": "NUTS2",
      "enrich_fields": [
        "Territorio",
        "Value"
        ]

  }
}
```

Executing the policy

```
POST /_enrich/policy/region-population-enrich/_execute
```

Defining a pipeline with enrich processors In this step, we define the pipeline, indicating just an enrich processor which uses the policy we defined above. It should be noted that here the data could be further cleaned up by, e.g., removing the NUTS2 field or renaming the fields embedded with the enrich processor. We avoided it to ensure compatibility with the rest of the dashboard, which we already started developing simultaneously. This is also the reason behind the difference in index naming (queries are mostly completed with the "vaccini" index,

the original database, while here we mention "somministrazioni-vaccini-latest", which has been enriched through the pipeline).

```
PUT _ingest/pipeline/enrich-vaccine-data
{
  "processors":[
    {
      "enrich": {
        "policy_name": "region-population-enrich",
        "field":"codice_NUTS2",
        "target_field": "region"
      }
    }]
}
```

Updating the entire index (selecting all)
A relevant aspect of this approach is that including new documents does not entail any restructuring or join operations: the pipeline works in the same way for multiple or single documents.

```
POST somministrazioni-vaccini-latest/
                    _update_by_query?pipeline=enrich-vaccine-data
```

Refreshing to have dynamic mapping on new fields

```
POST somministrazioni-vaccini-latest/_refresh
```

1.4.4 Final observations

This approach is in line with ElasticSearch's motives and philosophy, and this is highlighted, in our opinion, by the relative ease with which this operation is completed. Once complete, it allows for granular control on what data to enrich, allowing additional data sources and online implementations, and decouples it from the querying part, making the queries run faster than if a lookup was needed run-time. The main drawback is of course that instead of only needing a keyword to memorize the region, a small embedded document is saved, which uses a bit more space.

2 Duties

2.1 Queries

Query n. 1

Count the number of first doses from Pfizer by region

```
GET /vaccini/_search
{ "size": 0,
  "query":{
    "match":{
      "fornitore": "Pfizer/BioNTech"
    }
  },
  "aggs":{
    "region":{
      "terms":{
        "field":"nome_area"
      },
      "aggs":{
        "n_first_doses":{
          "sum":{
            "field": "prima_dose"
          }
        },
        "n_second_doses":{
          "sum":{
            "field": "seconda_dose"
          }
        }
      }
    }
  }
}
```

Response:

```
{
  "took" : 3,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
```

```
  "successful" : 1,
  "skipped" : 0,
  "failed" : 0
},
"hits" : {
  "total" : {
    "value" : 10000,
    "relation" : "gte"
  },
  "max_score" : null,
  "hits" : [ ]
},
"aggregations" : {
  "region" : {
    "doc_count_error_upper_bound" : 0,
    "sum_other_doc_count" : 33610,
    "buckets" : [
      {
        "key" : "Lazio",
        "doc_count" : 3272,
        "n_second_doses" : {
          "value" : 3046195.0
        },
        "n_first_doses" : {
          "value" : 3116503.0
        }
      },
      {
        "key" : "Lombardia",
        "doc_count" : 3269,
        "n_second_doses" : {
          "value" : 5387252.0
        },
        "n_first_doses" : {
          "value" : 5416764.0
        }
      },
      {
        "key" : "Sicilia",
        "doc_count" : 3268,
        "n_second_doses" : {
          "value" : 2628216.0
        },
        "n_first_doses" : {
          "value" : 2702408.0
        }
      },
      {
        "key" : "Toscana",
        "doc_count" : 3262,
        "n_second_doses" : {
          "value" : 2030666.0
        },
        "n_first_doses" : {
          "value" : 2073143.0
        }
      },
      {
        "key" : "Piemonte",
        "doc_count" : 3261,
        "n_second_doses" : {
```

```
            "value" : 2289609.0
          },
          "n_first_doses" : {
            "value" : 2345514.0
          }
        },
        {
          "key" : "Emilia-Romagna",
          "doc_count" : 3250,
          "n_second_doses" : {
            "value" : 2429725.0
          },
          "n_first_doses" : {
            "value" : 2464054.0
          }
        },
        {
          "key" : "Veneto",
          "doc_count" : 3247,
          "n_second_doses" : {
            "value" : 2582719.0
          },
          "n_first_doses" : {
            "value" : 2639864.0
          }
        },
        {
          "key" : "Campania",
          "doc_count" : 3239,
          "n_second_doses" : {
            "value" : 2868193.0
          },
          "n_first_doses" : {
            "value" : 2940366.0
          }
        },
        {
          "key" : "Liguria",
          "doc_count" : 3232,
          "n_second_doses" : {
            "value" : 827018.0
          },
          "n_first_doses" : {
            "value" : 830113.0
          }
        },
        {
          "key" : "Puglia",
          "doc_count" : 3218,
          "n_second_doses" : {
            "value" : 2124441.0
          },
          "n_first_doses" : {
            "value" : 2175610.0
          }
        }
      ]
    }
  }
}
```

Query n.2

Count how many doses have been administered in the age group 50-59 not by Moderna

```
GET /vaccini/_search
{ "size": 0,
  "query":{
    "bool":{
       "must":{
        "term":{
          "fascia_anagrafica":"50-59"
        }
      },
      "must_not":{
        "term":{
          "fornitore":"Moderna"
        }
      }
    }
  },
  "aggs":{
    "number_of_somministrations":{
      "terms":{
        "field":"data_somministrazione"
      }
    }
  }
}
```

Response:


```
{
  "took" : 3,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 10000,
      "relation" : "gte"
    },
```

```
  "max_score" : null,
  "hits" : [ ]
},
"aggregations" : {
  "number_of_somministrations" : {
    "doc_count_error_upper_bound" : 0,
    "sum_other_doc_count" : 13375,
    "buckets" : [
      {
        "key" : 1622160000000,
        "key_as_string" : "2021-05-28T00:00:00.000Z",
        "doc_count" : 63
      },
      {
        "key" : 1621641600000,
        "key_as_string" : "2021-05-22T00:00:00.000Z",
        "doc_count" : 62
      },
      {
        "key" : 1623369600000,
        "key_as_string" : "2021-06-11T00:00:00.000Z",
        "doc_count" : 62
      },
      {
        "key" : 1621555200000,
        "key_as_string" : "2021-05-21T00:00:00.000Z",
        "doc_count" : 61
      },
      {
        "key" : 1622505600000,
        "key_as_string" : "2021-06-01T00:00:00.000Z",
        "doc_count" : 61
      },
      {
        "key" : 1622678400000,
        "key_as_string" : "2021-06-03T00:00:00.000Z",
        "doc_count" : 61
      },
      {
        "key" : 1623110400000,
        "key_as_string" : "2021-06-08T00:00:00.000Z",
        "doc_count" : 61
      },
      {
        "key" : 1623196800000,
        "key_as_string" : "2021-06-09T00:00:00.000Z",
        "doc_count" : 61
      },
      {
        "key" : 1623283200000,
        "key_as_string" : "2021-06-10T00:00:00.000Z",
        "doc_count" : 61
      },
      {
        "key" : 1621468800000,
        "key_as_string" : "2021-05-20T00:00:00.000Z",
        "doc_count" : 60
      }
    ]
  }
}
```

```
}
```

Query n.3

Count the total number of vaccines given to men and women

```
GET /vaccini/_search
{ "size": 0,
  "aggs":{
    "age_group":{
        "terms":{
          "field": "fascia_anagrafica"
        },
      "aggs":{
        "n_men":{
          "sum":{
            "field": "sesso_maschile"
          }
        },
        "n_women":{
          "sum":{
            "field": "sesso_femminile"
          }
        }
      }
    }
  }
}
```

Response:

```
{
  "took" : 1,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 10000,
      "relation" : "gte"
    },
```

```
    "max_score" : null,
    "hits" : [ ]
  },
  "aggregations" : {
    "age_group" : {
      "doc_count_error_upper_bound" : 0,
      "sum_other_doc_count" : 0,
      "buckets" : [
        {
          "key" : "60-69",
          "doc_count" : 21560,
          "n_women" : {
            "value" : 8438621.0
          },
          "n_men" : {
            "value" : 8006667.0
          }
        },
        {
          "key" : "70-79",
          "doc_count" : 20590,
          "n_women" : {
            "value" : 7479885.0
          },
          "n_men" : {
            "value" : 6725759.0
          }
        },
        {
          "key" : "50-59",
          "doc_count" : 20326,
          "n_women" : {
            "value" : 9999951.0
          },
          "n_men" : {
            "value" : 9574883.0
          }
        },
        {
          "key" : "40-49",
          "doc_count" : 20035,
          "n_women" : {
            "value" : 8262284.0
          },
          "n_men" : {
            "value" : 8049574.0
          }
        },
        {
          "key" : "30-39",
          "doc_count" : 19432,
          "n_women" : {
            "value" : 6006066.0
          },
          "n_men" : {
            "value" : 6207028.0
          }
        },
        {
          "key" : "20-29",
          "doc_count" : 18790,
```

```
          "n_women" : {
            "value" : 5387890.0
          },
          "n_men" : {
            "value" : 5801165.0
          }
        },
        {
          "key" : "80-89",
          "doc_count" : 17737,
          "n_women" : {
            "value" : 5633385.0
          },
          "n_men" : {
            "value" : 4018073.0
          }
        },
        {
          "key" : "90+",
          "doc_count" : 14231,
          "n_women" : {
            "value" : 1501413.0
          },
          "n_men" : {
            "value" : 635740.0
          }
        },
        {
          "key" : "12-19",
          "doc_count" : 14099,
          "n_women" : {
            "value" : 3417399.0
          },
          "n_men" : {
            "value" : 3608041.0
          }
        },
        {
          "key" : "05-11",
          "doc_count" : 197,
          "n_women" : {
            "value" : 93717.0
          },
          "n_men" : {
            "value" : 101278.0
          }
        }
      ]
    }
  }
}
```

Query n.4

Count for each region the number of first, second and booster doses of Moderna except for the age group 12-19

```
GET /vaccini/_search
{ "size": 0,
  "query":{
    "bool":{
       "must":{
        "term":{
          "fornitore":"Moderna"
        }
      },
      "must_not":{
        "term":{
          "fascia_anagrafica":"12-19"
        }
      }
    }
  },
   "aggs":{
    "region":{
      "terms":{
        "field":"nome_area"
      },
      "aggs":{
        "n_first_doses":{
          "sum":{
            "field": "prima_dose"
          }
        },
        "n_second_doses":{
          "sum":{
            "field": "seconda_dose"
          }
        },
        "n_booster_doses":{
          "sum":{
            "field": "dose_addizionale_booster"
          }
        }
      }
    }
```

```
  }
}
```

Response:

```
{
  "took" : 15,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 10000,
      "relation" : "gte"
    },
    "max_score" : null,
    "hits" : [ ]
  },
  "aggregations" : {
    "region" : {
      "doc_count_error_upper_bound" : 0,
      "sum_other_doc_count" : 23017,
      "buckets" : [
        {
          "key" : "Campania",
          "doc_count" : 2758,
          "n_second_doses" : {
            "value" : 485937.0
          },
          "n_first_doses" : {
            "value" : 528467.0
          },
          "n_booster_doses" : {
            "value" : 622680.0
          }
        },
        {
          "key" : "Sicilia",
          "doc_count" : 2758,
          "n_second_doses" : {
            "value" : 379164.0
          },
          "n_first_doses" : {
            "value" : 431272.0
          },
          "n_booster_doses" : {
            "value" : 490796.0
          }
        },
        {
          "key" : "Lazio",
          "doc_count" : 2738,
          "n_second_doses" : {
            "value" : 481614.0
          },
```

```
      "n_first_doses" : {
        "value" : 467065.0
      },
      "n_booster_doses" : {
        "value" : 417312.0
      }
    },
    {
      "key" : "Emilia-Romagna",
      "doc_count" : 2646,
      "n_second_doses" : {
        "value" : 419040.0
      },
      "n_first_doses" : {
        "value" : 451458.0
      },
      "n_booster_doses" : {
        "value" : 843835.0
      }
    },
    {
      "key" : "Lombardia",
      "doc_count" : 2622,
      "n_second_doses" : {
        "value" : 974342.0
      },
      "n_first_doses" : {
        "value" : 994254.0
      },
      "n_booster_doses" : {
        "value" : 1750207.0
      }
    },
    {
      "key" : "Veneto",
      "doc_count" : 2605,
      "n_second_doses" : {
        "value" : 437711.0
      },
      "n_first_doses" : {
        "value" : 465367.0
      },
      "n_booster_doses" : {
        "value" : 642058.0
      }
    },
    {
      "key" : "Marche",
      "doc_count" : 2598,
      "n_second_doses" : {
        "value" : 132618.0
      },
      "n_first_doses" : {
        "value" : 140468.0
      },
      "n_booster_doses" : {
        "value" : 175920.0
      }
    },
    {
      "key" : "Calabria",
```

```
        "doc_count" : 2510,
        "n_second_doses" : {
          "value" : 122795.0
        },
        "n_first_doses" : {
          "value" : 134826.0
        },
        "n_booster_doses" : {
          "value" : 161542.0
        }
      },
      {
        "key" : "Toscana",
        "doc_count" : 2484,
        "n_second_doses" : {
          "value" : 406376.0
        },
        "n_first_doses" : {
          "value" : 429816.0
        },
        "n_booster_doses" : {
          "value" : 532881.0
        }
      },
      {
        "key" : "Puglia",
        "doc_count" : 2482,
        "n_second_doses" : {
          "value" : 354539.0
        },
        "n_first_doses" : {
          "value" : 371059.0
        },
        "n_booster_doses" : {
          "value" : 404426.0
        }
      }
    ]
  }
 }
}
```

Query n.5

Given a date, count how many first doses have been done, divided by region.

```
GET /vaccini/_search
{
  "size": 0,
  "query":
  {
    "match":
    {
      "@timestamp": "2021-10-12"
```

```
    }
  },
    "aggs": {
      "by_nome_area": {
        "terms": {
          "field": "nome_area"
        },
        "aggs": {
         "do_a_sum_on_field_prima_dose":  {
            "sum": {
               "field": "prima_dose"
            }
         }
        }
      }
    }
}
```

Response:

```
{
  "took" : 1,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 484,
      "relation" : "eq"
    },
    "max_score" : null,
    "hits" : [ ]
  },
  "aggregations" : {
    "by_nome_area" : {
      "doc_count_error_upper_bound" : 0,
      "sum_other_doc_count" : 215,
      "buckets" : [
        {
          "key" : "Lazio",
          "doc_count" : 30,
          "do_a_sum_on_field_prima_dose" : {
            "value" : 5802.0
          }
        },
        {
          "key" : "Emilia-Romagna",
```

```
  "doc_count" : 29,
  "do_a_sum_on_field_prima_dose" : {
    "value" : 4467.0
  }
},
{
  "key" : "Veneto",
  "doc_count" : 29,
  "do_a_sum_on_field_prima_dose" : {
    "value" : 5736.0
  }
},
{
  "key" : "Lombardia",
  "doc_count" : 28,
  "do_a_sum_on_field_prima_dose" : {
    "value" : 8413.0
  }
},
{
  "key" : "Abruzzo",
  "doc_count" : 27,
  "do_a_sum_on_field_prima_dose" : {
    "value" : 1581.0
  }
},
{
  "key" : "Piemonte",
  "doc_count" : 27,
  "do_a_sum_on_field_prima_dose" : {
    "value" : 5186.0
  }
},
{
  "key" : "Puglia",
  "doc_count" : 26,
  "do_a_sum_on_field_prima_dose" : {
    "value" : 3424.0
  }
},
{
  "key" : "Calabria",
  "doc_count" : 25,
  "do_a_sum_on_field_prima_dose" : {
    "value" : 2979.0
  }
},
{
  "key" : "Marche",
  "doc_count" : 25,
  "do_a_sum_on_field_prima_dose" : {
    "value" : 1857.0
  }
},
{
  "key" : "Campania",
  "doc_count" : 23,
  "do_a_sum_on_field_prima_dose" : {
    "value" : 6591.0
  }
}
```

```
      ]
    }
  }
}
```

Query n.6

Count how many total somministrations have been done in the population, aggregated by age.

```
GET /vaccini/_search
{
  "size": 0,
  "aggs": {
    "by_nome_area": {
      "terms": {
        "field": "fascia_anagrafica"
      },
      "aggs": {
       "all_vaccines":  {
          "sum": {
             "script": "doc['prima_dose'].value +
             doc['seconda_dose'].value +
             doc['dose_addizionale_booster'].value"
          }
       }
      }
    }
  }
}
```

Response:

```
{
  "took" : 278,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 10000,
      "relation" : "gte"
```

```
  },
  "max_score" : null,
  "hits" : [ ]
},
"aggregations" : {
  "by_nome_area" : {
    "doc_count_error_upper_bound" : 0,
    "sum_other_doc_count" : 0,
    "buckets" : [
      {
        "key" : "60-69",
        "doc_count" : 21560,
        "all_vaccines" : {
          "value" : 1.621365E7
        }
      },
      {
        "key" : "70-79",
        "doc_count" : 20590,
        "all_vaccines" : {
          "value" : 1.4047524E7
        }
      },
      {
        "key" : "50-59",
        "doc_count" : 20326,
        "all_vaccines" : {
          "value" : 1.9235695E7
        }
      },
      {
        "key" : "40-49",
        "doc_count" : 20035,
        "all_vaccines" : {
          "value" : 1.6026101E7
        }
      },
      {
        "key" : "30-39",
        "doc_count" : 19432,
        "all_vaccines" : {
          "value" : 1.2004224E7
        }
      },
      {
        "key" : "20-29",
        "doc_count" : 18790,
        "all_vaccines" : {
          "value" : 1.098289E7
        }
      },
      {
        "key" : "80-89",
        "doc_count" : 17737,
        "all_vaccines" : {
          "value" : 9575045.0
        }
      },
      {
        "key" : "90+",
        "doc_count" : 14231,
```

```
        "all_vaccines" : {
          "value" : 2117358.0
        }
      },
      {
        "key" : "12-19",
        "doc_count" : 14099,
        "all_vaccines" : {
          "value" : 6910141.0
        }
      },
      {
        "key" : "05-11",
        "doc_count" : 197,
        "all_vaccines" : {
          "value" : 194803.0
        }
      }
    ]
  }
 }
}
```

Query n.7

Calculate what percentage of the population has received the first dose, divided by region. This does not account for people being vaccinated away from home. Since both values of 'prima_dose' and 'region.population' are considered 'Long', we have to convert them to 'Float' by multiplying in advance by '1.0'.

```
GET somministrazioni-vaccini-latest/_search
{
  "size":0,
  "aggs":{
    "population":{
      "terms":{
        "field":"nome_area"
      },
      "aggs":{
        "first_dose": {
          "sum": {
              "script": "doc['prima_dose'].value /
              (doc['region.population'].value * 1.0) "
            }
        }
      }
    }
  }
```

```
}
```

Response:

```
{
  "took" : 36,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 10000,
      "relation" : "gte"
    },
    "max_score" : null,
    "hits" : [ ]
  },
  "aggregations" : {
    "population" : {
      "doc_count_error_upper_bound" : 0,
      "sum_other_doc_count" : 78090,
      "buckets" : [
        {
          "key" : "Lazio",
          "doc_count" : 10010,
          "first_dose_percentage" : {
            "value" : 8074121.0
          }
        },
        {
          "key" : "Lombardia",
          "doc_count" : 9499,
          "first_dose_percentage" : {
            "value" : 8114575.0
          }
        },
        {
          "key" : "Emilia-Romagna",
          "doc_count" : 9150,
          "first_dose_percentage" : {
            "value" : 7931515.0
          }
        },
        {
          "key" : "Campania",
          "doc_count" : 9053,
          "first_dose_percentage" : {
            "value" : 7631556.0
          }
        },
        {
          "key" : "Sicilia",
          "doc_count" : 8809,
          "first_dose_percentage" : {
            "value" : 7452349.0
```

```
        }
      },
      {
        "key" : "Veneto",
        "doc_count" : 8791,
        "first_dose_percentage" : {
          "value" : 7683254.0
        }
      },
      {
        "key" : "Piemonte",
        "doc_count" : 8768,
        "first_dose_percentage" : {
          "value" : 7789971.0
        }
      },
      {
        "key" : "Calabria",
        "doc_count" : 8649,
        "first_dose_percentage" : {
          "value" : 7572308.0
        }
      },
      {
        "key" : "Marche",
        "doc_count" : 8629,
        "first_dose_percentage" : {
          "value" : 7673145.0
        }
      },
      {
        "key" : "Puglia",
        "doc_count" : 8407,
        "first_dose_percentage" : {
          "value" : 8056354.0
        }
      }
    ]
  }
 }
}
```

Query n.8

Count what percentage of the population of every region has received a Moderna vaccine, and return the results sorted in descending order. This query was an elaboration on the previous one, to highlight ordering in aggregations and selecting a limited number of documents on which to perform the aggregation.

```
GET somministrazioni-vaccini-latest/_search
{
  "size":0,
  "query":{
    "term":{
      "fornitore": "Moderna"
```

```
    }
  },
  "aggs":{
    "population":{
      "terms":{
        "field":"nome_area",
        "order": {"first_dose_percentage": "desc"}
      },
      "aggs":{
        "first_dose_percentage": {
          "sum": {
              "script": "10000000 * doc['prima_dose'].value / doc['region.population'].v
            }
        }
      }
    }
  }
}
```

Response:

```
{
  "took" : 1,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 10000,
      "relation" : "gte"
    },
    "max_score" : null,
    "hits" : [ ]
  },
  "aggregations" : {
    "population" : {
      "doc_count_error_upper_bound" : 0,
      "sum_other_doc_count" : 28580,
      "buckets" : [
        {
          "key" : "Toscana",
          "doc_count" : 2820,
          "first_dose_percentage" : {
            "value" : 1302260.0
          }
        },
```

```
{
  "key" : "Umbria",
  "doc_count" : 2409,
  "first_dose_percentage" : {
    "value" : 1268722.0
  }
},
{
  "key" : "Sardegna",
  "doc_count" : 2541,
  "first_dose_percentage" : {
    "value" : 1218292.0
  }
},
{
  "key" : "Molise",
  "doc_count" : 1769,
  "first_dose_percentage" : {
    "value" : 1156841.0
  }
},
{
  "key" : "Piemonte",
  "doc_count" : 2741,
  "first_dose_percentage" : {
    "value" : 1145774.0
  }
},
{
  "key" : "Lombardia",
  "doc_count" : 2926,
  "first_dose_percentage" : {
    "value" : 1104515.0
  }
},
{
  "key" : "Emilia-Romagna",
  "doc_count" : 2951,
  "first_dose_percentage" : {
    "value" : 1092927.0
  }
},
{
  "key" : "Abruzzo",
  "doc_count" : 2736,
  "first_dose_percentage" : {
    "value" : 1082611.0
  }
},
{
  "key" : "Veneto",
  "doc_count" : 2914,
  "first_dose_percentage" : {
    "value" : 1056627.0
  }
},
{
  "key" : "Basilicata",
  "doc_count" : 2390,
  "first_dose_percentage" : {
    "value" : 1056571.0
```

```
        }
      }
    ]
  }
 }
}
```

2.2 Commands

Command n. 1

Delete all the somministrations in the age group 12-19

```
POST vaccini/_delete_by_query
{ "query":{
    "match":{
      "fascia_anagrafica": "12-19"
    }
  }
}
```

Command n. 2

Change the supllier name "Vaxzevria (AstraZeneca)" into "Astrazeneca"

```
POST vaccini/_update_by_query
{ "query":{
    "match":{
      "fornitore": "Vaxzevria (AstraZeneca)"
    }
  },
  "script": {
    "inline": "ctx._source.fornitore='Astrazeneca'",
    "lang": "painless"
  }
}
```

3 Kibana Implementation

Kibana implements an interface that allows you to use its tools very easily from the first moment, thanks to the auto-completion of the code, features and panels already created and only to be set. However, these proved to be an obstacle when the representation of queries required data to be obtained through operations with other fields.

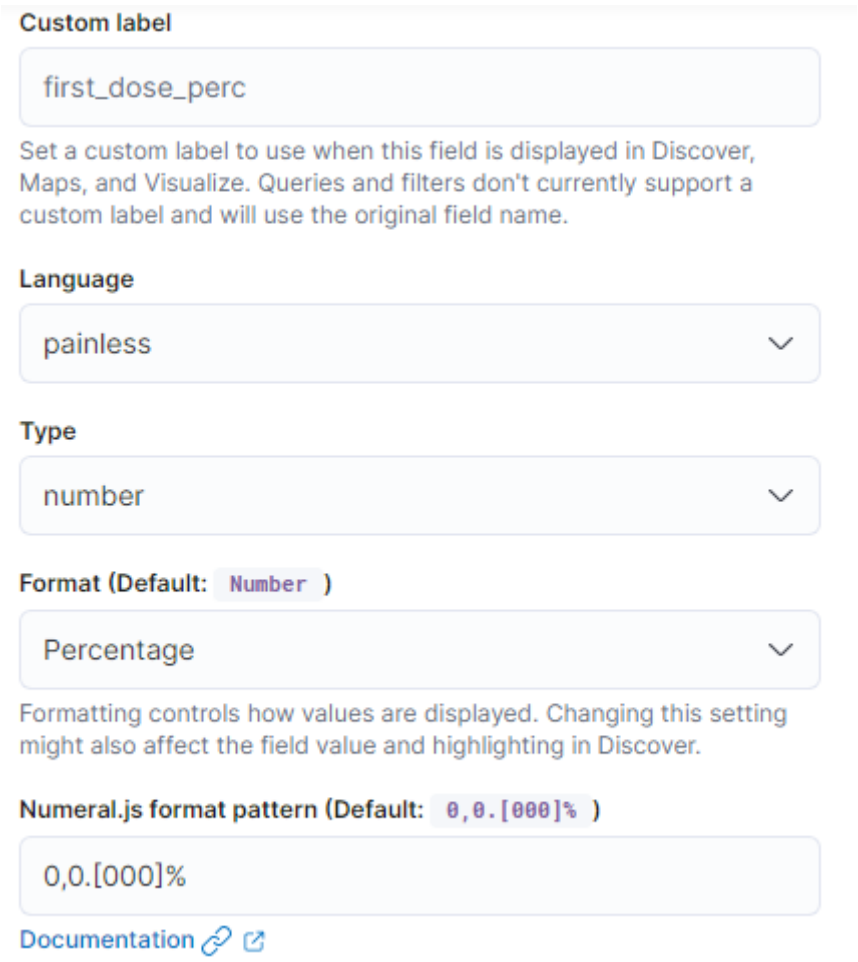
3.1 Settings

We had to create additional fields for the second, seventh and eighth query. We preferred scripted fields to changing the database for two reasons: relative portability (the installation of our dashboard does not rely on manual modification of the script, which is updated often with the same schema) and learning purposes ("manually" editing the database with a script does not require elastic-specific knowledge). Total Vaccines is the scripted field for the second query, created because in the map settings the aggregation field allows one to only perform a on a single field.

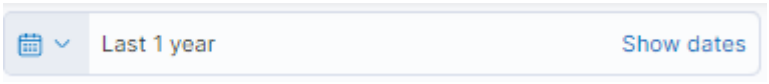
Name	Lang	Script	Format
Totale_Vaccini	painless	doc['prima_dose'].value+doc['seconda_dose'].value+doc['dose_addizionale_booster'].value	

First Dose Per is the scripted field for the seventh and eighth query, we created this field because in the map settings there is no possibility to do other operations besides the preset ones, we also wanted to obtain a visualization in percentage.

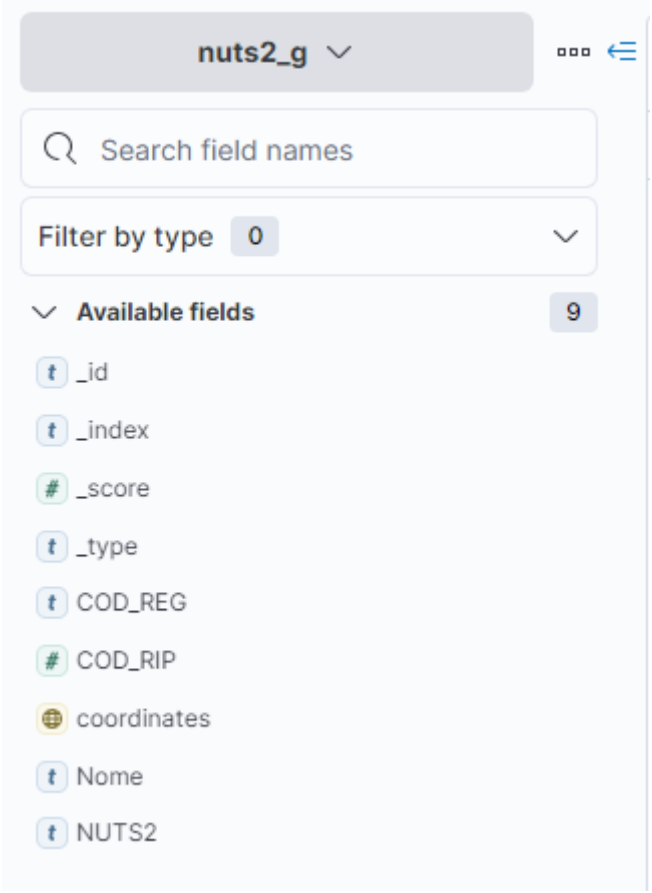
Name	Lang	Script	Format
first_dose_perc	painless	doc['prima_dose'].value/(doc['region.population'].value*1.0)	

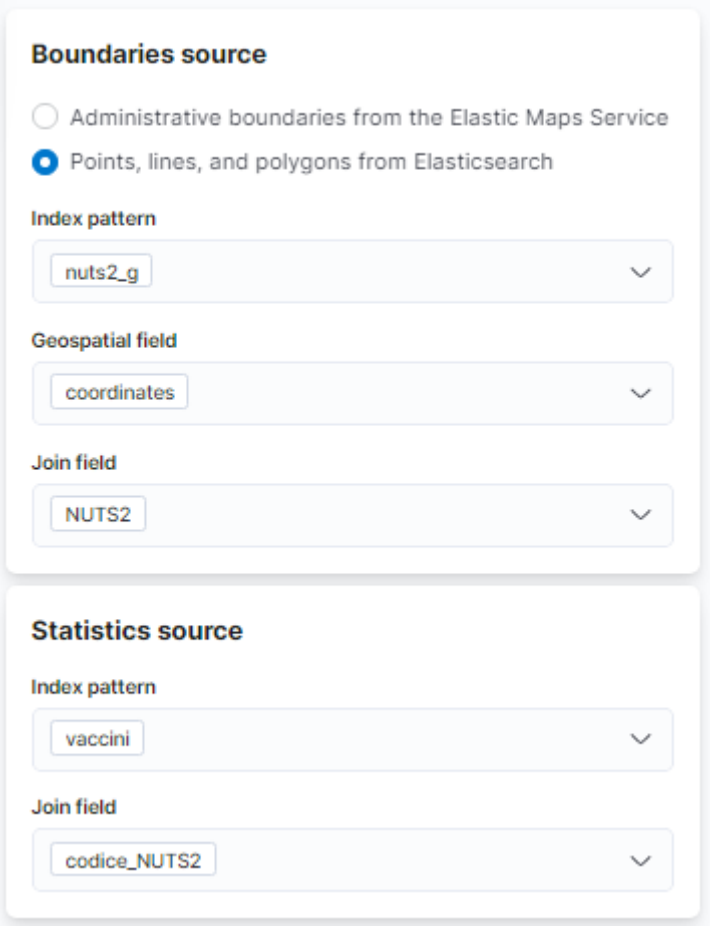


The time setting we used:



We have found a problem on the association of the Kibana native map , to solve the problem we have uploaded a geojson dataset that maps the Italian regions.[link]





3.2 Queries and visualizations

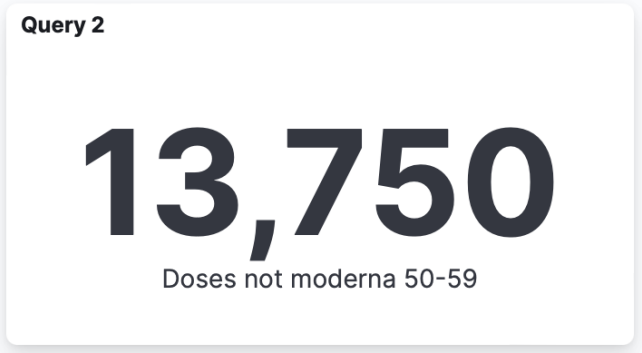
When possible, we tried to maintain a consistent color scheme for vaccine types. Overall, we focused on map-oriented visualizations, as we found these were the ones that revealed the most interesting data.

We included three "bonus" visualizations, which were not picked as queries, but still provided us with relevant data.

3.2.1 Query 1

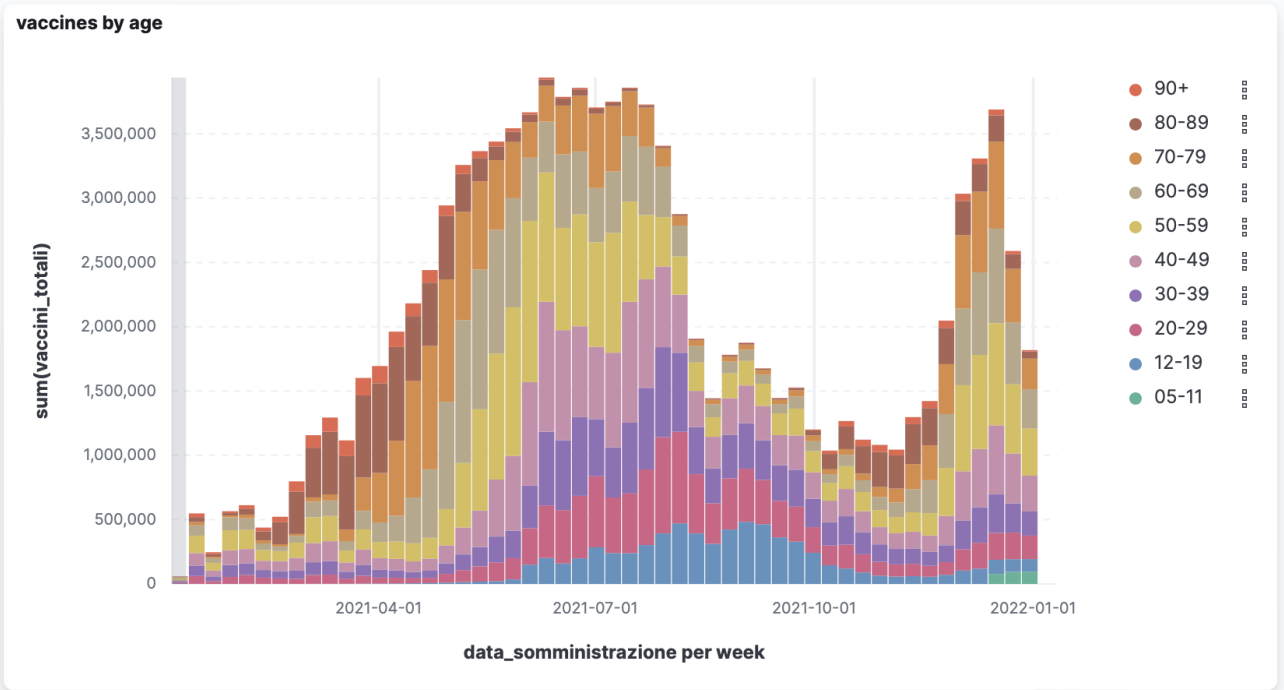

Query 1
München
Wien
Zürich
AUSTRIA
HUNGARY
SWITZERLAND
265,320
Genève
326,249
625,267
Ljubljana
63,958
5,355,730
2,574,789
CROATIA
2,300,492
2,399,660
BOSNIA AND HERZEGOVINA
812,490
MONACO
Marseille
2,030,082
786,634
448,755
665,443
3,049,229
162,480
2,879,218
2,138,426
ALBANIA
296,986
852,347
Mar Tirreno / Mer Tyrrhénienne
1,001,499
Palermo
2,636,973
Tunis
تونس
Constantine
قسنطينة
Sousse
MALTA
zoom: 4.62
Elastic Maps Service, OpenMapTiles, OpenStreetMap contributors

3.2.2 Query 2



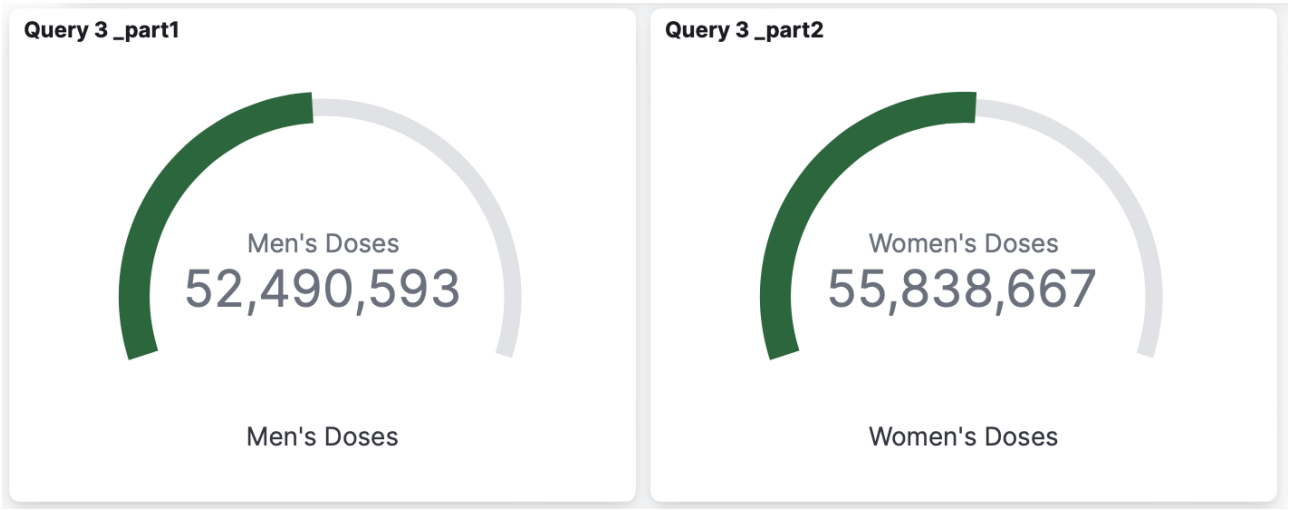
3.2.3 Vaccines segmented by age group

This is the first "bonus" visualization we included, and showcases some interesting data for both when different segments of the population *were allowed* or *chose* to get the vaccine. For example, vaccines were only recently approved for young children under the age of 11, and this is appropriately portrayed by the data.



3.2.4 Query 3

In this query, the total amount represented by the background grey line corresponds to the total amount of vaccines administered for that sex on the entire timeframe.

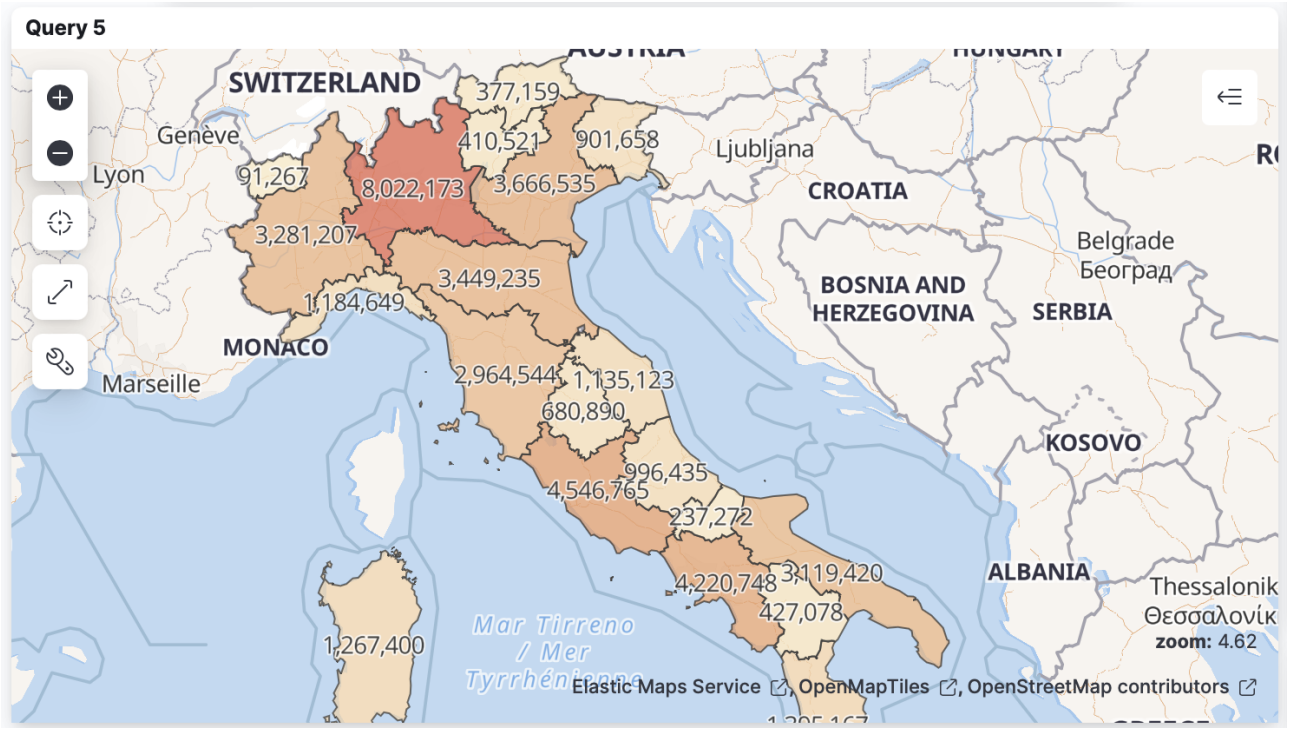

Query 3 _part1
Men's Doses
52,490,593
Men's Doses
Query 3 _part2
Women's Doses
55,838,667
Women's Doses

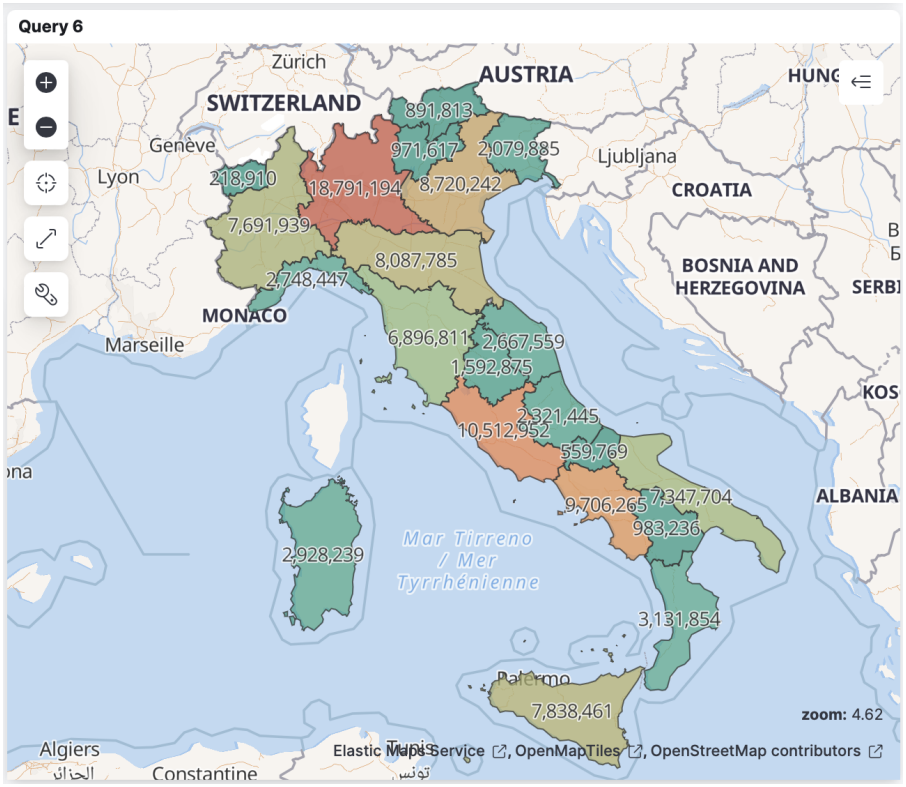
3.2.5 Query 4


Query 4
SWITZERLAND
Genève
Lyon
Marseille
MONACO
168,353
139,032
314,267
51,105
3,718,803
1,545,136
1,663,530
1,714,332
512,170
1,369,073
449,005
250,693
411,216
1,365,869
91,529
1,637,082
1,130,024
162,066
557,910
Ljubljana
CROATIA
Belgrade
Београд
BOSNIA AND
HERZEGOVINA
SERBIA
MONTENEGRO
KOSOVO
NORTH
MACEDONIA
ALBANIA
Mar Tirreno
lat: 42.86109, lon: 2.7804, zoom: 4.8
Elastic Maps Service, OpenMapTiles, OpenStreetMap contributors

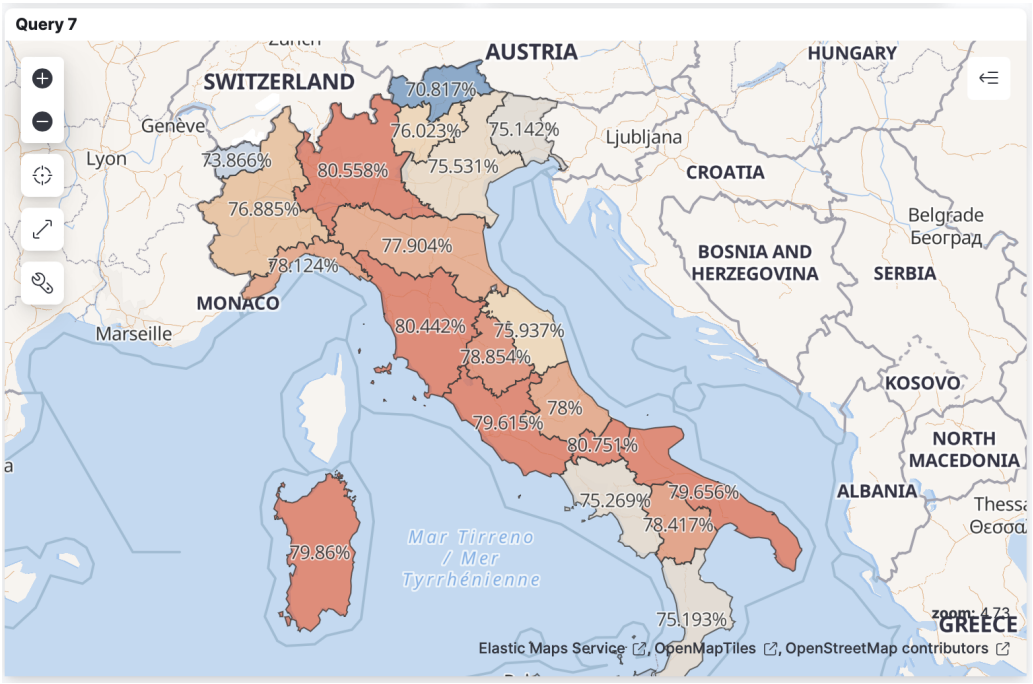
3.2.6 Queries 5 and 6

From this point on, all queries refer to the selection box which allows one to filter for a specific region or age; we found this accomplished two main tasks: increasing and showcasing the dynamic "feel" of the dashboard, and focusing on specific regions.

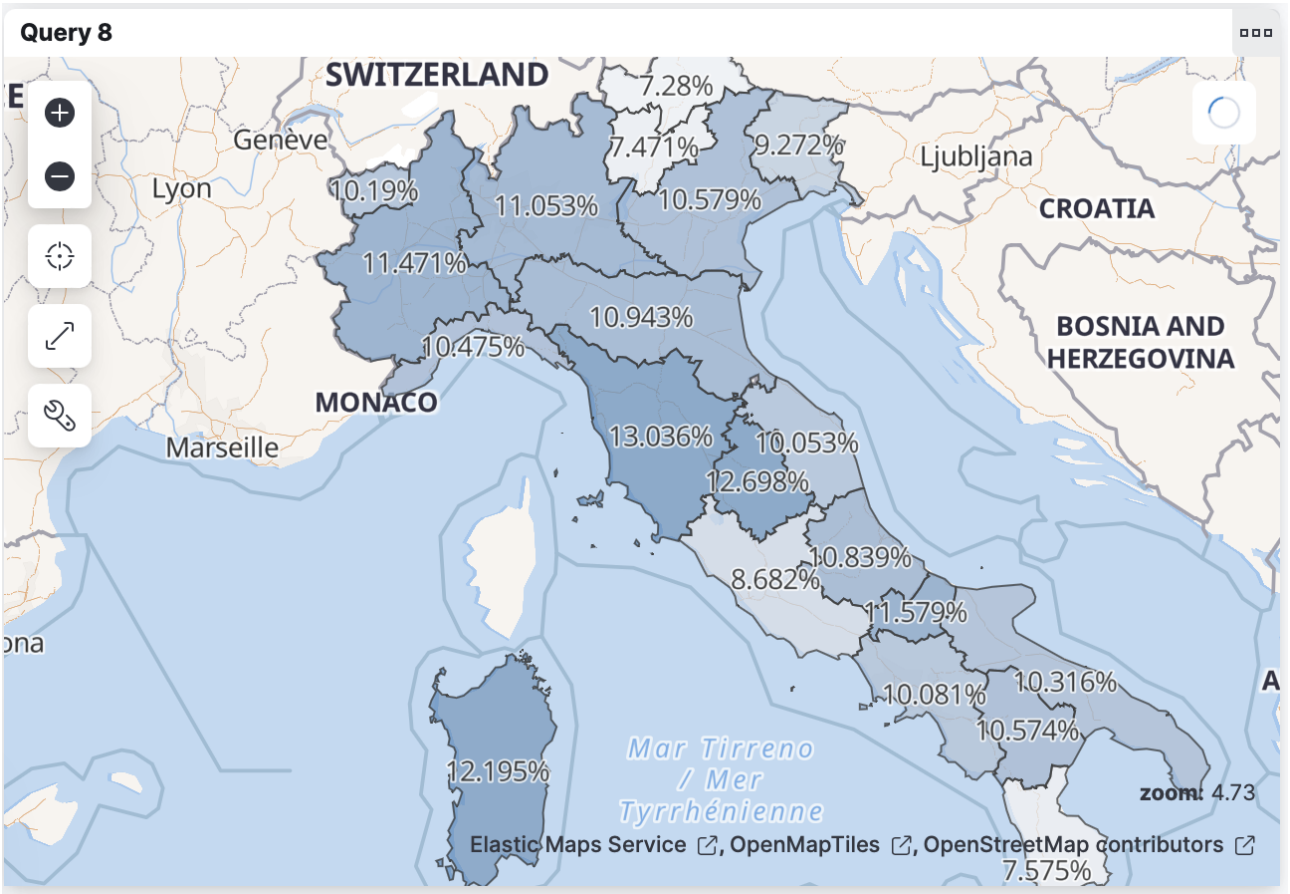




3.2.7 Query 7

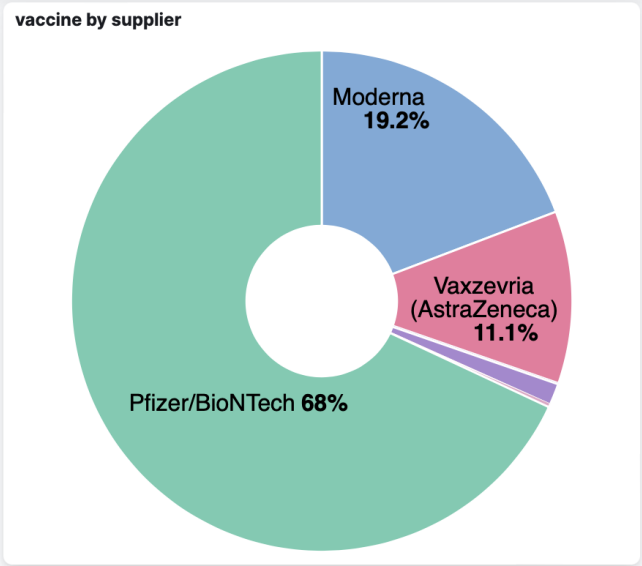


3.2.8 Query 8



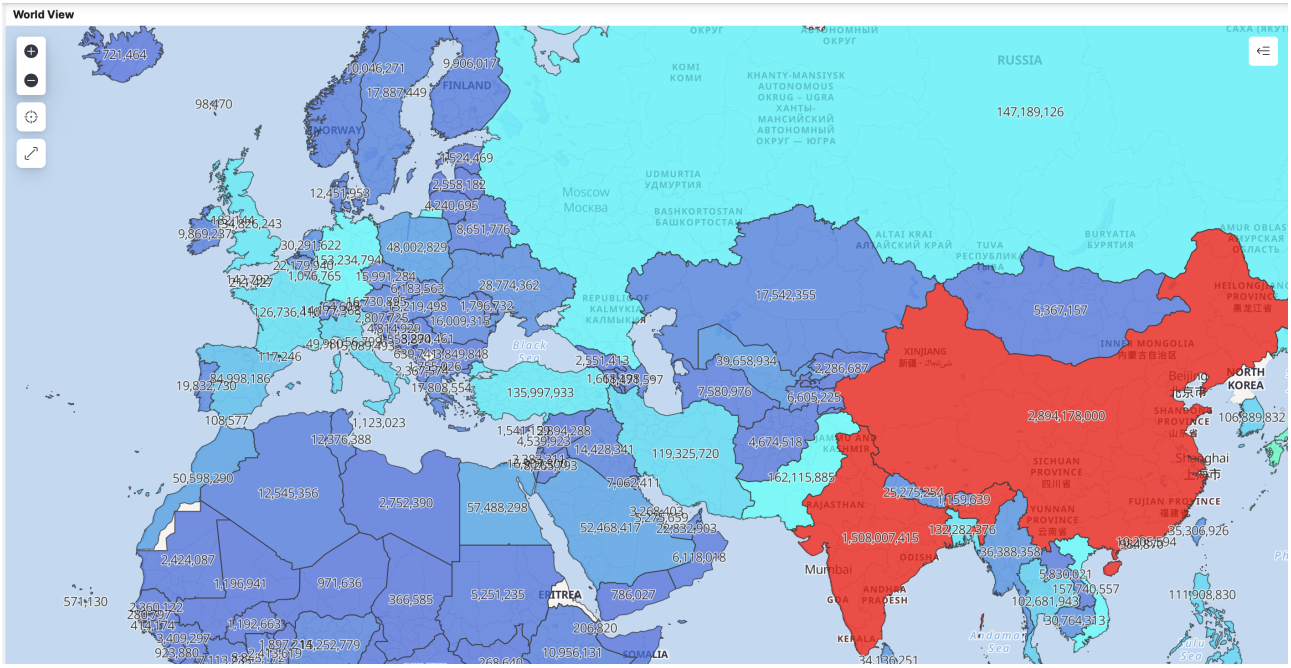
3.2.9 Supplier

This is the second "bonus" visualization, and highlights the distribution of vaccine across suppliers. We decided for this "simpler" view, when compared to the segmented vertical graph shown earlier, because the only insight that could be gleamed from a time-related graph was how the AstraZeneca vaccine was rarely recently administered, and made for a less understandable dashboard.

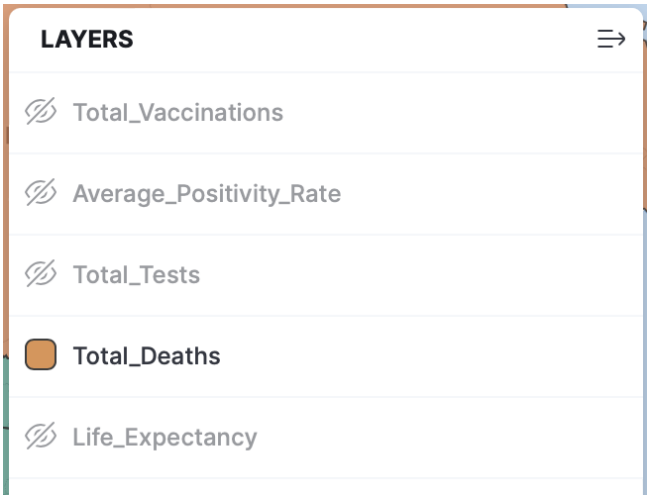


3.3 YADI - Yet Another Dataset Integration

We found a massive, resourceful, interesting dataset here, which we wanted to add to our Kibana implementation for a world view. The last tile in our dashboard consist in a full world view map with plenty of information layers.
For example we can see the total number of death by state:



Following this are the layers that are available



A very nice feature would be introducing a time slider, Kibana has one in the 'Edit Panel' but only there. There are a some implementation found in GitHub made by the community.

4 Conclusion

In this project, we experimented with some functionalities of the ElasticStack, like Elastic-Search for queries, Kibana for visualization and also Ingestion pipelines. This report highlights the potential of Elasticsearch and Kibana visualization, as a modern and interactive way of storing and managing data compared to normal databases. From the practical point of view it has been interesting to see many personalizations ElasticStack offers, but it's not easy to master and the query language was not immediate and user-friendly. We found Kibana to be suitable for smaller projects or simple, immediate dashboards: when used this way, it is extremely quick to create visually appealing visualizations; at the same time, it falls off quickly once more advanced features are needed, as its query language is simplified, only some operations are possible and the library of possible visualizations, although visually appealing, isn't comparable to a stand-alone solution.
Still, this last section of the project allowed us to refine the data-engineering-oriented skills we matured throughout the course (by tackling complex problems with both elastic-specific features and general purpose solutions) and round off our data-scientist-oriented abilities (such as figuring out relevant visualizations and obtaining insights from them).