AY 2021-2022

POLITECNICO
MILANO 1863

SMBUD Project

ELK Stack based Vaccination Campaign Analysis System

Pablo Giaccaglia - 10626428
Salvatore Cassata - 10560790
Stefano Vighini - 10622788
Zhitao He - 10763530

Master of Science in
Computer Science & Engineering

10th January 2022

Contents

1 Introduction

Considering the scenario in which there's the need to build a system suitable for analysis over data about **COVID-19** vaccination statistics, we designed and build a **Kibana**[4] visualization tool relying on the **ELK stack**[2]. The data stored allows to extract actionable insights concerning various statistical purposes, involving information such as vaccine deliveries, administered doses amounts, suppliers and age groups. The third-part data entered in the database is updated daily as it concerns vaccination campaign and so it has been also used to build insightful dashboards providing visualization at various levels of granularity.

2 Specification & Hypothesis

In order to fulfill the purpose described in the introduction, we focused on the Italian vaccination campaign by relying on the daily updated open data about delivery and administration of **COVID-19** vaccines provided by the **Italian Ministry of Health**. Of the data available at this Github repository. , only three datasets were picked to feed the system.

The idea is to provide a tool aware of data updates, reason for why we implemented a small data processing pipeline to fetch, slightly change and standardize the data uploaded to the aforementioned repository. More details about this process will be provided in the next sections.

3 Data schema

In the following subsections the three used datasets are now showed from the structural point of view and briefly described.

3.1 "somministrazioni-vaccini-latest" dataset

Italian vaccination campaign data of daily administered vaccines divided by region and age group of the vaccinated subjects. The shape of the data is the same as in the one of the Github repository, with the exception of translated fields from Italian to English and the addition of the **"region_coordinates"** field. More details about these changes are shown in section 4.

Field	**Data type**	**Description**
administration_date	Date	Administration date of the vaccines
supplier	Keyword	Complete name of the supplier of the vaccine
area	Text	Acronyms of the region of delivery
age_group	Keyword	Age group of the people administered with the vaccine
male_count	Integer	Number of vaccinations administered to males
female_count	Integer	Number of vaccinations administered to females
first_doses	Integer	Number of people administered with first dose
second_doses	Integer	Number of people administered with second dose
post_infection_doses	Integer	Number of people administered with a dose after they have been infected in the previous 3-6 months
booster_doses	Integer	Number of people administered with an additional dose/recall
NUTS1_code	Text	NUTS (Nomenclature of Territorial Units for Statistics) code 1 of Italy
NUTS2_code	Text	NUTS (Nomenclature of Territorial Units for Statistics) code 2 of Italy
region_ISTAT_code	Integer	ISTAT region code
region_name	Keyword	Name of the region
region_coordinates	Geo point	Latitude-longitude pair of region coordinates

3.2 ”anagrafica-vaccini-summary-latest” dataset

Italian vaccination campaign data of overall administered vaccines, including female and male count. The shape of the data is the same as in the one of the Github repository, with the exception of translated fields from Italian to English. More details about these changes are shown in section 4.

Field	**Data type**	**Description**
age_group	Keyword	Age group of vaccinated people
total_administered	Integer	Overall amount of administered vaccine doses
male_count	Integer	Overall number of vaccinations administered to males
female_count	Integer	Overall number of vaccinations administered to females
first_doses	Integer	Overall number people administered with first dose
second_doses	Integer	Overall number people administered with second dose
post_infection_doses	Integer	Overall number of people administered with a dose after they have been infected in the previous 3-6 months
booster_doses	Integer	Overall number of people administered with an additional dose/recall
last_update	Date	Latest update date

3.3 ”consegne-vaccini-latest” dataset

Italian vaccination campaign data of daily delivered vaccines divided by region and supplier. The shape of the data is the same as in the one of the Github repository, with the exception of translated fields from Italian to English and the addition of the **”region_coordinates”** field. More details about these changes are shown in section 4.

Field	**Data type**	**Description**
area	Text	Acronyms of the region of delivery
supplier	Keyword	Complete name of the supplier of the vaccine
doses_amount	Integer	Number of delivered doses
delivery_date	Date	Date of the delivery
NUTS1_code	Text	NUTS (Nomenclature of Territorial Units for Statistics) code 1 of Italy
NUTS2_code	Text	NUTS (Nomenclature of Territorial Units for Statistics) code 2 of Italy
region_ISTAT_code	Integer	ISTAT region code
region_name	Keyword	Name of the region
region_coordinates	Geo point	Latitude-longitude pair of region coordinates

4 Database Creation

As stated in the introduction section, we decided to build a Python[5] data pipeline to provide a daily database update, while ensuring to be consistent with the modifications we applied to the original datasets. Obviously a manual approach isn't feasible for this purpose, so different steps taken into account by the daily routines are now briefly explained, referencing to a real use case scenario.

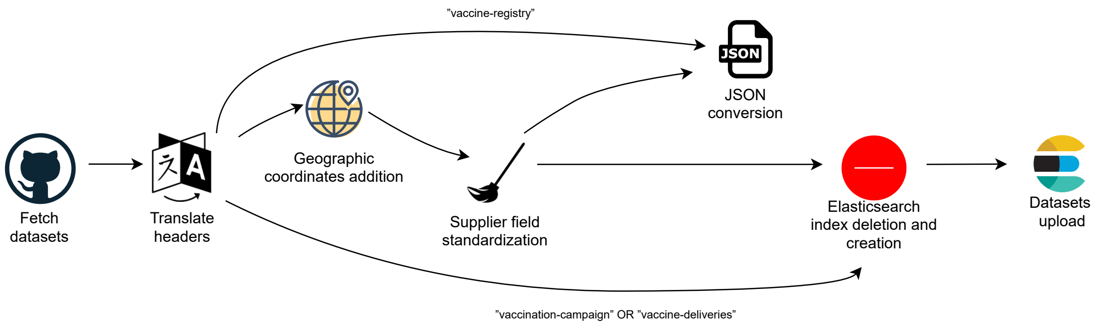

"vaccine-registry"
JSON
JSON conversion
Fetch datasets
Translate headers
Geographic coordinates addition
Supplier field standardization
Elasticsearch index deletion and creation
Datasets upload
"vaccination-campaign" OR "vaccine-deliveries"

Figure 1: Pipeline steps

- **Datasets fetch from Github repository**: **CSV files** are updated every day at around **5:15 AM**, then some sort of watcher-notifier tool (not integrated at the moment), such as **GitHub File Watcher**[?], could trigger the daily routines functions (implemented), whose first step is the files fetch for processing.
- **CSV Header translation**: the original CSV files' headers are in Italian, so we decided to translate them in English to widen the audience of users. This modification doesn't affect the content of the datasets and it is easily adaptable to header names changes, since the translation is performed daily through a lookup table (**Python Enum subclass**) which can be easily changed through add-ons, modifications and deletions. File names were also translated, and the resulting names are: "vaccination-campaign" for **"somministrazioni-vaccini-latest"**, **"vaccine-registry"** for **"anagrafica-vaccini-summary-latest"** and "vaccine-deliveries" for **"consegne-vaccini-latest"**. Note that these translations correspond to the **Elasticsearch**[3] Indexes names and from now these document and their fields will be named after their English translation.
- **Adding region coordinates**: This addition is performed only on **"vaccine-deliveries"** and **"vaccination-campaign"** datasets, since these contain a **"region_name"** field, through which is possible to associate the corresponding region coordinates as a pair. The coordinates are retrieved from a previously created file containing mappings of Italian regions to their corresponding coordinates, retrieved through the **GMaps Geocoding API**[1]. This information doesn't alter the original data and is used to provide **Kibana** Geo data visualizations of the vaccination campaign. Note that this operation is performed by a Python function that lets choosing the way in which the data is added: through a single CSV column named **"region_coordinates"** or through 2 CSV columns named **"region_longitude"** and **"region_latitude"**, containing the corresponding float values. This addition is compatible with the daily updates, since we assume that it is unlikely a change of the **"region_name"** filed values. But if there will be such case, for example concerning lower/upper case characters, it is almost immediate to face it by changing the lookup table region names, or by writing appropriate functions to face multiple scenarios.
- **Supplier filed "standardization"**: This is the only significant modification we applied to two of the original datasets: **"vaccination-campaign"** and **"vaccine-deliveries"**. The alteration involves **"supplier"** field of both datasets, whose values are the names of vaccine's suppliers, e.g: **Pfizer/BioNTech, Moderna**. In detail among the possible values of this textual field there are **"Pfizer/BioNTech"** and **"Pfizer Pediatrico"**, that is "Pediatric Pfizer". Since we don't consider this distinction meaningful for the insight we want to provide the user through this system, we decided to standardize both fields under the name of the former. Note that also in this case same reasoning done for the previous step concerning daily updates and format changes can be applied.
- **Datasets upload to Elasticsearch**: After the data manipulation phase, some additional steps are taken to successfully upload the data. Firstly the three indexes, if previously created, are deleted from Elasticsearch, then recreated to upload the newer data from the final CSVs. This

step, which could sound not optimal, is taken because we noticed eventual updates of older data (e.g number of doses amount was partial). Even though more sophisticated solutions involving detection changes methods and upload of only new/changed entries could be adopted, we didn't do that to focus on other aspects of the project.

- **Additional notes**: Through the provided Python routines in the last step of the process, final CSVs are converted to Json format for eventual further usages. The datasets are uploaded online through **Elasticsearch DSL Python library**, manual mapping is performed in this phase for 2 reasons: aim to have uploaded data ready for interaction without further tweaks and to have full control on data types, to avoid eventual unwanted automatic mappings and to specify Keyword fields, to perform aggregation both in Elasticsearch queries and Kibana dashboards on textual fields.

5 Kibana dashboard panels

In the following chapter we are describing the Kibana dashboards we implemented. We used different indexes and we also tried to cover different ways of **representation** (bar charts, maps, pie charts) and different **functions** (cumulative sums, total sums, daily data). Most of the dashboards are **interactive**: this means you can select an interval with the left click of the mouse and the graph will zoom into it; or for other types you can select/deselect some data (for example, in the map).
Note that even though in Italy the first vaccine against **COVID-19** was administered on **27/12/2020**, the following charts, when the **x-axis** is divided into week intervals, the starting week is on **21/12/2020.**

5.1 Gender distribution of weekly vaccine doses administered

This bar chart shows the amount of vaccine doses administered each week, from the one starting on **21/12/2020**. The shown amounts are based uniquely on the overall doses, the only distinction made is in the gender of vaccinated people. Note that if daily database updates occur, the current week's bar may refer partial amounts. Thanks to Kibana tools, week range can be customized, impacting the **x-axis**, and fields can be hidded to shift the focus on a portion of them. This chart relies on the **"vaccination-campaign"** index and its data can be obtained through the query shown at subsection 6.1.3

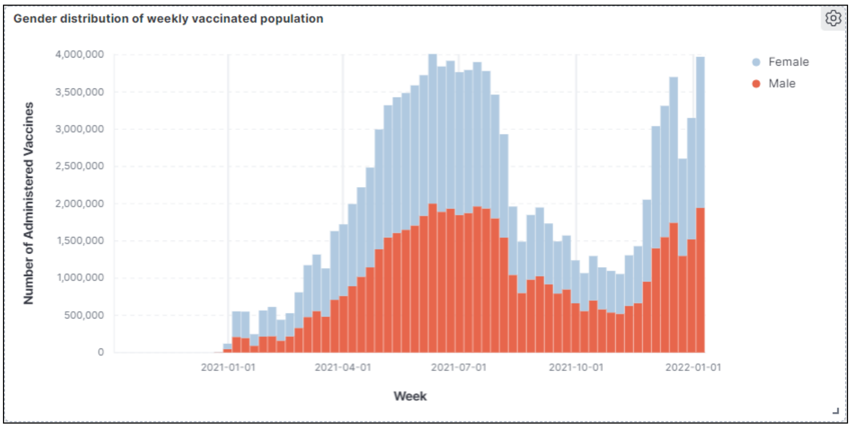


Figure 2: Gender distribution of weekly vaccine doses administered

5.2 Number of not vaccinated people

This bar chart shows the decrease of overall amount of not vaccinated people each week, from the one starting on **21/12/2020**. The shown amounts are based uniquely on the cumulative sum of first doses. Note that, since it is not possible to have some sort of tool able to provide the current population given the filters applicable on the data, the values shown are computed by subtracting the cumulative sum from the overall Italian population value (**60,327,146**), which is hardcoded. Finally note that the number of not vaccinated people includes not vaccinable people. This chart relies on the **"vaccination-campaign"** index and its data can be obtained through the query shown at subsection 6.1.7

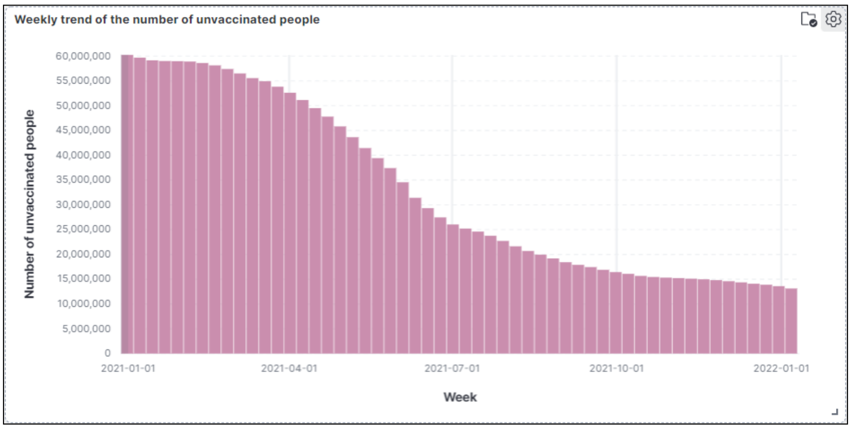


Figure 3: Number of not vaccinated people

5.3 Weekly trend of the number of vaccinations

This line chart shows the amount of vaccine doses administered each week, from the one starting on **21/12/2020**. The shown amount doesn't consider distinction between males and females, but uniquely the distinction between first, second and booster dose. Thanks to Kibana tools, week range can be customized, impacting the **x-axis**, and fields can be hided to shift the focus on a portion of them. This chart relies on the **"vaccination-campaign"** index and its data can be obtained through the query shown at subsection 6.1.1

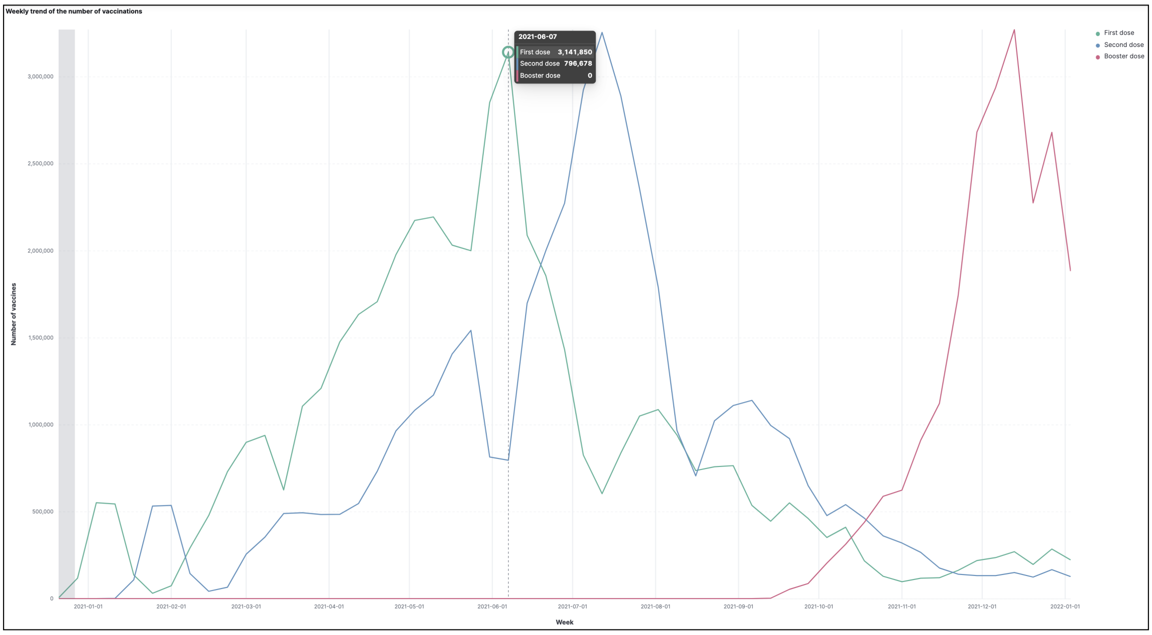


Figure 4: Weekly trend of the number of vaccinations

5.4 Total number of doses delivered

This area chart shows the increase of overall amount of vaccines delivered each week, from the one starting on **21/12/2020**. The shown amount doesn't consider distinction between suppliers. Thanks to Kibana tools, week range can be customized, impacting the **x-axis**. This chart relies on the **"vaccine-deliveries"** index and its data can be obtained through the query shown at subsection 6.1.11.

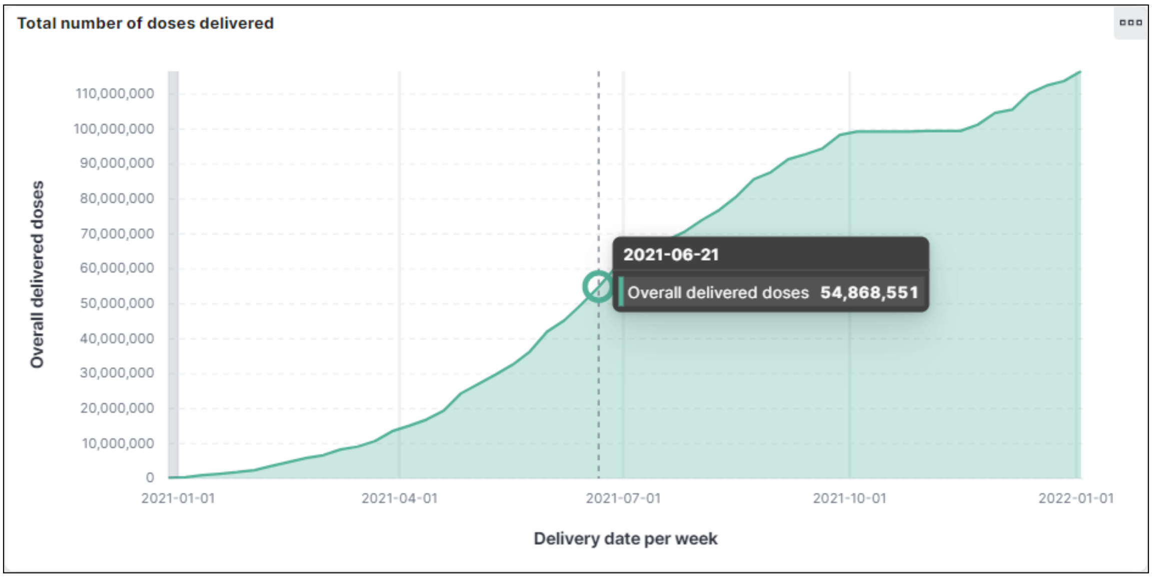


Figure 5: Total number of doses delivered

5.5 Distribution of suppliers of all administered vaccines

This pie chart shows the distribution of doses administered from the supplier point of view. The shown amount doesn't consider distinction between males and females, nor the distinction between first, second and booster dose. The shown amounts are based uniquely on the cumulative sum of first, second and booster doses. Thanks to Kibana tools, date range can be customized, impacting the chart aspect and allowing to focus on the distribution trend of a certain period of time. This chart relies on the **"vaccination-campaign"** index and its data is the one obtained through the query shown at subsection 6.1.4.

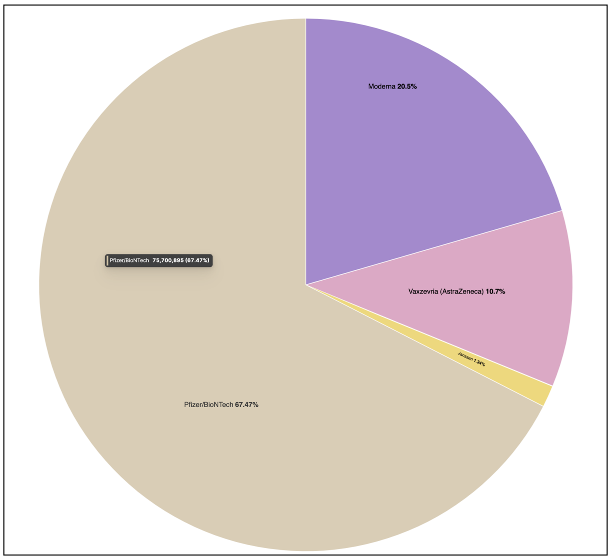


Figure 6: Distribution of suppliers of all administered vaccines

5.6 Vaccination status of all age groups

This bar chart shows the vaccination status of all age groups, considering percentages relative to the people who got at least one vaccine, to compute the ones relative to both people waiting the second dose, the booster dose and the ones fully vaccinated. Not vaccinated people are not considered. **"waiting for taking the second dose"** percentages are computed considering the difference between the sum of first doses and second doses, by age range. **"waiting for taking the booster dose"** percentages are computed considering the difference between the sum of second doses and booster doses, by age range. Finally the **"fully vaccinated"** percentage is simply the sum of booster doses administered, by age range. Thanks to Kibana tools, week range can be customized, impacting the **x-axis**, and fields can be hidded to shift the focus on a portion of them. This chart relies on the "vaccination-campaign" index and its data can be obtained through the query shown at subsection 6.1.6

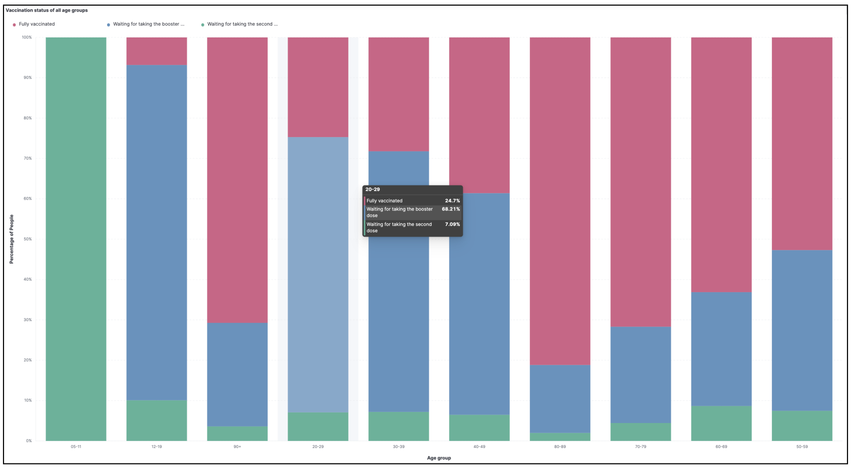


Figure 7: Vaccination status of all age groups

5.7 Doses map

This map shows the total number of doses delivered from the start of the pandemic up to now. As you can see, bigger amounts of doses delivered correspond to bigger circles with a red color, while smaller amounts to smaller circles with green color. On the left, you can see multiple layers for this map have been created, just to show you it's possible to show more data at once and to hide some layers. However, this is not useful in our case as the region coordinates make this layers overlap and not effectively readable. The data of the map can be retrieved by query 6.1.15.

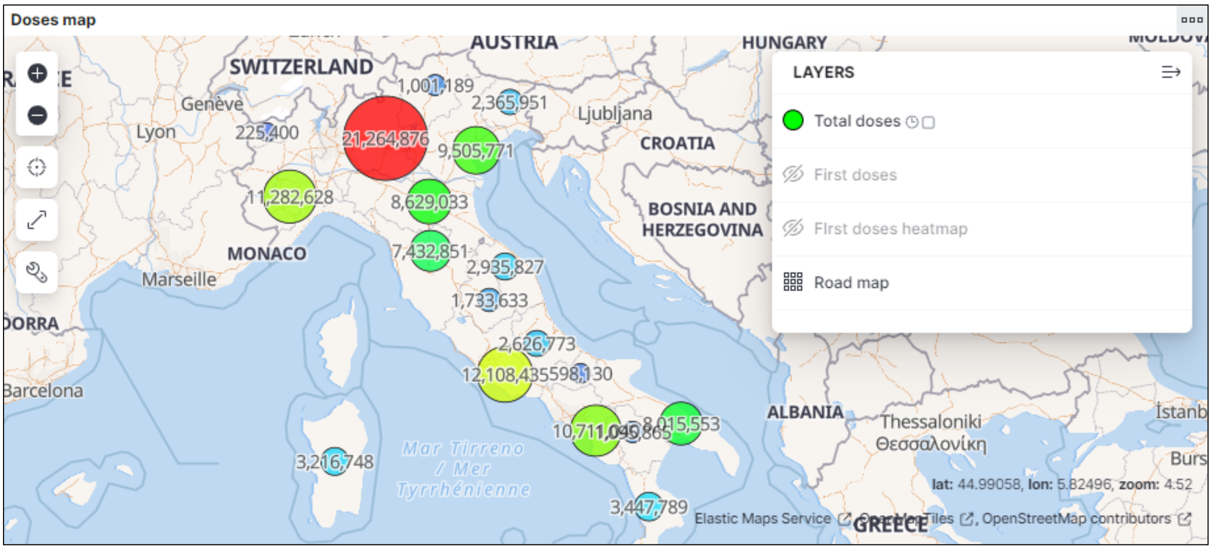


Figure 8: Map of vaccine's doses delivered and administered from the beginning of the campaign

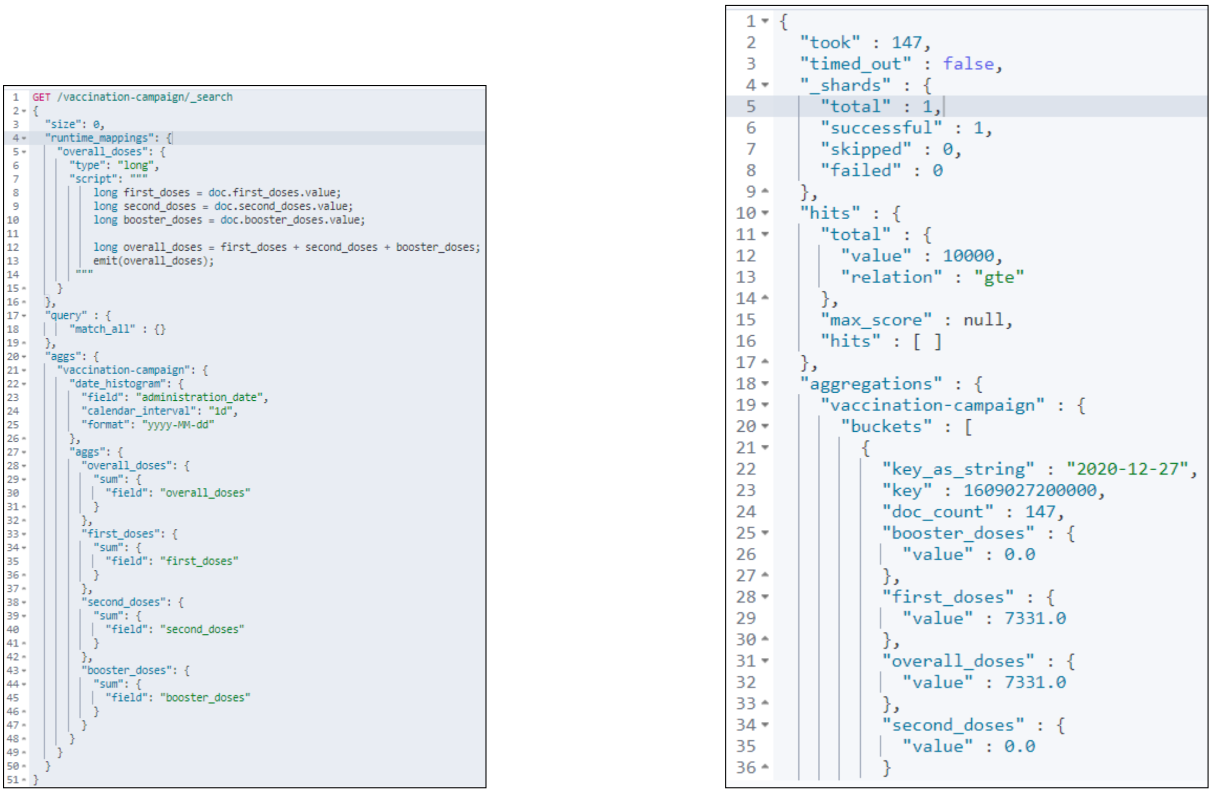
6 Queries & Commands

6.1 Queries

The following subsection presents some useful queries and their possible textual result, with the aim of showing database's utility

6.1.1 Daily number of overall, first, second and booster vaccine doses

Note that this query has a dedicated dashboard's panel, as explained in subsection 5.3



```
GET /vaccination-campaign/_search
{
  "size": 0,
  "runtime_mappings": {
    "overall_doses": {
      "type": "long",
      "script": """
          long first_doses = doc.first_doses.value;
          long second_doses = doc.second_doses.value;
          long booster_doses = doc.booster_doses.value;

          long overall_doses = first_doses + second_doses + booster_doses;
          emit(overall_doses);
        """
    }
  },
  "query" : {
      "match_all" : {}
  },
  "aggs": {
    "vaccination-campaign": {
      "date_histogram": {
        "field": "administration_date",
        "calendar_interval": "1d",
        "format": "yyyy-MM-dd"
      },
      "aggs": {
        "overall_doses": {
          "sum": {
            "field": "overall_doses"
          }
        },
        "first_doses": {
          "sum": {
            "field": "first_doses"
          }
        },
        "second_doses": {
          "sum": {
            "field": "second_doses"
          }
        },
        "booster_doses": {
          "sum": {
            "field": "booster_doses"
          }
        }
      }
    }
  }
}
```

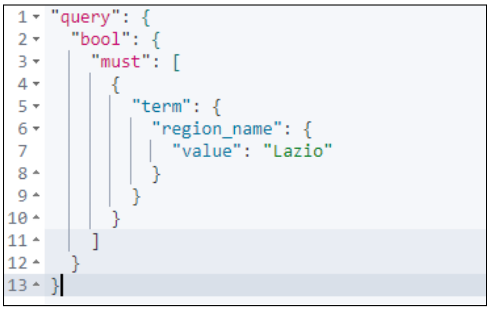
(a)

```
{
  "took" : 147,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 10000,
      "relation" : "gte"
    },
    "max_score" : null,
    "hits" : [ ]
  },
  "aggregations" : {
    "vaccination-campaign" : {
      "buckets" : [
        {
          "key_as_string" : "2020-12-27",
          "key" : 1609027200000,
          "doc_count" : 147,
          "booster_doses" : {
            "value" : 0.0
          },
          "first_doses" : {
            "value" : 7331.0
          },
          "overall_doses" : {
            "value" : 7331.0
          },
          "second_doses" : {
            "value" : 0.0
          }
```

(b)

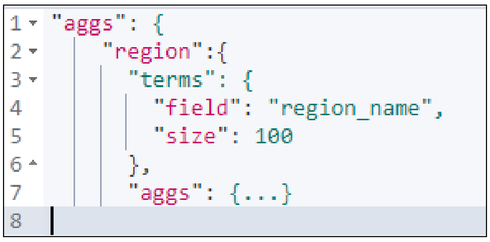
Figure 9: (a) shows the query, while (b) shows a portion of the result

With little modification to the above query, we can obtain interesting results. If we want to group to filter by region-name, we can simply add:



```
"query": {
  "bool": {
    "must": [
      {
        "term": {
          "region_name": {
            "value": "Lazio"
          }
        }
      }
    ]
  }
}
```

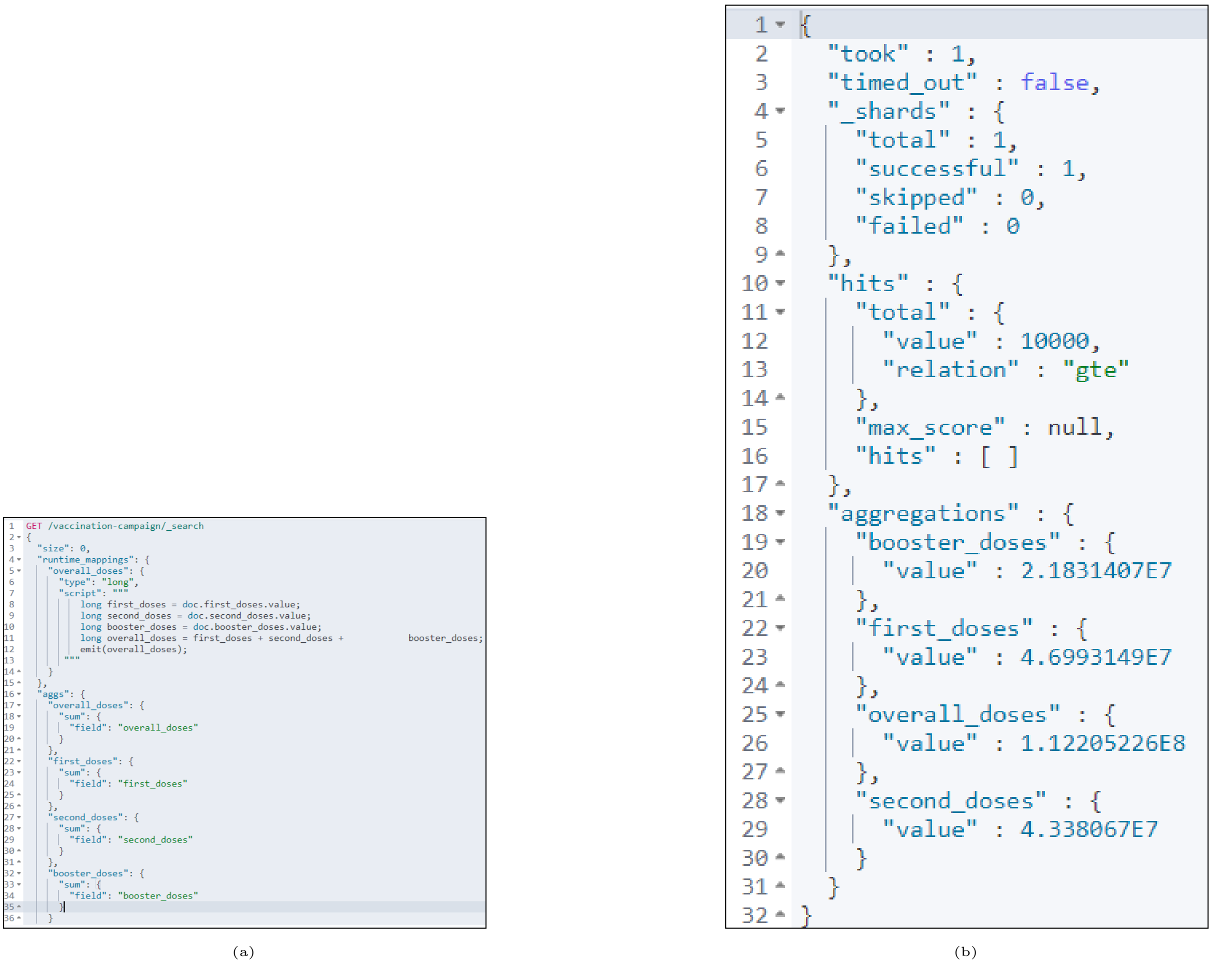
If instead we would like to group by region-name, we can add outside the "aggs" of "vaccination-campaign" this other aggregation:



```
"aggs": {
    "region":{
      "terms": {
        "field": "region_name",
        "size": 100
      },
      "aggs": {...}

```

Further examples (with different starting queries) showing this features are provided in the next queries.

6.1.2 COVID-19 Overall vaccine doses administered



```
GET /vaccination-campaign/_search
{
  "size": 0,
  "runtime_mappings": {
    "overall_doses": {
      "type": "long",
      "script": """
          long first_doses = doc.first_doses.value;
          long second_doses = doc.second_doses.value;
          long booster_doses = doc.booster_doses.value;
          long overall_doses = first_doses + second_doses +            booster_doses;
          emit(overall_doses);
        """
    }
  },
  "aggs": {
    "overall_doses": {
      "sum": {
        "field": "overall_doses"
      }
    },
    "first_doses": {
      "sum": {
        "field": "first_doses"
      }
    },
    "second_doses": {
      "sum": {
        "field": "second_doses"
      }
    },
    "booster_doses": {
      "sum": {
        "field": "booster_doses"
      }
    }
```

(a)

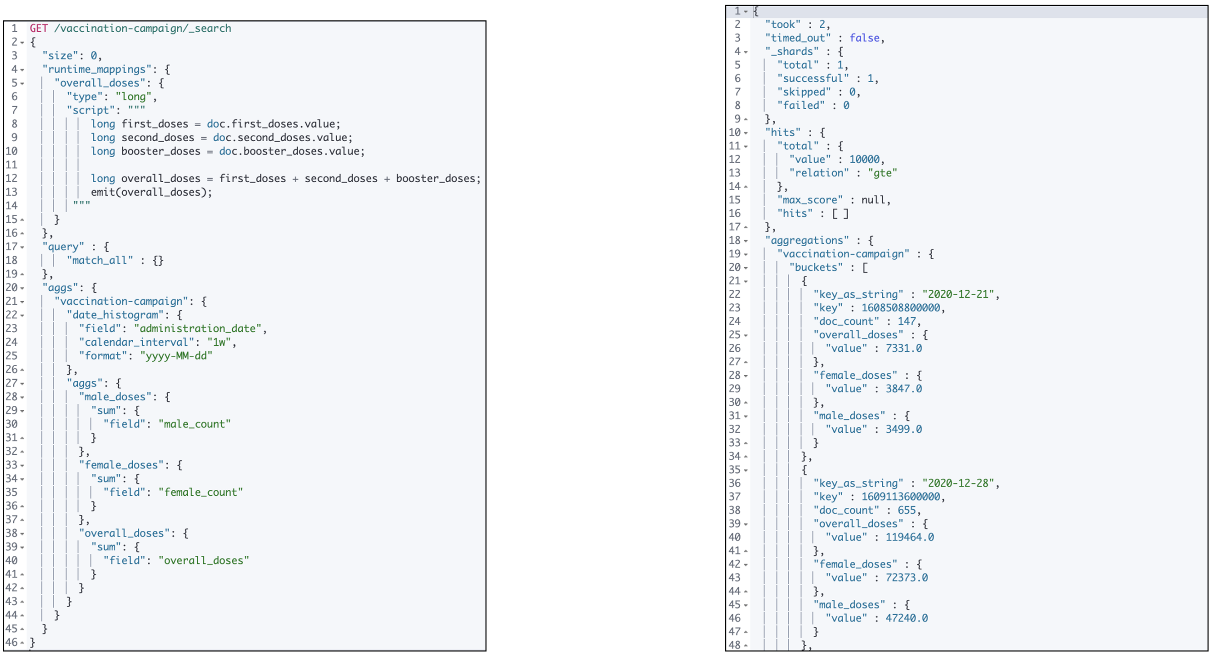
```
{
  "took" : 1,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 10000,
      "relation" : "gte"
    },
    "max_score" : null,
    "hits" : [ ]
  },
  "aggregations" : {
    "booster_doses" : {
      "value" : 2.1831407E7
    },
    "first_doses" : {
      "value" : 4.6993149E7
    },
    "overall_doses" : {
      "value" : 1.12205226E8
    },
    "second_doses" : {
      "value" : 4.338067E7
    }
  }
}
```

(b)

Figure 10: (a) shows the query, while (b) shows a portion of the result

6.1.3 Gender distribution of weekly vaccine doses administered

Note that this query has a dedicated dashboard's panel, as explained in section 5.1.



```
GET /vaccination-campaign/_search
{
  "size": 0,
  "runtime_mappings": {
    "overall_doses": {
      "type": "long",
      "script": """
          long first_doses = doc.first_doses.value;
          long second_doses = doc.second_doses.value;
          long booster_doses = doc.booster_doses.value;

          long overall_doses = first_doses + second_doses + booster_doses;
          emit(overall_doses);
        """
    }
  },
  "query" : {
      "match_all" : {}
  },
  "aggs": {
    "vaccination-campaign": {
      "date_histogram": {
        "field": "administration_date",
        "calendar_interval": "1w",
        "format": "yyyy-MM-dd"
      },
      "aggs": {
        "male_doses": {
          "sum": {
            "field": "male_count"
          }
        },
        "female_doses": {
          "sum": {
            "field": "female_count"
          }
        },
        "overall_doses": {
          "sum": {
            "field": "overall_doses"
          }
        }
      }
    }
  }
}
```

(a)

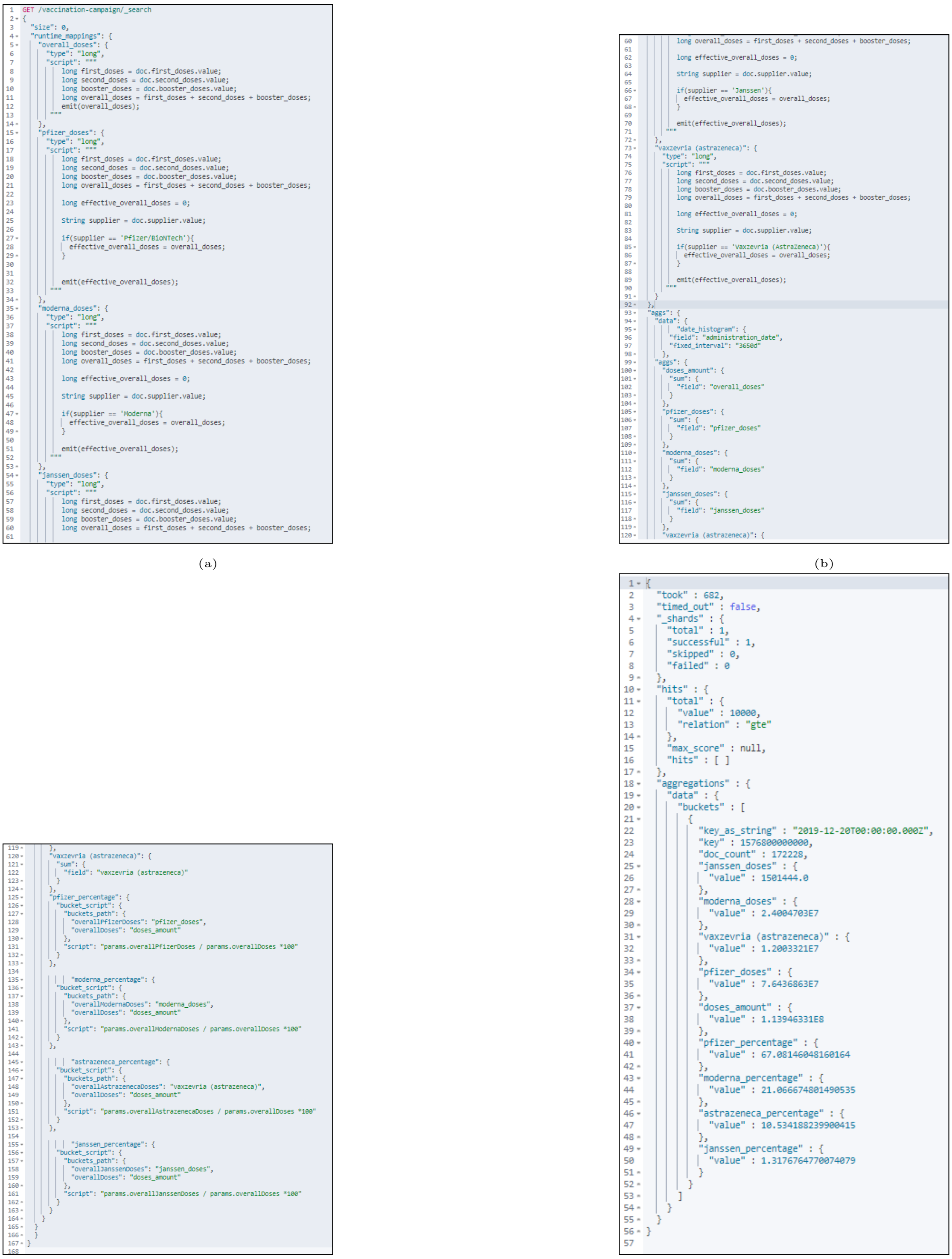
```
{
  "took" : 2,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 10000,
      "relation" : "gte"
    },
    "max_score" : null,
    "hits" : [ ]
  },
  "aggregations" : {
    "vaccination-campaign" : {
      "buckets" : [
        {
          "key_as_string" : "2020-12-21",
          "key" : 1608508800000,
          "doc_count" : 147,
          "overall_doses" : {
            "value" : 7331.0
          },
          "female_doses" : {
            "value" : 3847.0
          },
          "male_doses" : {
            "value" : 3499.0
          }
        },
        {
          "key_as_string" : "2020-12-28",
          "key" : 1609113600000,
          "doc_count" : 655,
          "overall_doses" : {
            "value" : 119464.0
          },
          "female_doses" : {
            "value" : 72373.0
          },
          "male_doses" : {
            "value" : 47240.0
          }
        },
```

(b)

Figure 11: (a) shows the query, while (b) shows a portion of the result

6.1.4 Suppliers distribution percentage of all administered vaccines

Note that this query has a dedicated dashboard's panel, as explained in section 5.5.



```
GET /vaccination-campaign/_search
{
  "size": 0,
  "runtime_mappings": {
    "overall_doses": {
      "type": "long",
      "script": """
          long first_doses = doc.first_doses.value;
          long second_doses = doc.second_doses.value;
          long booster_doses = doc.booster_doses.value;
          long overall_doses = first_doses + second_doses + booster_doses;
          emit(overall_doses);
        """
    },
    "pfizer_doses": {
      "type": "long",
      "script": """
          long first_doses = doc.first_doses.value;
          long second_doses = doc.second_doses.value;
          long booster_doses = doc.booster_doses.value;
          long overall_doses = first_doses + second_doses + booster_doses;

          long effective_overall_doses = 0;

          String supplier = doc.supplier.value;

          if(supplier == 'Pfizer/BioNTech'){
            effective_overall_doses = overall_doses;
          }


          emit(effective_overall_doses);
        """
    },
    "moderna_doses": {
      "type": "long",
      "script": """
          long first_doses = doc.first_doses.value;
          long second_doses = doc.second_doses.value;
          long booster_doses = doc.booster_doses.value;
          long overall_doses = first_doses + second_doses + booster_doses;

          long effective_overall_doses = 0;

          String supplier = doc.supplier.value;

          if(supplier == 'Moderna'){
            effective_overall_doses = overall_doses;
          }

          emit(effective_overall_doses);
        """
    },
    "janssen_doses": {
      "type": "long",
      "script": """
          long first_doses = doc.first_doses.value;
          long second_doses = doc.second_doses.value;
          long booster_doses = doc.booster_doses.value;
          long overall_doses = first_doses + second_doses + booster_doses;

```

(a)

```
          long overall_doses = first_doses + second_doses + booster_doses;

          long effective_overall_doses = 0;

          String supplier = doc.supplier.value;

          if(supplier == 'Janssen'){
            effective_overall_doses = overall_doses;
          }

          emit(effective_overall_doses);
        """
    },
    "vaxzevria (astrazeneca)": {
      "type": "long",
      "script": """
          long first_doses = doc.first_doses.value;
          long second_doses = doc.second_doses.value;
          long booster_doses = doc.booster_doses.value;
          long overall_doses = first_doses + second_doses + booster_doses;

          long effective_overall_doses = 0;

          String supplier = doc.supplier.value;

          if(supplier == 'Vaxzevria (AstraZeneca)'){
            effective_overall_doses = overall_doses;
          }

          emit(effective_overall_doses);
        """
    }
  },
  "aggs": {
    "data": {
          "date_histogram": {
        "field": "administration_date",
        "fixed_interval": "3650d"
      },
    "aggs": {
      "doses_amount": {
        "sum": {
          "field": "overall_doses"
        }
      },
      "pfizer_doses": {
        "sum": {
          "field": "pfizer_doses"
        }
      },
      "moderna_doses": {
        "sum": {
          "field": "moderna_doses"
        }
      },
      "janssen_doses": {
        "sum": {
          "field": "janssen_doses"
        }
      },
      "vaxzevria (astrazeneca)": {
```

(b)

```
      },
      "vaxzevria (astrazeneca)": {
        "sum": {
          "field": "vaxzevria (astrazeneca)"
        }
      },
      "pfizer_percentage": {
        "bucket_script": {
          "buckets_path": {
            "overallPfizerDoses": "pfizer_doses",
            "overallDoses": "doses_amount"
          },
          "script": "params.overallPfizerDoses / params.overallDoses *100"
        }
      },

            "moderna_percentage": {
        "bucket_script": {
          "buckets_path": {
            "overallModernaDoses": "moderna_doses",
            "overallDoses": "doses_amount"
          },
          "script": "params.overallModernaDoses / params.overallDoses *100"
        }
      },

            "astrazeneca_percentage": {
        "bucket_script": {
          "buckets_path": {
            "overallAstrazenecaDoses": "vaxzevria (astrazeneca)",
            "overallDoses": "doses_amount"
          },
          "script": "params.overallAstrazenecaDoses / params.overallDoses *100"
        }
      },

            "janssen_percentage": {
        "bucket_script": {
          "buckets_path": {
            "overallJanssenDoses": "janssen_doses",
            "overallDoses": "doses_amount"
          },
          "script": "params.overallJanssenDoses / params.overallDoses *100"
        }
      }
    }
  }
  }
}

```

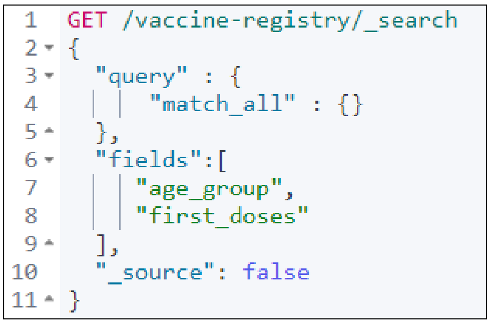
(c)

```
{
  "took" : 682,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 10000,
      "relation" : "gte"
    },
    "max_score" : null,
    "hits" : [ ]
  },
  "aggregations" : {
    "data" : {
      "buckets" : [
        {
          "key_as_string" : "2019-12-20T00:00:00.000Z",
          "key" : 1576800000000,
          "doc_count" : 172228,
          "janssen_doses" : {
            "value" : 1501444.0
          },
          "moderna_doses" : {
            "value" : 2.4004703E7
          },
          "vaxzevria (astrazeneca)" : {
            "value" : 1.2003321E7
          },
          "pfizer_doses" : {
            "value" : 7.6436863E7
          },
          "doses_amount" : {
            "value" : 1.13946331E8
          },
          "pfizer_percentage" : {
            "value" : 67.08146048160164
          },
          "moderna_percentage" : {
            "value" : 21.066674801490535
          },
          "astrazeneca_percentage" : {
            "value" : 10.534188239900415
          },
          "janssen_percentage" : {
            "value" : 1.3176764770074079
          }
        }
      ]
    }
  }
}

```

(d)

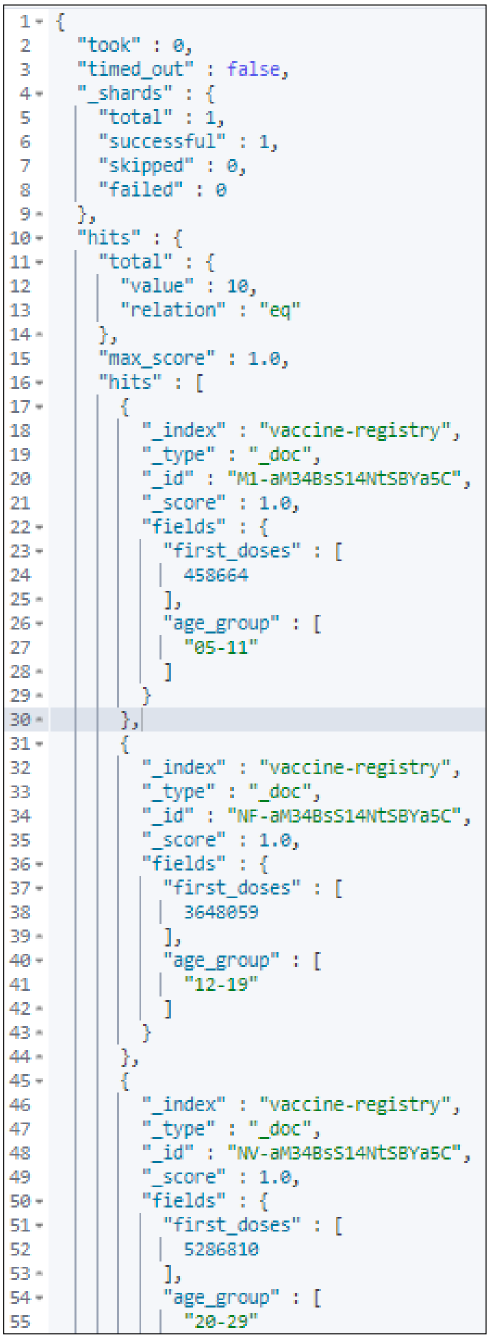
Figure 12: (a), (b), (c) show the query, while (d) shows a portion of the result

6.1.5 Number of people that have received at least one dose by age range.



```
GET /vaccine-registry/_search
{
  "query" : {
      "match_all" : {}
  },
  "fields":[
     "age_group",
     "first_doses"
  ],
  "_source": false
}
```

(a)



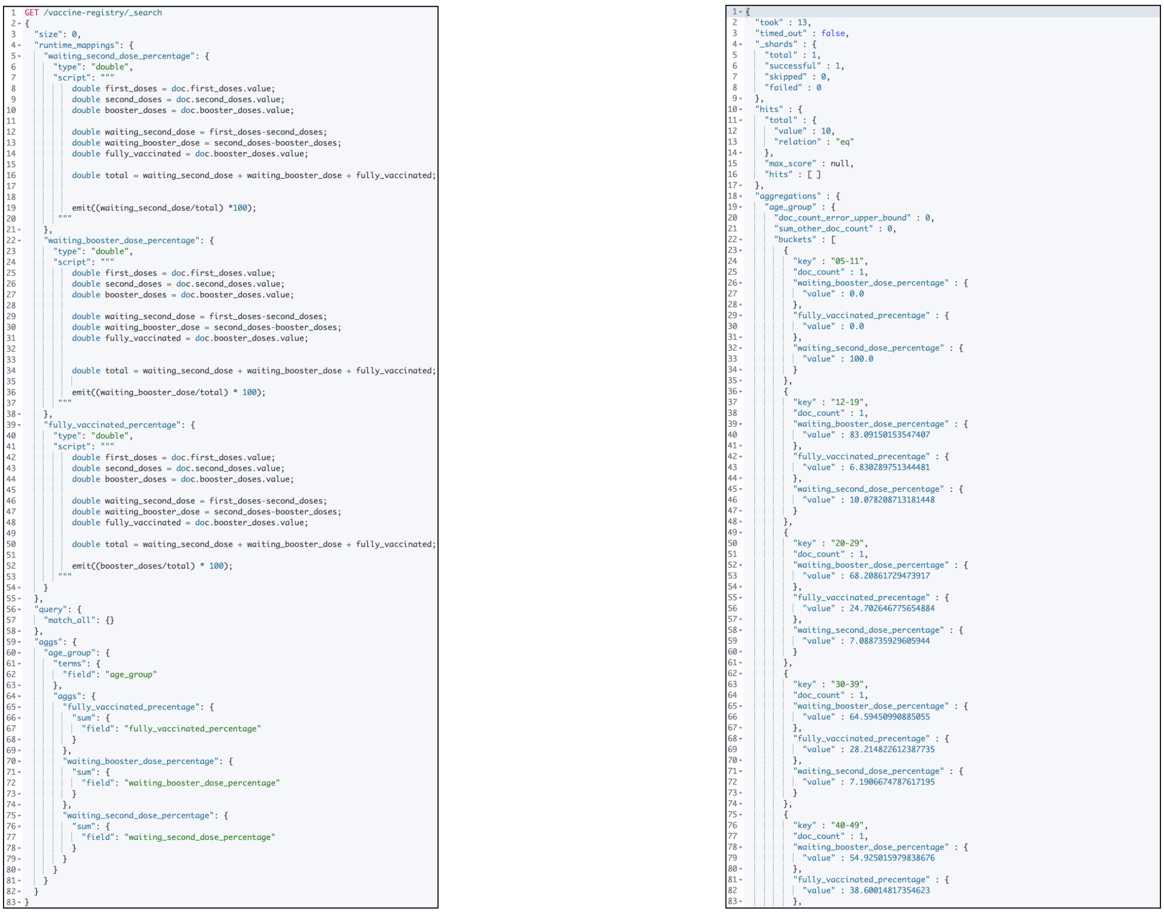
```
{
  "took" : 0,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 10,
      "relation" : "eq"
    },
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : "vaccine-registry",
        "_type" : "_doc",
        "_id" : "M1-aM34BsS14NtSBYa5C",
        "_score" : 1.0,
        "fields" : {
          "first_doses" : [
            458664
          ],
          "age_group" : [
            "05-11"
          ]
        }
      },
      {
        "_index" : "vaccine-registry",
        "_type" : "_doc",
        "_id" : "NF-aM34BsS14NtSBYa5C",
        "_score" : 1.0,
        "fields" : {
          "first_doses" : [
            3648059
          ],
          "age_group" : [
            "12-19"
          ]
        }
      },
      {
        "_index" : "vaccine-registry",
        "_type" : "_doc",
        "_id" : "NV-aM34BsS14NtSBYa5C",
        "_score" : 1.0,
        "fields" : {
          "first_doses" : [
            5286810
          ],
          "age_group" : [
            "20-29"
```

(b)

Figure 13: (a) shows the query, while (b) shows a portion of the result

6.1.6 Get vaccination status of all age groups

Note that this query has a dedicated dashboard's panel, as explained in section 5.6



```
GET /vaccine-registry/_search
{
  "size": 0,
  "runtime_mappings": {
    "waiting_second_dose_percentage": {
      "type": "double",
      "script": """
          double first_doses = doc.first_doses.value;
          double second_doses = doc.second_doses.value;
          double booster_doses = doc.booster_doses.value;

          double waiting_second_dose = first_doses-second_doses;
          double waiting_booster_dose = second_doses-booster_doses;
          double fully_vaccinated = doc.booster_doses.value;

          double total = waiting_second_dose + waiting_booster_dose + fully_vaccinated;


          emit((waiting_second_dose/total) *100);
        """
    },
    "waiting_booster_dose_percentage": {
      "type": "double",
      "script": """
          double first_doses = doc.first_doses.value;
          double second_doses = doc.second_doses.value;
          double booster_doses = doc.booster_doses.value;

          double waiting_second_dose = first_doses-second_doses;
          double waiting_booster_dose = second_doses-booster_doses;
          double fully_vaccinated = doc.booster_doses.value;


          double total = waiting_second_dose + waiting_booster_dose + fully_vaccinated;

          emit((waiting_booster_dose/total) * 100);
        """
    },
    "fully_vaccinated_percentage": {
      "type": "double",
      "script": """
          double first_doses = doc.first_doses.value;
          double second_doses = doc.second_doses.value;
          double booster_doses = doc.booster_doses.value;

          double waiting_second_dose = first_doses-second_doses;
          double waiting_booster_dose = second_doses-booster_doses;
          double fully_vaccinated = doc.booster_doses.value;

          double total = waiting_second_dose + waiting_booster_dose + fully_vaccinated;

          emit((booster_doses/total) * 100);
        """
    }
  },
  "query": {
    "match_all": {}
  },
  "aggs": {
    "age_group": {
      "terms": {
        "field": "age_group"
      },
      "aggs": {
        "fully_vaccinated_precentage": {
          "sum": {
            "field": "fully_vaccinated_percentage"
          }
        },
        "waiting_booster_dose_percentage": {
          "sum": {
            "field": "waiting_booster_dose_percentage"
          }
        },
        "waiting_second_dose_percentage": {
          "sum": {
            "field": "waiting_second_dose_percentage"
          }
        }
      }
    }
  }
}
```

(a)

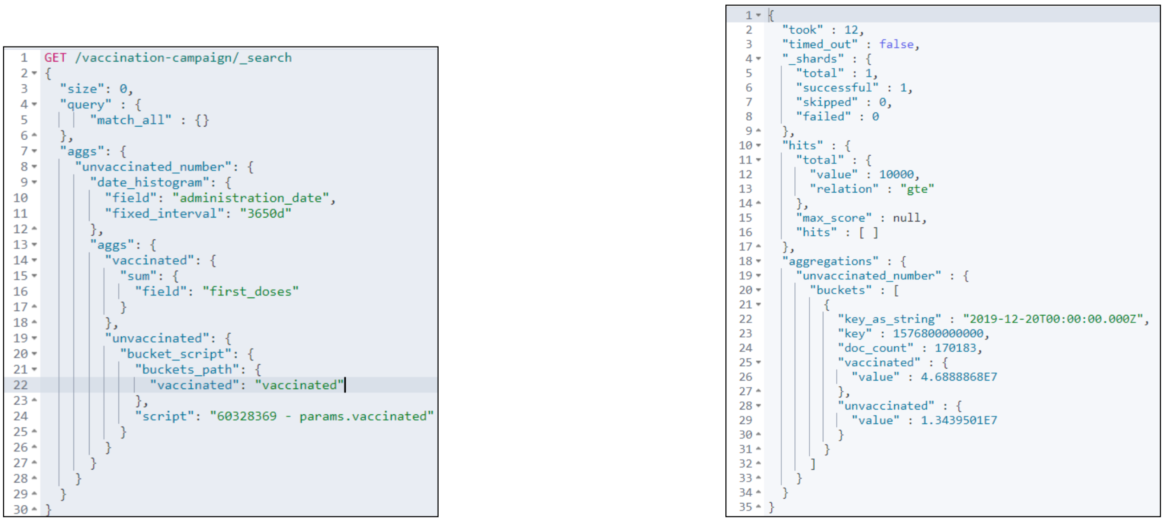
```
{
  "took" : 13,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 10,
      "relation" : "eq"
    },
    "max_score" : null,
    "hits" : [ ]
  },
  "aggregations" : {
    "age_group" : {
      "doc_count_error_upper_bound" : 0,
      "sum_other_doc_count" : 0,
      "buckets" : [
        {
          "key" : "05-11",
          "doc_count" : 1,
          "waiting_booster_dose_percentage" : {
            "value" : 0.0
          },
          "fully_vaccinated_precentage" : {
            "value" : 0.0
          },
          "waiting_second_dose_percentage" : {
            "value" : 100.0
          }
        },
        {
          "key" : "12-19",
          "doc_count" : 1,
          "waiting_booster_dose_percentage" : {
            "value" : 83.09150153547407
          },
          "fully_vaccinated_precentage" : {
            "value" : 6.830289751344481
          },
          "waiting_second_dose_percentage" : {
            "value" : 10.078208713181448
          }
        },
        {
          "key" : "20-29",
          "doc_count" : 1,
          "waiting_booster_dose_percentage" : {
            "value" : 68.20861729473917
          },
          "fully_vaccinated_precentage" : {
            "value" : 24.702646775654884
          },
          "waiting_second_dose_percentage" : {
            "value" : 7.088735929605944
          }
        },
        {
          "key" : "30-39",
          "doc_count" : 1,
          "waiting_booster_dose_percentage" : {
            "value" : 64.59450990885055
          },
          "fully_vaccinated_precentage" : {
            "value" : 28.214822612387735
          },
          "waiting_second_dose_percentage" : {
            "value" : 7.1906674787617195
          }
        },
        {
          "key" : "40-49",
          "doc_count" : 1,
          "waiting_booster_dose_percentage" : {
            "value" : 54.925015979838676
          },
          "fully_vaccinated_precentage" : {
            "value" : 38.60014817354623
          },
```

(b)

Figure 14: (a) shows the query, while (b) shows a portion of the result

6.1.7 Get the total number of not vaccinated people

Note that this query has a dedicated dashboard's panel, as explained in section 5.2



```
GET /vaccination-campaign/_search
{
  "size": 0,
  "query" : {
    "match_all" : {}
  },
  "aggs": {
    "unvaccinated_number": {
      "date_histogram": {
        "field": "administration_date",
        "fixed_interval": "3650d"
      },
      "aggs": {
        "vaccinated": {
          "sum": {
            "field": "first_doses"
          }
        },
        "unvaccinated": {
          "bucket_script": {
            "buckets_path": {
              "vaccinated": "vaccinated"
            },
            "script": "60328369 - params.vaccinated"
          }
        }
      }
    }
  }
}
```

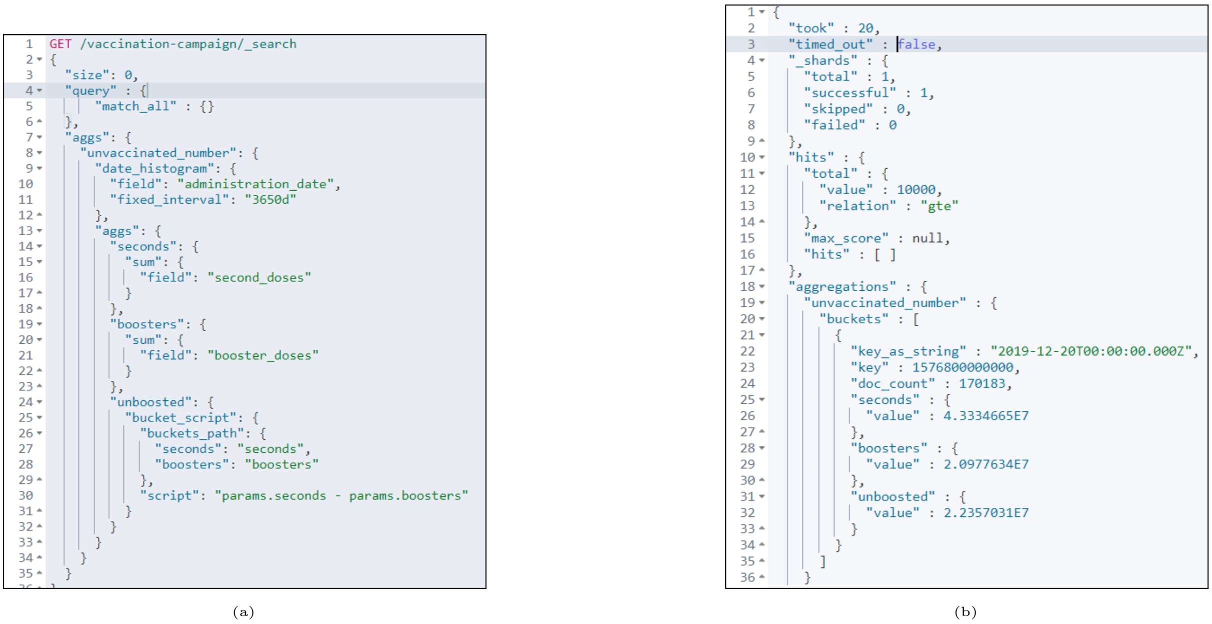
(a)

```
{
  "took" : 12,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 10000,
      "relation" : "gte"
    },
    "max_score" : null,
    "hits" : [ ]
  },
  "aggregations" : {
    "unvaccinated_number" : {
      "buckets" : [
        {
          "key_as_string" : "2019-12-20T00:00:00.000Z",
          "key" : 1576800000000,
          "doc_count" : 170183,
          "vaccinated" : {
            "value" : 4.6888868E7
          },
          "unvaccinated" : {
            "value" : 1.3439501E7
          }
        }
      ]
    }
  }
}
```

(b)

Figure 15: (a) shows the query, while (b) shows a portion of the result

6.1.8 Get the number of vaccinated people that still have to take the booster dose



```
GET /vaccination-campaign/_search
{
  "size": 0,
  "query" : {
      "match_all" : {}
  },
  "aggs": {
    "unvaccinated_number": {
      "date_histogram": {
        "field": "administration_date",
        "fixed_interval": "3650d"
      },
      "aggs": {
        "seconds": {
          "sum": {
            "field": "second_doses"
          }
        },
        "boosters": {
          "sum": {
            "field": "booster_doses"
          }
        },
        "unboosted": {
          "bucket_script": {
            "buckets_path": {
              "seconds": "seconds",
              "boosters": "boosters"
            },
            "script": "params.seconds - params.boosters"
          }
        }
      }
    }
  }
```

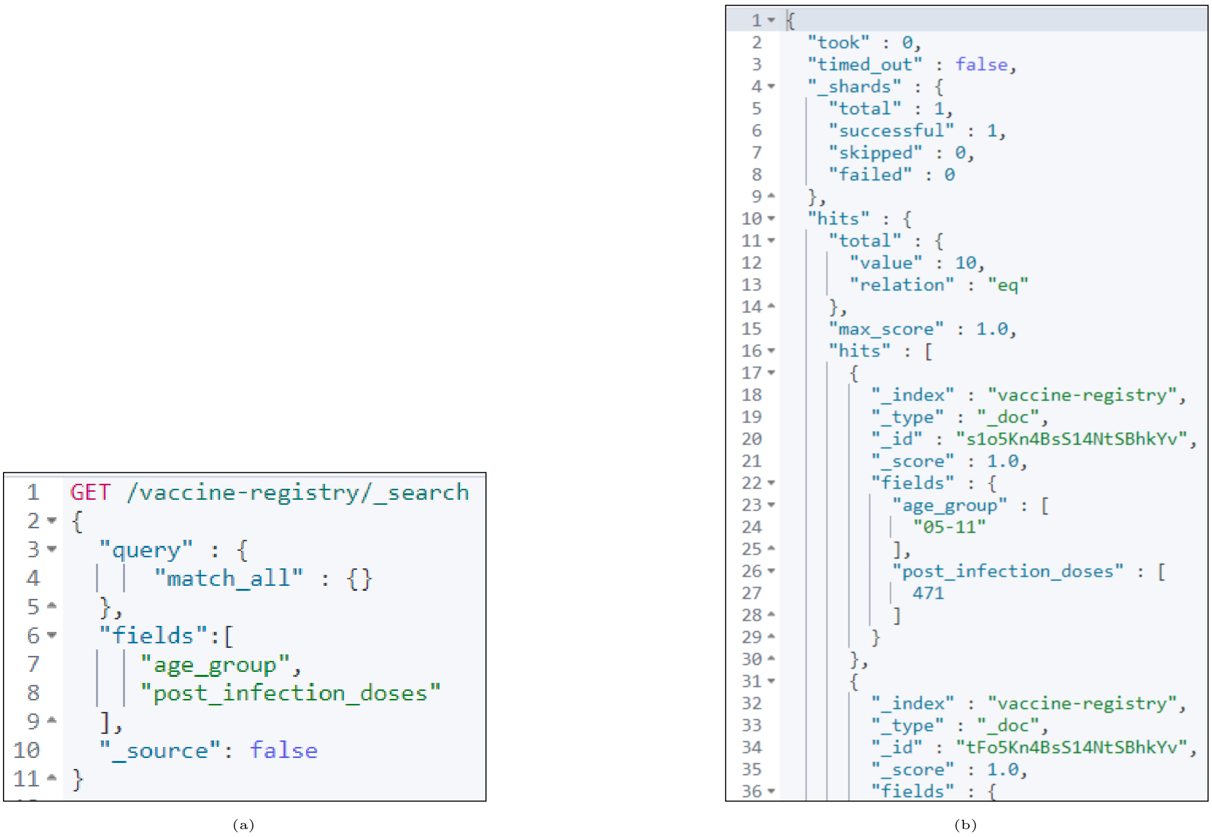
(a)

```
{
  "took" : 20,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 10000,
      "relation" : "gte"
    },
    "max_score" : null,
    "hits" : [ ]
  },
  "aggregations" : {
    "unvaccinated_number" : {
      "buckets" : [
        {
          "key_as_string" : "2019-12-20T00:00:00.000Z",
          "key" : 1576800000000,
          "doc_count" : 170183,
          "seconds" : {
            "value" : 4.3334665E7
          },
          "boosters" : {
            "value" : 2.0977634E7
          },
          "unboosted" : {
            "value" : 2.2357031E7
          }
        }
      ]
    }
```

(b)

Figure 16: (a) shows the query, while (b) shows a portion of the result

6.1.9 Get the number of people that have been tested positive in the previous 3/6 months and so needed only one dose of vaccine, by age-group



```
GET /vaccine-registry/_search
{
  "query" : {
      "match_all" : {}
  },
  "fields":[
      "age_group",
      "post_infection_doses"
  ],
  "_source": false
}
```

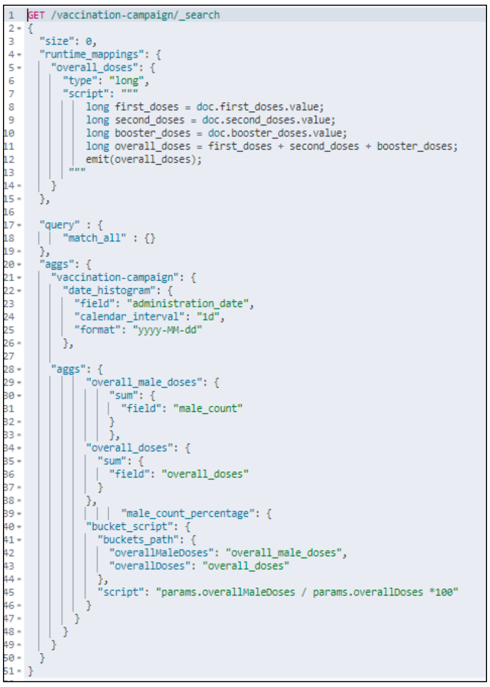
(a)

```
{
  "took" : 0,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 10,
      "relation" : "eq"
    },
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : "vaccine-registry",
        "_type" : "_doc",
        "_id" : "s1o5Kn4BsS14NtSBhkYv",
        "_score" : 1.0,
        "fields" : {
          "age_group" : [
            "05-11"
          ],
          "post_infection_doses" : [
            471
          ]
        }
      },
      {
        "_index" : "vaccine-registry",
        "_type" : "_doc",
        "_id" : "tFo5Kn4BsS14NtSBhkYv",
        "_score" : 1.0,
        "fields" : {
```

(b)

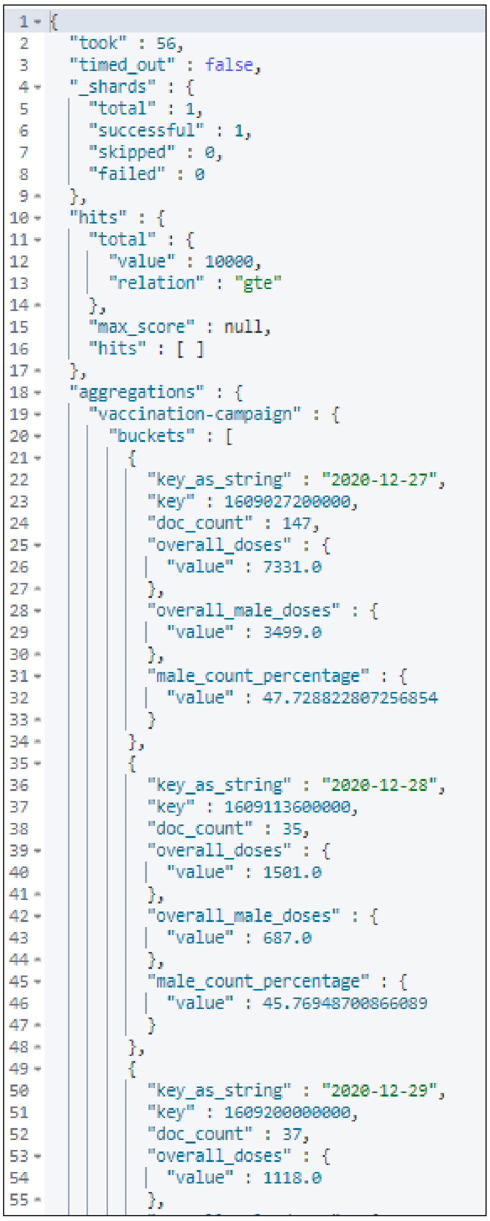
Figure 17: (a) shows the query, while (b) shows a portion of the result

6.1.10 Get the male percentage of administered vaccines by date



```
GET /vaccination-campaign/_search
{
  "size": 0,
  "runtime_mappings": {
    "overall_doses": {
      "type": "long",
      "script": """
          long first_doses = doc.first_doses.value;
          long second_doses = doc.second_doses.value;
          long booster_doses = doc.booster_doses.value;
          long overall_doses = first_doses + second_doses + booster_doses;
          emit(overall_doses);
       """
    }
  },

  "query" : {
      "match_all" : {}
  },
  "aggs": {
    "vaccination-campaign": {
      "date_histogram": {
        "field": "administration_date",
        "calendar_interval": "1d",
        "format": "yyyy-MM-dd"
      },

    "aggs": {
          "overall_male_doses": {
              "sum": {
                "field": "male_count"
              }
              },
          "overall_doses": {
            "sum": {
              "field": "overall_doses"
            }
          },
                "male_count_percentage": {
          "bucket_script": {
            "buckets_path": {
              "overallMaleDoses": "overall_male_doses",
              "overallDoses": "overall_doses"
            },
            "script": "params.overallMaleDoses / params.overallDoses *100"
          }
        }
      }
    }
  }
}
```

(a)

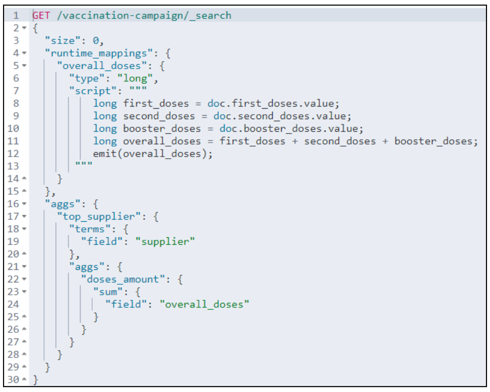


```
{
  "took" : 56,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 10000,
      "relation" : "gte"
    },
    "max_score" : null,
    "hits" : [ ]
  },
  "aggregations" : {
    "vaccination-campaign" : {
      "buckets" : [
        {
          "key_as_string" : "2020-12-27",
          "key" : 1609027200000,
          "doc_count" : 147,
          "overall_doses" : {
            "value" : 7331.0
          },
          "overall_male_doses" : {
            "value" : 3499.0
          },
          "male_count_percentage" : {
            "value" : 47.728822807256854
          }
        },
        {
          "key_as_string" : "2020-12-28",
          "key" : 1609113600000,
          "doc_count" : 35,
          "overall_doses" : {
            "value" : 1501.0
          },
          "overall_male_doses" : {
            "value" : 687.0
          },
          "male_count_percentage" : {
            "value" : 45.76948700866089
          }
        },
        {
          "key_as_string" : "2020-12-29",
          "key" : 1609200000000,
          "doc_count" : 37,
          "overall_doses" : {
            "value" : 1118.0
          },
```

(b)

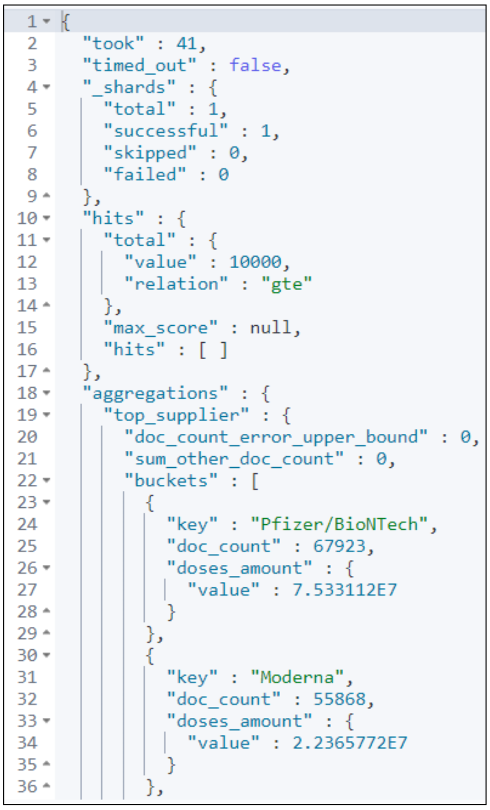
Figure 18: (a) shows the query, while (b) shows a portion of the result

6.1.11 Get total doses delivered by each supplier



```
GET /vaccination-campaign/_search
{
  "size": 0,
  "runtime_mappings": {
    "overall_doses": {
      "type": "long",
      "script": """
          long first_doses = doc.first_doses.value;
          long second_doses = doc.second_doses.value;
          long booster_doses = doc.booster_doses.value;
          long overall_doses = first_doses + second_doses + booster_doses;
          emit(overall_doses);
        """
    }
  },
  "aggs": {
    "top_supplier": {
      "terms": {
        "field": "supplier"
      },
      "aggs": {
        "doses_amount": {
          "sum": {
            "field": "overall_doses"
          }
        }
      }
    }
  }
}
```

(a)

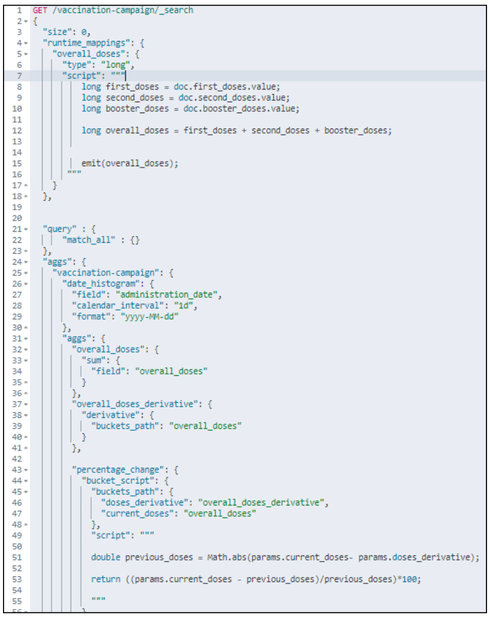


```
{
  "took" : 41,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 10000,
      "relation" : "gte"
    },
    "max_score" : null,
    "hits" : [ ]
  },
  "aggregations" : {
    "top_supplier" : {
      "doc_count_error_upper_bound" : 0,
      "sum_other_doc_count" : 0,
      "buckets" : [
        {
          "key" : "Pfizer/BioNTech",
          "doc_count" : 67923,
          "doses_amount" : {
            "value" : 7.533112E7
          }
        },
        {
          "key" : "Moderna",
          "doc_count" : 55868,
          "doses_amount" : {
            "value" : 2.2365772E7
          }
        },
```

(b)

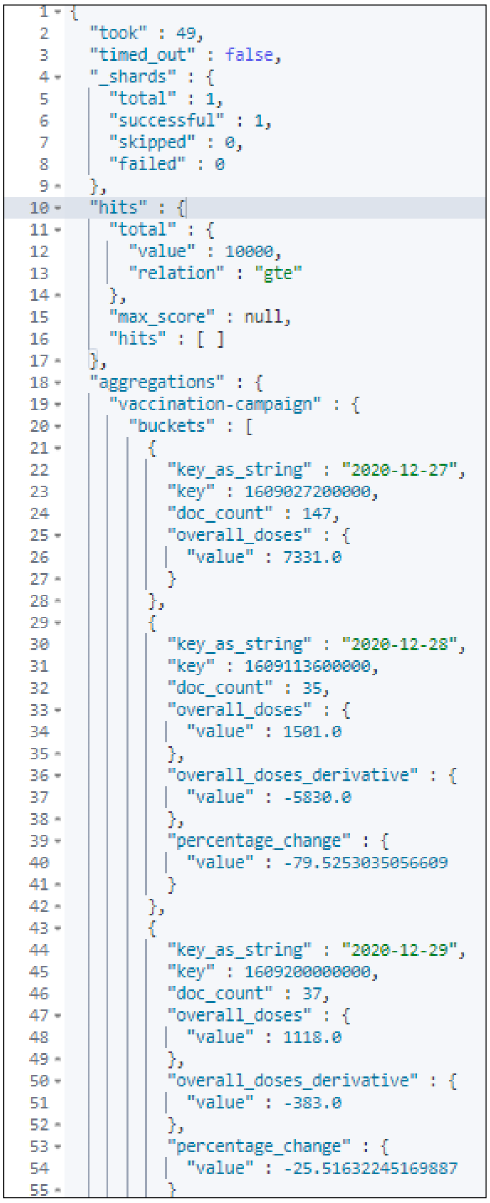
Figure 19: (a) shows the query, while (b) shows a portion of the result

6.1.12 Get the increase in percentage of vaccines done related to the previous day



```
GET /vaccination-campaign/_search
{
  "size": 0,
  "runtime_mappings": {
    "overall_doses": {
      "type": "long",
      "script": """
        long first_doses = doc.first_doses.value;
        long second_doses = doc.second_doses.value;
        long booster_doses = doc.booster_doses.value;

        long overall_doses = first_doses + second_doses + booster_doses;


        emit(overall_doses);
       """
    }
  },


  "query" : {
      "match_all" : {}
  },
  "aggs": {
    "vaccination-campaign": {
      "date_histogram": {
        "field": "administration_date",
        "calendar_interval": "1d",
        "format": "yyyy-MM-dd"
      },
      "aggs": {
        "overall_doses": {
          "sum": {
            "field": "overall_doses"
          }
        },
        "overall_doses_derivative": {
          "derivative": {
            "buckets_path": "overall_doses"
          }
        },

        "percentage_change": {
          "bucket_script": {
            "buckets_path": {
              "doses_derivative": "overall_doses_derivative",
              "current_doses": "overall_doses"
            },
            "script": """

            double previous_doses = Math.abs(params.current_doses- params.doses_derivative);

            return ((params.current_doses - previous_doses)/previous_doses)*100;

            """
          }
```

(a)



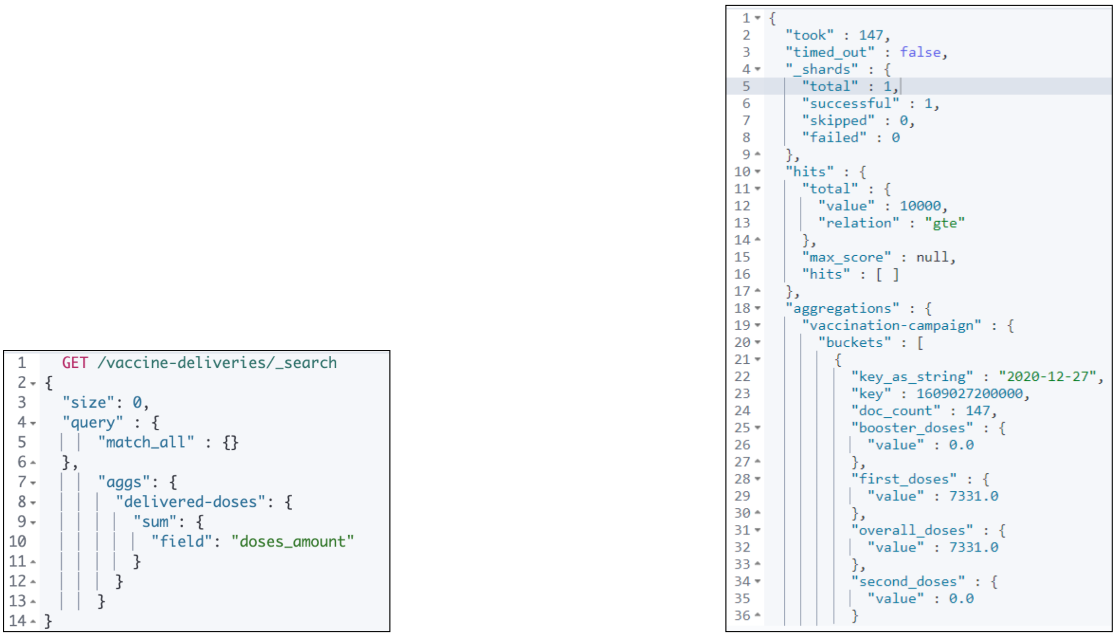
```
{
  "took" : 49,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 10000,
      "relation" : "gte"
    },
    "max_score" : null,
    "hits" : [ ]
  },
  "aggregations" : {
    "vaccination-campaign" : {
      "buckets" : [
        {
          "key_as_string" : "2020-12-27",
          "key" : 1609027200000,
          "doc_count" : 147,
          "overall_doses" : {
            "value" : 7331.0
          }
        },
        {
          "key_as_string" : "2020-12-28",
          "key" : 1609113600000,
          "doc_count" : 35,
          "overall_doses" : {
            "value" : 1501.0
          },
          "overall_doses_derivative" : {
            "value" : -5830.0
          },
          "percentage_change" : {
            "value" : -79.5253035056609
          }
        },
        {
          "key_as_string" : "2020-12-29",
          "key" : 1609200000000,
          "doc_count" : 37,
          "overall_doses" : {
            "value" : 1118.0
          },
          "overall_doses_derivative" : {
            "value" : -383.0
          },
          "percentage_change" : {
            "value" : -25.51632245169887
          }
```

(b)

Figure 20: (a) shows the query, while (b) shows a portion of the result

6.1.13 Get the overall number of doses delivered

Note that this query has a dedicated dashboard's panel, as explained in section 5.4



```
GET /vaccine-deliveries/_search
{
  "size": 0,
  "query" : {
      "match_all" : {}
  },
      "aggs": {
        "delivered-doses": {
          "sum": {
            "field": "doses_amount"
          }
        }
      }
}
```

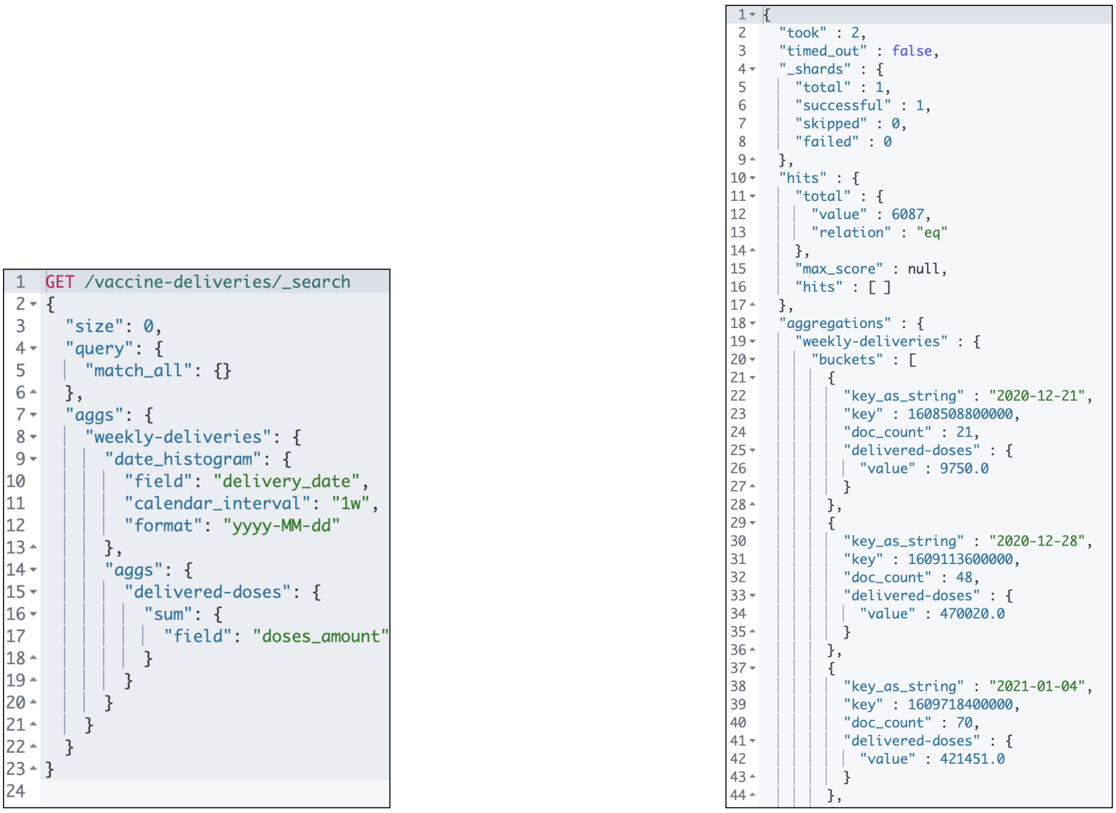
(a)

```
{
  "took" : 147,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 10000,
      "relation" : "gte"
    },
    "max_score" : null,
    "hits" : [ ]
  },
  "aggregations" : {
    "vaccination-campaign" : {
      "buckets" : [
        {
          "key_as_string" : "2020-12-27",
          "key" : 1609027200000,
          "doc_count" : 147,
          "booster_doses" : {
            "value" : 0.0
          },
          "first_doses" : {
            "value" : 7331.0
          },
          "overall_doses" : {
            "value" : 7331.0
          },
          "second_doses" : {
            "value" : 0.0
          }
```

(b)

Figure 21: (a) shows the query, while (b) shows a portion of the result

6.1.14 Get overall weekly vaccine deliveries amount



```
GET /vaccine-deliveries/_search
{
  "size": 0,
  "query": {
    "match_all": {}
  },
  "aggs": {
    "weekly-deliveries": {
      "date_histogram": {
        "field": "delivery_date",
        "calendar_interval": "1w",
        "format": "yyyy-MM-dd"
      },
      "aggs": {
        "delivered-doses": {
          "sum": {
            "field": "doses_amount"
          }
        }
      }
    }
  }
}
```

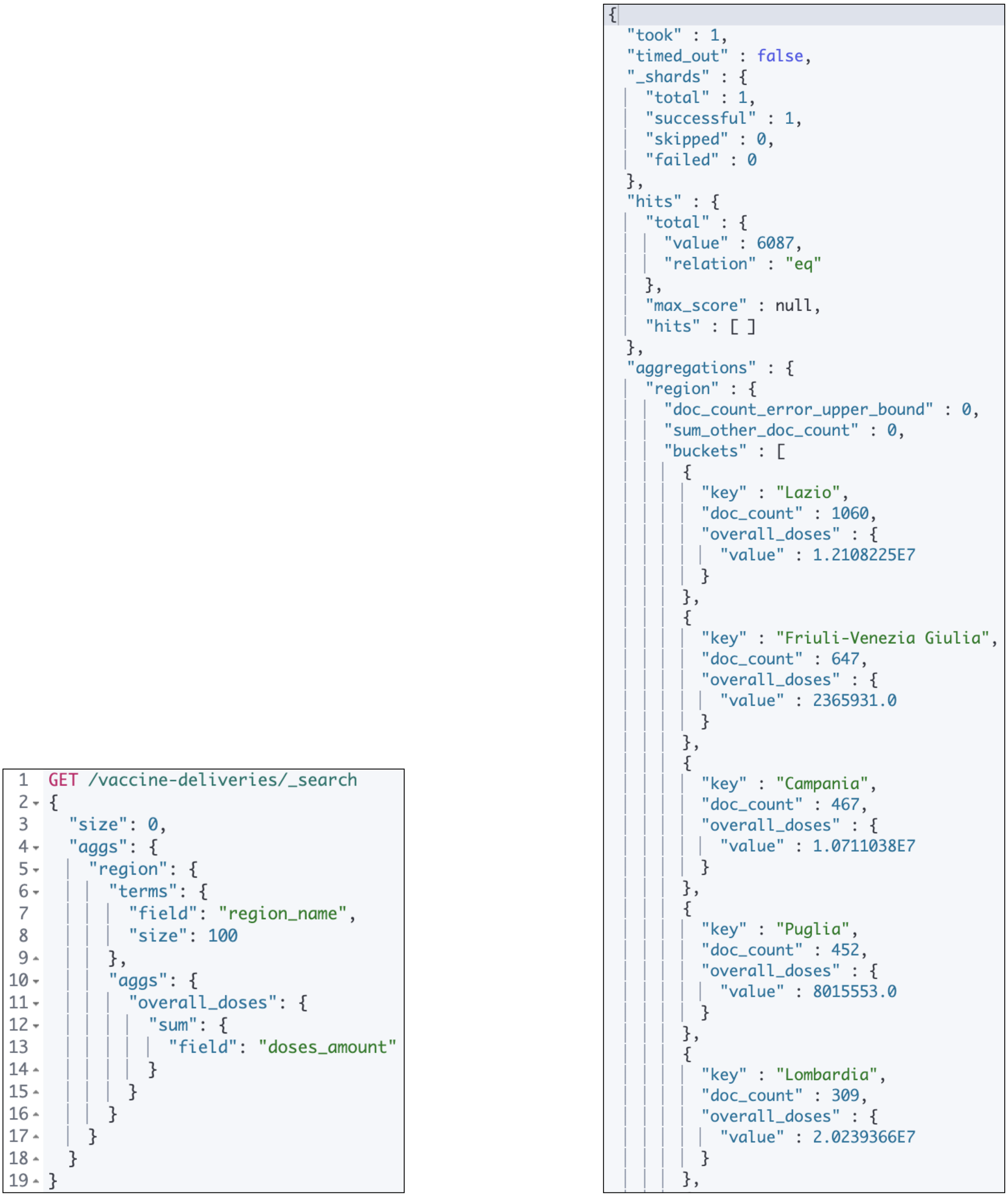
(a)

```
{
  "took" : 2,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 6087,
      "relation" : "eq"
    },
    "max_score" : null,
    "hits" : [ ]
  },
  "aggregations" : {
    "weekly-deliveries" : {
      "buckets" : [
        {
          "key_as_string" : "2020-12-21",
          "key" : 1608508800000,
          "doc_count" : 21,
          "delivered-doses" : {
            "value" : 9750.0
          }
        },
        {
          "key_as_string" : "2020-12-28",
          "key" : 1609113600000,
          "doc_count" : 48,
          "delivered-doses" : {
            "value" : 470020.0
          }
        },
        {
          "key_as_string" : "2021-01-04",
          "key" : 1609718400000,
          "doc_count" : 70,
          "delivered-doses" : {
            "value" : 421451.0
          }
        },
```

(b)

Figure 22: (a) shows the query, while (b) shows a portion of the result

6.1.15 Get vaccine deliveries amount by region name



```
GET /vaccine-deliveries/_search
{
  "size": 0,
  "aggs": {
    "region": {
      "terms": {
        "field": "region_name",
        "size": 100
      },
      "aggs": {
        "overall_doses": {
          "sum": {
            "field": "doses_amount"
          }
        }
      }
    }
  }
}
```

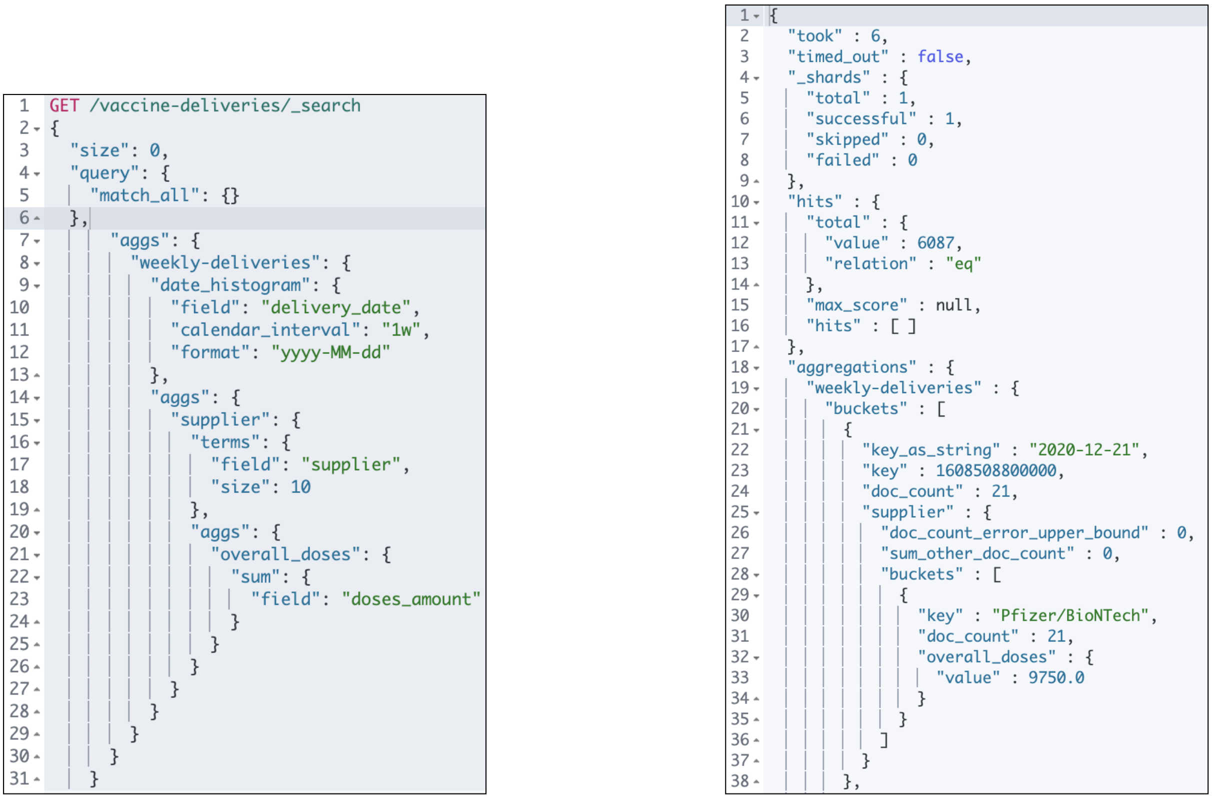
(a)

```
{
  "took" : 1,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 6087,
      "relation" : "eq"
    },
    "max_score" : null,
    "hits" : [ ]
  },
  "aggregations" : {
    "region" : {
      "doc_count_error_upper_bound" : 0,
      "sum_other_doc_count" : 0,
      "buckets" : [
        {
          "key" : "Lazio",
          "doc_count" : 1060,
          "overall_doses" : {
            "value" : 1.2108225E7
          }
        },
        {
          "key" : "Friuli-Venezia Giulia",
          "doc_count" : 647,
          "overall_doses" : {
            "value" : 2365931.0
          }
        },
        {
          "key" : "Campania",
          "doc_count" : 467,
          "overall_doses" : {
            "value" : 1.0711038E7
          }
        },
        {
          "key" : "Puglia",
          "doc_count" : 452,
          "overall_doses" : {
            "value" : 8015553.0
          }
        },
        {
          "key" : "Lombardia",
          "doc_count" : 309,
          "overall_doses" : {
            "value" : 2.0239366E7
          }
        },
```

(b)

Figure 23: (a) shows the query, while (b) shows a portion of the result

6.1.16 Get weekly vaccine deliveries amount by manufacturer



```
GET /vaccine-deliveries/_search
{
  "size": 0,
  "query": {
    "match_all": {}
  },
      "aggs": {
        "weekly-deliveries": {
          "date_histogram": {
            "field": "delivery_date",
            "calendar_interval": "1w",
            "format": "yyyy-MM-dd"
          },
          "aggs": {
            "supplier": {
              "terms": {
                "field": "supplier",
                "size": 10
              },
              "aggs": {
                "overall_doses": {
                  "sum": {
                    "field": "doses_amount"
                  }
                }
              }
            }
          }
        }
      }
    }
```

(a)

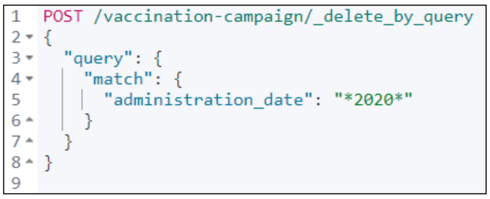
```
{
  "took" : 6,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 6087,
      "relation" : "eq"
    },
    "max_score" : null,
    "hits" : [ ]
  },
  "aggregations" : {
    "weekly-deliveries" : {
      "buckets" : [
        {
          "key_as_string" : "2020-12-21",
          "key" : 1608508800000,
          "doc_count" : 21,
          "supplier" : {
            "doc_count_error_upper_bound" : 0,
            "sum_other_doc_count" : 0,
            "buckets" : [
              {
                "key" : "Pfizer/BioNTech",
                "doc_count" : 21,
                "overall_doses" : {
                  "value" : 9750.0
                }
              }
            ]
          }
        },
```

(b)

Figure 24: (a) shows the query, while (b) shows a portion of the result

6.2 Commands

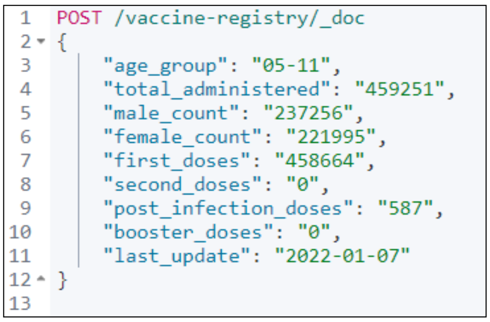
6.2.1 Delete all documents related to 2020 from vaccine campaign index



```
POST /vaccination-campaign/_delete_by_query
{
  "query": {
    "match": {
      "administration_date": "*2020*"
    }
  }
}

```

Figure 25: shows the deletion command

6.2.2 Insert a new document into vaccine registry index



```
POST /vaccine-registry/_doc
{
    "age_group": "05-11",
    "total_administered": "459251",
    "male_count": "237256",
    "female_count": "221995",
    "first_doses": "458664",
    "second_doses": "0",
    "post_infection_doses": "587",
    "booster_doses": "0",
    "last_update": "2022-01-07"
}

```

Figure 26: shows the insertion command

A Delivery content

Apart from this report, the delivery folder also contains the following files :

- **Python scripts** to create CSVs and indexes. The Python project is mainly composed by 5 files
  - **utils.py**: it contains various utility functions manipulate CSV files and to retrive files from Github repository.
  - **main.py**: it contains functions to handle the daily routine.
  - **constants.py**: it contains all the various constants, such as api keys, enums and files' paths, used among different functions.
  - **elasticutils.py**: it contains utility functions related to Elasticsearch, to create & delete indexes, create data mappings and upload data.
  - **csvmanipulation.py**: it contains various utility functions manipulate CSV files.
  - **setup.ini**: configuration file holding information about files' paths and Elasticsearch/GMaps connection API keys.
- **Folders** containing the various versions of CSVs files, to keep track of the subsequent modifications (these are located inside the Python project folder, inside the **datasets** folder) (note that in case of files deletion, these will be automatically recreated through **main.py** executable file)
- **Elasticsearch indexes**.
- **Text file** with queries&commands.
- **Kibana Complete dashboard & Single graphs** files.

References

[1] Google Maps API - `https://developers.google.com/maps/documentation/geocoding/overview`

[2] ELK stack - `https://www.elastic.co/what-is/elk-stack`

[3] Elasticsearch - `https://www.postdicom.com/it`

[4] Kibana - `https://www.elastic.co/kibana/`

[5] Python `https://www.python.org` b6Github File Watched `https://app.github-file-watcher.com`